

**DataGeneral**

---

---

**DIAGNOSTIC  
LISTING**

---

---

LISTING

096-000239-03

PROGRAM

EXERCISER FOR ECLIPSE  
PART 1

TAPE

095-000224-03

ABSTRACT

'ECLIPSE10' IS AN EXERCISER PROGRAM USED TO TEST THE RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF THE ECLIPSE COMPUTER. 'ECLIPSE10' EXERCISES THE EXTENDED INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES OF ITS RELIABLE OPERATION.

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

11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100

101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200



```

0001 ECL10 MACRO REV 03.00          14:20:43 08/06/76
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

*****
: NAME: ECLIPSE10.SR          PART NUMBER: 094-000623
:
: DESCRIPTION: ECLIPSE EXERCISER, PART 1
: REVISION HISTORY:
:   REV.          DATE
:   --          ---
:   00          08/02/74
:   01          12/20/74
:   02          04/11/75
:   03          08/06/76
:
: COPYRIGHT (C) DATA GENERAL CORPORATION, 1974, 1975, 1976
: ALL RIGHTS RESERVED.
*****

*****
: TITLE ECL10
: ECLIPSE10
:
: ECLIPSE10
:
: PART 1 OF EXERCISER FOR ECLIPSE
:
*****

00002 ECL10
02
03
04
05
06
07

```

10003 ECL10

01 ;  
02 ;  
03 ;  
04 ;  
05 ;  
06 ;  
07 ;  
08 ;  
09 ;  
10 ;  
11 ;  
12 ;  
13 ;  
14 ;  
15 ;  
16 ;  
17 ;  
18 ;  
19 ;  
20 ;  
21 ;  
22 ;  
23 ;  
24 ;  
25 ;  
26 ;  
27 ;  
28 ;  
29 ;  
30 ;  
31 ;  
32 ;  
33 ;  
34 ;  
35 ;  
36 ;  
37 ;  
38 ;  
39 ;  
40 ;  
41 ;  
42 ;  
43 ;  
44 ;  
45 ;  
46 ;  
47 ;  
48 ;

EXERCISER FOR ECLIPSE: PART 1  
PROGRAM NAME  
ECLIPSE10  
GENERAL DESCRIPTION  
ECLIPSE10 IS AN EXERCISER PROGRAM USED TO TEST THE RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF THE ECLIPSE COMPUTER. ECLIPSE10 EXERCISES THE EXTENDED INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES OF ITS RELIABLE OPERATION.  
THE INSTRUCTIONS EXERCISED ARE AS FOLLOWS:  
ADI,SGI,IOR,XGR,ANC,XCH,SGE,SGT,LSH,DLSH,LDB AND COB  
LOCATIONS 200 TO 205 IN PAGE 0 ARE FIXED FOR ECLIPSE10 PROGRAM.  
LOCATION 203 KEEPS TRACK OF NUMBER OF PASSES RUN THROUGH ECLIPSE10 PROGRAM.  
LOCATION 201 KEEPS TRACK OF THE TEST RUNNING AT PRESENT AND IS USEFUL FOR DEBUG WHEN LOOPING OCCURS IN THE PROGRAM.  
LOCATION 202 CONTAINS THE STARTING ADDRESS OF ECLIPSE10 PROGRAM.  
LOCATION 200 IS USED BY OTOS.  
LOCATION 204 KEEPS TRACK OF INTERNAL PASS COUNT WHICH IS FIXED BY LOCATION 205.  
FIRST PASS THROUGH ECLIPSE10 TEST WILL RUN SUPERFAST. NEXT PASSES WILL RUN SLOWER AS EACH TEST IS RUN SEVERAL TIMES TO RUN ALL RANDOM NUMBER COMBINATIONS.  
MACHINE REQUIREMENTS  
ECLIPSE PROCESSOR  
4K READ-WRITE MEMORY  
CONSOLE EQUIPMENT

10004 ECL10

01 ;  
02 ;  
03 ;  
04 ;  
05 ;  
06 ;  
07 ;  
08 ;  
09 ;  
10 ;  
11 ;  
12 ;  
13 ;  
14 ;  
15 ;  
16 ;  
17 ;  
18 ;  
19 ;  
20 ;  
21 ;  
22 ;  
23 ;  
24 ;  
25 ;  
26 ;  
27 ;  
28 ;  
29 ;  
30 ;  
31 ;  
32 ;  
33 ;  
34 ;  
35 ;  
36 ;  
37 ;  
38 ;  
39 ;  
40 ;  
41 ;  
42 ;  
43 ;  
44 ;

SWITCH SETTINGS  
THIS PROGRAM USES DATA SWITCHES AS FOLLOWS  
SW"0" - USE CONTENTS OF "SWREG" IF 0  
USE DATA SWITCHES IF 1  
SW"1" - LOOP ON FAILING TEST IF 0  
PROCEED TO NEXT TEST IF 1  
SW"2" - OUTPUT TO TTY IF 0  
INHIBIT PRINTING TO TTY IF 1  
SW"3" - DO NOT PRINT % ERRORS IF 0  
PRINT FAILURE RATE IF 1  
SW"4" - PRINT PASS COUNT IF 0  
INHIBIT PRINTING PASS COUNT IF 1  
SW"5" - INHIBIT OUTPUT TO LINE PRINTER IF 0  
OUTPUT TO LINE PRINTER IF 1  
OTOS PRESUMES THAT ALL SWITCHES ARE 0. RETURN IS MADE TO THE OPER SYS UPON AN ERROR AFTER FIRST PRINTING THE FAILURE REPORT. IN STAND ALONE MODE, THE OPERATOR IS ASKED TO PRESET THESE SWITCHES (THE MACHINE HALTS FOR HIM TO DO SO) AND PRESS CONTINUE. THIS PRESET VALUE CAN BE CHANGED AT ANY TIME BY SETTING DATA SWITCH "0" TO A ONE. THE PROGRAM WILL THEN FOLLOW ACTUAL SETTINGS.  
STAND ALONE STARTING ADDRESS = 200  
IF 'CAT' OR 'KITTEM' WAS LOADED FROM OTOS AND RESTART WAS NEEDED, THEN USE AS FOLLOWS:  
STARTING ADDR = 170 (FOR START WITH NO 'CAT')  
STARTING ADDR = 171 (FOR START WITH 'CAT')  
MONITOR LOCATION 203 TO CHECK THE CURRENT PASS COUNT  
MONITOR LOCATION X6000 TO MAKE SURE THAT 'CAT' OR 'KITTEM' IS RUNNING. IN CASES WHERE PROGRAM IS STARTED WITH 'CAT' OR 'KITTEM' LOCATION X6000 WILL SHOW A PATTERN CHANGING FROM ZEROS TO ALL ONES TO AN INC/SWAP PATTERN.  
(X= THE NUMBER OF THE HIGHEST MEMORY MODULE IN THE SYSTEM AND MAY BE A VALUE 0 - 7)

```

10005 ECL10
01
02 OPERATING PROCEDURE/OPERATOR INPUT
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

PROGRAM DESCRIPTION/THEORY OF OPERATION
-----
EACH TEST IS COMPLETE IN ITSELF, SO THE PROGRAM CAN
BE STARTED FROM ANY TEST WITHOUT CAUSING ANY
INITIALIZATION ERRORS.
WHEN 'ECLIPSE10' IS STARTED AT LOCATION 200 OR BY
DTDS, IT WILL SIZE UP THE MEMORY AND WILL PRINT
THE TOP OF THE MEMORY.
AFTER SETTING UP THE SWITCHES AND PRESSING CONTINUE,
THE EXERCISER WILL RUN THE FIRST PASS VERY FAST. IN
THE FIRST PASS EACH TEST IS RUN ONLY ONCE. ALL OTHER
PASSES WILL TAKE MORE TIME AS EACH TEST IS RUN
ACCORDING TO THE TEST COUNT SPECIFIED IN EACH TEST.
AFTER THE 1ST PASS, ECLIPSE10 IS RELOCATED IN THE
AVAILABLE MEMORY FOR ALL NEXT PASSES AND THE AREA
BELOW AND ABOVE THE RELOCATED PROGRAM IS USED
FOR SCRATCH BUFFER AREA. REFER TO THE LISTING TO
FIND OUT THE INFORMATION ABOUT EACH TEST.

RESTRICTIONS/MISC
-----
CERTAIN INSTRUCTIONS LIKE BLM, VCT, BAM, ETC.,
DO ALLOW INTERRUPTS TO OCCUR DURING THEIR
EXECUTION. THIS FEATURE OF THOSE INSTRUCTIONS IS
NOT CHECKED IN THIS TEST.

10006 ECL10
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

LOAD THE PROGRAM VIA THE BINARY LOADER OR INSERT A
PRELOADED MEMORY MODULE.
SET SWITCHES TO 200.
PRESS START.
PROGRAM WILL HALT AFTER PRINTING THE MESSAGE
/SET DATA SWITCHES AND PRESS CONTINUE/.
SET REQUIRED SWITCH SETTINGS AND PRESS CONTINUE. IN CASE
THE PROGRAM WILL RUN UNTIL MANUALLY STOPPED. IN CASE
OF MALFUNCTIONING, THE PROGRAM WILL PRINT ERROR
MESSAGE AND TAKE APPROPRIATE ACTION AS PER THE SW
SETTINGS.

PROGRAM OUTPUT/ERROR DESCRIPTION
-----
FOR ANY ERRORS DETECTED, THE PROGRAM WILL PRINT ERROR
REPORT OR % FAILURES DEPENDING UPON THE SW SETTINGS.
ERROR REPORT CONSISTS OF ALL ACCUMULATORS,CARRY,
RELOCATED PROGRAM COUNTER OF THE TEST THAT IS FAILING
AND PC IN THE LISTING AT THE TIME OF FAILURE.
THE PROGRAM WILL LOOP IN THE TEST THAT IS FAILING IF
SW#1 IS 0.
SW#1 IS 0.
THE PRINTING OF ERROR REPORT CAN BE ABORTED BY SETTING
SW#2 TO 1. OCCURS IN THE PROGRAM, STOP THE COMPUTER
IF LOOPING LOCATION 201 TO FIND OUT THE TEST THAT WAS
RUNNING BEFORE THE LOOPING OCCURRED.

```

```

10007 ECL10
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

: *****STANDARD MACROS*****
.MACRO LOOP
JSR @ENTLO
:ITERATE TEST ROUTINE
X

.MACRO SETUP
JSR @ENTIN
:INITIALIZE TEST.
X

.MACRO RAND
JSR @ENTRA
:C(CACO)=RANDOM #
X

.MACRO CALL
JSR @ICAL
:CALL SUBROUTINE *1
X

.MACRO ERROR
JMP *+2
**
JMP *+3
**
STA 3,AC3
**
JSR @ENTER
X

.MACRO JMPER
LDA 3,ITRER
MOV 3,3,5ZR
JMP *1 :JMP TO *1
X

:*****MACROS FOR THIS TEST*****
.MACRO ZEROAC
SUB 0,0
:CLEAR ALL AC'S
AND C(CARRY)
X

.MACRO SUBAL
SBI *1,0
SBI *1,1
SBI *1,2
SBI *1,3
:SUBTRACT FROM ALL AC'S
X

.MACRO ADDAL
ADI *1,0
ADI *1,1
ADI *1,2
ADI *1,3
:ADD TO ALL AC'S
X

.MACRO ASSEND
INC 0,1
INC 1,2
INC 2,3
:SET C(AC1) TO 1 GREATER
:THEN C(ACO), ETC.
X

.MACRO ACD
ZEROAC
ACD *2,*2
:THE VALUE IN AC*2
:SHOULD NOT EFFECT
:OTHER AC'S. CHECK
MOV# *2,*3&3,*2+2&3,*2R
:AC DESTINATION FOR
:THE *1 INSTRUCTION.
ERROR
X

```



```

10009 ECL10
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

.MACRO
XORACD
ZEROAC
ADC 1,1
XOR 0,1
ADD# 1+183,1+283,SNR
MOV# 1+383,1+383,SZR
ERROR
X
.MACRO
ANCACD
ZEROAC
ADC 1,1
ANC 0,1
ADD# 1+183,1+283,SNR
MOV# 1+383,1+383,SZR
ERROR
X
.MACRO
XCHACD
ZEROAC
ADC 1,1
XCH 1,1
ADD# 1+183,1+283,SNR
MOV# 1+383,1+383,SZR
ERROR
X
.MACRO
LSHACD
ZEROAC
ACC 1,1
LSH 1,1
ADD# 1+183,1+283,SNR
MOV# 1+383,1+383,SZR
ERROR
X

10010 ECL10
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

.MACRO
LOBACD
ANC 0,0
ANC 1,1
ANC 2,2
ANC 3,3
ADC 1,1
LOB 1,1
ADD# 1+183,1+283,SNR
MOV# 1+383,1+383,SZR
ERROR
X
.MACRO
LOBAL
SUBZL 1,1
SUB 1+283,1+283,SKP
120
LOB 1,1+283
MOVZL 1,1,SZR
JMP -2
LOA 1+383,-4
SUB# 1+283,1+383,SZR
ERROR
X
.MACRO
LRBACD
XOR 0,0
XOR 1,1
XOR 2,2
XOR 3,3
ACC 1,1
LRB 1,1
ADD# 1+183,1+283,SNR
MOV# 1+383,1+383,SZR
ERROR
X

;TEST THAT LRB RESULT
;GOES ONLY TO AC ^1
;RESET AC'S TO 0
;RESULT TO AC ^1

;THE "LOB INSTRUCTION WILL
;FIND 15 ZEROS THEN 14
;ETC. THESE VALUES SHOULD
;FORM THE SUM 120 IN C(ACD)
;(AC ^1+2)=LOB RESULT
;LOB FAILED

;TEST THAT LRB RESULT
;GOES ONLY TO AC ^1
;RESET ACC-3 TO ZERO
;RESULT TO AC ^1

```

10011 ECL10

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

```

\*MACRO  
 COBACD  
 ANC 0,0  
 ANC 1,1  
 ANC 2,2  
 ANC 3,3  
 COM 1,1  
 COB 1,1  
 ADD# 1+183,1+283,SNR  
 MOV# 1+383,1+383, SZR  
 ERROR  
 \*

\*MACRO  
 HSLACD  
 ANC 0,0  
 ANC 1,1  
 ANC 2,2  
 ANC 3,3  
 ADC 1,1  
 HXL 1,1  
 ADD# 1+183,1+283,SNR  
 MOV# 1+383,1+383, SZR  
 ERROR  
 \*

\*MACRO  
 HSRACD  
 XOR 0,0  
 XOR 1,1  
 XOR 2,2  
 XOR 3,3  
 ADC 1,1  
 HXR 1,1  
 ADD# 1+183,1+283,SNR  
 MOV# 1+383,1+383, SZR  
 ERROR  
 \*

\*MACRO  
 COB RESULT  
 \*GOES ONLY TO AC 1  
 \*PRESET ACO-3 TO ZERO  
 \*RESULT TO AC 1  
 \*

\*MACRO  
 HSL RESULT  
 \*GOES ONLY TO AC 1  
 \*PRESET ACO-3 TO ZERO  
 \*RESULT TO AC 1  
 \*

\*MACRO  
 HSR RESULT  
 \*GOES ONLY TO AC 1  
 \*PRESET ACO-3 TO ZERO  
 \*RESULT TO AC 1  
 \*

10012 ECL10

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42

```

\*MACRO  
 DUBL  
 SETUP 5  
 ADC 1,1  
 SUB 1+183,1+183  
 XOR 1+283,1+283  
 ANC 1+383,1+383  
 \*2 4,1  
 ADD# 0,1,SNR  
 ADD# 2,3, SZR  
 ERROR  
 LOOP  
 \*

\*MACRO  
 LRSHIFT  
 SETUP 40  
 RAND  
 MOV 0,1  
 MOV 0,2  
 MOV 0,3  
 DHXR 4,1  
 DHXL 4,1  
 MOV 1+183,1+183,SNR  
 SUB# 1,1+283, SZR  
 ERROR  
 LOOP  
 \*

\*MACRO  
 DL8  
 SETUP 40  
 RAND  
 MOV 0,1  
 MOV 1,1+183  
 MOVOS 1,1+383  
 DHXL 2,1  
 SUB# 1,1+383,SNR  
 MOV 0,0,SNR  
 ERROR  
 LOOP  
 \*

\*MACRO  
 \*TEST \*-2\*  
 \*SET AC 1 TO 17777  
 \*OTHER AC'S SET TO 0  
 \*  
 \*2 SHOULD CLEAR AC 1  
 \*ALL AC'S SHOULD NOW  
 \*BE CLEARED  
 \*  
 \*TEST \*DHSL/DHXR\*  
 \*(AC 1) SHOULD NOT CHANGE  
 \*  
 \*DOUBLE SHIFT 16 PLACES  
 \*RIGHT THEN LEFT  
 \*  
 \*SHIFT DOUBLE LEFT 8  
 \*  
 \*TEST THE HIGH ORDER PART  
 \*OF A DOUBLE SHIFT LEFT  
 \*8 PLACES TO AC 1  
 \*(CARRY) SHOULD NOT CHANGE  
 \*DHSL FAILED  
 \*



```

10013 ECL10
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

:DOUBLE SHIFT RIGHT 8
:TEST THE LOW ORDER PART
:OF THE DOUBLE SHIFT
:(CARRY) SHOULD NOT CHANGE
:SHIFT AC *1 AND *1+1.

.MACRO DRS8
  SETUP 40
  RAND
  MOV 0,*1
  MOV *1,*1+183
  MOVZ *1,*1+383
  DHR 2,*1
  SUB# *1+183,*1+383,SNR
  MOV 0,0,SZC
  ERROR
  LOOP
X

.MACRO DSR8
  SETUP 40
  RAND
  MOV 0,*1
  LDA *1+283,*2
  ANDS *1,*1+283,SKP
  177800
  MOVZ *1,*1+383
  MOV *1,*1+183
  DHR 2,*1
  SUB# *1+183,*1+283,SNR
  ERROR
  LOOP
Z

10014 ECL10
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

:DOUBLE SHIFT RIGHT 8
:TEST "DHSL" 8 TIMES LEFT
:1=A,B
:1+2=8,0
:1+3=8,A
:1+1=A,B
:1=B,A,*1+1=8,0
:DHSL FAILED
X

:8 PLACE DOUBLE SHIFT
:1=A,B
:1+2=0,A
:1+3=8,A
:1+1=A,B
:SHIFT *1 RIGHT 8 TIMES
:TEST HIGH PART

.MACRO DSL8
  SETUP 10
  ADCZ *1,*1,SKP
  7777
  LDA *1+183,*-1
  DMXL 1,*1
  SUB# *1,*1+183,SNR
  MOV 0,0,SZC
  ERROR
  LOOP
X

:TEST "DHSL" 4 TIMES LEFT
:7777 SHIFTED LEFT WILL
:EQUAL 177740, SAME AS
:HIGH ORDER PART
:(CARRY) SHOULD NOT CHANGE
:DHSL FAILED
X

:TEST "DHSR" 4 TIMES RIGHT
:17760 SHIFTED RIGHT=7777
:FALL ONES SHIFTED RIGHT
:ALSO EQUALS 7777
:(CARRY) SHOULD NOT CHANGE
:DHSR FAILED
X

```

```

10015 ECL10
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

;TEST DMSR/DMSL
OS12
SETUP 20
RAND
MOV 0,1
MOV 1,1+183
MOV 1+183,1+283
DHXR 3,1
DHXL 3,1
LDA 1+383,+2
AND 1+283,1+383,SMP
170000
SUB# 1,1+283,SNR
SUB# 1+183,1+383,SZR
ERROR
LOOP

;SHIFT RIGHT 12 PLACES
;THEN SHIFT LEFT 12

;CHECK HIGH PART
;CHECK LOW PART
ERROR
LOOP

;TEST DMSL/DMSR
DMSL2
SETUP 20
RAND
MOV 0,1
MOV 1,1+183
MOV 1+183,1+283
DHXL 3,1
DHXR 3,1
LDA 1+383,+2
AND 1+283,1+383,SMP
17
SUB# 1,1+383,SNR
SUB# 1+183,1+283,SZR
ERROR
LOOP

;DECIMAL ADD
;TEST THAT RESULT GOES TO -1, ONLY
DADAC0
SETUP 5
ZER0AC
ADCL 1,1
DAD 1,1
ADD# 1+183,1+283,SNR
MOV# 1+383,1+383,SZR
ERROR
LOOP

;DECIMAL SUBTRACT
;TEST THAT RESULT GOES
;TO AC -1, ONLY
DSBAC0
ZER0AC
SUBZL 1,1
DSB 1,1
ADD# 1+183,1+283,SNR
MOV# 1+383,1+383,SZR
ERROR

;CLEAR AC -1, SET C(CARRY).
;DECIMAL SUBTRACT SHOULD
;GIVE 0 RESULT AND SET
;C(CARRY). C(AC -1) NOT
;0 OR C(CARRY) IS 0
DSB1
SETUP 10
SUBZ 1,1
DSB 1,1
MOV 1,1,SNR
MOV 1,1,SZR
ERROR
LOOP

```



```

0017 ECL10
01
02
03 ***** DIAGNOSTIC PROGRAM PREAMBLE *****
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

; ***** LOCAL ZREL *****
; END OF PROGRAM
; RELOCATION RANGE OF PROGRAM
; POINTERS TO NON-RELOCATING SUBROUTINES
; RELOCATED CALL SUBROUTINE
; MESSAGE PRINT SUBROUTINE
; MEMORY SIZING SUBROUTINE
; PRINT OCTAL SUBROUTINE
; PRINT DECIMAL SUBROUTINE
; TEMPORARY STORAGE FOR
; MACHINE STATE IN 'LOADP'
; AND 'ERROR'
; SUBROUTINES.
; RETURN SAVE CHAR ROUT.
; CHARACTER LINE COUNT
; RETURN SAVE FR PRINT
; RETURN FR MESSAGE ROUT.
; RANDOM NUMBER
; TEMPORARYS FR CALL ROUT.
; TEMPORARYS FR BINARY ROUT.
; BLOCK MOVE ON PROGRAM
; ADJUST POINTERS TO
; RELOCATED SUBROUTINES
; EXIT TO TEST PROGRAM
; CALL SUBROUTINE
; INITIAL LOCATIONS OF RELOCATING SUBROUTINES
; TEST INITIALIZER
; TEST TERMINATOR
; ERROR ROUTINE
; RANDOM NUMBER GEN.
; CORRECTED VALUE=
; INITIAL VALUE +
; C(RELOC)
; STACK CONTROL LOCATIONS
; POINTER TO DIRTY BLOCK
; POINTER TO EGGS' BLOCK
; A LOCATIONS RESERVED
; FOR DEBUG BREAKPOINTS
; PERMANENT POINTER TO EGGS
; TOP OF MEMORY FROM SIZE
; START LOCATION OF CAT
; SPACE RESERVED FOR CAT/NO CAT RESTART ENTRIES
; THESE LOCATIONS
; RESERVED FOR FUTURE USE
; LOCATIONS 200 - 215 RESERVED FOR ECLIPSE TESTS
; PAGE ZERO STARTING LOC.
; LAST TEST ENTERED
; POINTER TO TEST SETUP ROUTINES
; PASS COUNT
; INTERVAL PASS COUNT
; INITIAL VALUE, INT. PASS COUNT
; ITERATION VALUE FOR THIS TEST
; ITERATION COUNTER
; ERROR SWITCH
; ERROR COUNTER
; ERROR RETURN
; LAST PLACE LOOP EXECUTED
; RELOCATION VALUE
; LISTING ADDR OF FAILING TEST
; LOC
; DIRT
; LOC
; LOC
; EGGS
; ZREL
; .BLK
; EGGS
; ICAT
; OFF
; ON
; CATSM
; JMP
; SWMESS
; LOC
; .BLK
; JMP
; OSBR
; ITRCT
; ITRER
; ITRFC
; ERRET
; LOPRET
; LOPRET
; LISTNG

```

```

10019 ECL10          .LOC 500
01 000500
02
03
04
05 00500 062677 NSTR: IORST
06 00501 06222 JSR @ISIZE
07 00502 150400 NEG
08 00503 150000 CDM
09 00504 050011- STA
10 00505 122470 ELDA
11 00506 06526 MOV
12 00507 101005 JMP NOCAT
13 00510 000414
14
15
16 00511 122470 ELDA
17 00512 06526 LDA
18 00513 024031- LDA
19 00514 125000 ADD
20 00515 112033 ADCZ#
21 00516 000406 JMP NOCAT
22
23 00517 132400 SUB
24 00520 024030- LDA
25 00521 147000 ADD
26 00522 044012- STA
27 00523 000406 JMP
28
29 00524 024027- NOCAT: LDA
30 00525 132400 SUB
31 00526 126400 SUB
32 00527 164470 ESTA
33 006504
34
35 00531 024216 STLOC: LDA
36 00532 132400 SUB
37 00533 050217 STA

```

```

10020 ECL10
01
02
03

```

```

? *****OUTPUT STRT MESSAGE & READ SWITCHES*****
04 00534 06221 @IMESS JSR @IMESS
05 00535 01302 MESIZ LDA
06 00536 024011- LDA 1, MEMTOP
07 00537 101040 MOV 0,0
08 00540 06223 JSR @IPOCT
09 00541 06221 JSR @IMESS
10 00542 001316 KCRLF
11
12 00543 126400 SUB
13 00544 044203 STA 1, PASS
14
15 00545 126470 ELDA 1, AUTO
16 00547 125004 MOV 1,1, SZR
17 00550 000413 JMP START
18
19
20 00551 006221 JSR @IMESS
21 00552 001350 SETSW @IMESS
22 00553 06221 JSR
23 00554 001316 KCRLF
24 00555 063077 HALT
25 00556 000401 JMP
26 00557 060477 READS 0
27 00560 142470 ESTA 0, SWREG
28
29 00562 000401 JMP START
30
31
32 00563 176400 SUB
33 00564 034214 STA 3,3
34 00565 000250 JMP BAMBY+1

```

```

START:

```

```

?SET RELOCATION CONSTANT TO 0
?AND START PROGRAM

```

```

?INIT PASS COUNT
?RUNNING IN AUTO MODE?
?YES START PROGRAM.
?NO, PRINT SET SWITCHES MESS.
?READ NEW STATE OF SWITCHES

```

```

SUB
STA
JMP

```

```

10021 ECL10
01
02
03
04
05
06 00566 175400 INIT: INC 3,3
07 00567 054201 STA 3,ITRER
08 00570 040225 STA 0,AC0
09
10 00571 021777 LDA 0,1,3
11 00572 040206 STA 0,ITR
12 00573 040207 STA 0,ITRCT
13
14 00574 020314 LDA 0,RELOC
15 00575 116400 SUB 0,3
16 00576 054215 STA 3,LISTNG
17
18 00577 176400 SUB 3,3
19 00600 054210 STA 3,ITRER
20 00601 054211 STA 3,ITREC
21
22 00602 034203 LDA 3,PASS
23 00603 175004 MOV 3,3,SZR
24 00604 000404 JMP INI1
25
26 00605 176320 SUBZL 3,3
27 00606 054206 STA 3,ITR
28 00607 054207 STA 3,ITRCT
29
30 00610 020225 INIT1: LDA 0,AC0
31 00611 002201 JMP @ITREY

```

```

; *****TEST UTILITY SUBROUTINES*****
; SUBROUTINE TO INITIALIZE A TEST LOOP
;
; TEST LOOP INITIALIZER
; SAVE RETURN LOCATION
; SAVE CONTENTS OF AC0
;
; GET # OF ITERATIONS
; SET ITER. VALUE
; SET ITER. COUNT
;
; COMPUTE AND SAVE
; THE LISTING ADDRESS
; OF THIS TEST.
;
; CLEAR ERROR SWITCH
; CLEAR ERROR COUNT
;
; TEST FOR FIRST PASS
;
; THIS IS 1'ST PASS
; SET ITERATIONS FOR
; 11 LOOP ONLY.
;
; RESTORE AC'S AND
; EXIT TO TEST

```

```

10022 ECL10
01
02
03
04 00612 054213 LOP: STA 3,LOPRET
05 00613 014207 ITRCT
06 00614 000440 JMP LOP3
07 00615 034210 LDA 3,ITRER
08 00616 175005 MOV 3,3,SNR
09 00617 002313 JMP @LOPRET
10 00620 034206 LDA 3,ITR
11 00621 054207 STA 3,ITRCT
12
13 00622 074477 LOP1: READS 3
14 00623 175112 MOVL# 3,3,SZC
15 00624 000403 JMP *3
16 00625 136470 ELDA 3,SMREG
17
18 00627 177100 ADDL 3,3
19 00630 177103 ADDL 3,3,SNC
20 00631 000421 JMP LOP2
21 00632 040225 STA 0,AC0
22 00633 044226 STA 1,AC1
23 00634 050227 STA 2,AC2
24 00635 006221 JSR @IMESS
25 00636 001234 PERCENT
26 00637 102400 SUB 0,0
27 00640 024211 LDA 1,ITREC
28 00641 040211 STA 0,ITREC
29 00642 030216 LDA 2,100
30 00643 143710 MUL
31 00644 030206 LDA 2,ITR
32 00645 153710 DIV
33 00646 006224 JSR @IPDEC
34 00647 020225 LDA 0,AC0
35 00650 024226 LDA 1,AC1
36 00651 030227 LDA 2,AC2
37 00652 176400 SUB 3,3
38 00653 054211 STA 3,ITREC
39
40 00654 034210 LOP3: LDA 3,ITRER
41 00655 175004 MOV 3,3,SZR
42 00656 074477 REAO5 3
43 00657 175112 MOVL# 3,3,SZC
44 00660 000403 JMP *3
45 00661 136470 ELDA 3,SMREG
46
47 00663 177113 ADDL# 3,3,SNC
48 00664 002201 JMP @ITREY
49 00665 002213 JMP @LOPRET

```

```

; SUBROUTINE TO TERMINATE A TEST LOOP
;
; END OF TEST ROUTINE
;
; ITERATION COMPLETE
;
; WHEN NO ERROR, EXIT TO NEXT
;
; RESET ITERATION COUNTER
;
; LOOK AT SWITCH #0"
; TO SEE WHETHER SMREG
; FOR DATA SWITCHES USED.
; USE SMREG
;
; IF SWITCH #3" PRINT %
; NO % PRINT OUT REQUIRED
;
; SAVE AC'S
; MESSAGE *#12><15> % FAIL="
;
; GET ERROR COUNT
; CLEAR ERROR COUNT.
;
; (COUNT X 100)/ ITERATIONS=
; PERCENTAGE OF FAILURE
; DECIMAL PRINTER
;
; IF NO ERROR, ITERATE
; OTHERWISE LOOK AT DATA
; #1" SWITCH FOR PROCEED,
; FOR NOT.

```

```

10023 ECL10
01
02
03
04 00656 054212 ERR: STA 3,ERR1
05 00667 040225 SUBCL 0,ACO
06 00670 102560 STA 0,0
07 00671 040231 STA 0,CRY
08 00672 010211 ISZ ITREC
09
10 00673 020210 LDA 0,ITRER
11 00674 116033 ADCZ# 0,3,SNCR
12 00675 000451 JMP ERR1
13
14 00676 054210 STA 3,ITRER
15 00677 044226 STA 1,AC1
16 00700 050227 STA 2,ACC2
17 00701 006221 JSR @IMESS
18 00702 001241 ERMSG
19 00703 024203 LDA 1,PASS
20 00704 125420 INCZ 1,1
21 00705 006224 JSR @IPDEC
22 00706 006221 JSR @IMESS
23 00707 011253 HEADER
24 00710 101020 MOVZ 0,0
25 00711 024231 LDA 1,CRY
26 00712 006226 JSR @IPDEC
27 00713 101040 MOVZ 0,0
28 00714 024225 LDA @IPOCT
29 00715 006225 JSR @IPOCT
30 00716 024226 LDA 1,AC1
31 00717 006223 JSR @IPOCT
32 00720 024227 LDA 1,AC2
33 00721 006223 JSR @IPOCT
34 00722 024230 LDA 1,AC3
35 00723 006223 JSR @IPOCT
36
37 00724 024212 LDA 1,ERR1
38 00725 020214 LDA 0,RELOC
39 00726 106420 SUBZ 0,1
40 00727 006223 JSR @IPOCT
41 00730 024212 LDA 1,ERR1
42 00731 006223 JSR @IPOCT
43 00732 006221 JSR @IMESS
44 00733 001316 KCRLF
45
46 00734 024226 LDA 1,AC1
47 00735 030227 LDA 2,ACC2
48 00736 136470 ELDA 3,AUTO
49 006273
50 00740 175005 MOV 3,3,SNR
51 00741 000405 JMP ERR1
52
53 00742 062677 IORST
54 00743 034010- LDA 3,EGGS
55 00744 035404 LDA 3,4,3
56 00745 001400 JMP 0,3
57
58 00746 020225 ERR1: LDA 0,ACO
59 00747 034230 LDA 3,AC3
60 00750 002212 JMP @ERR1

```

```

; ERROR SUBROUTINE: SETS ITRER FLAG, PRINTS MACHINE STATE.
;
; SAVE PC OF ERROR
; SAVE TEMPORARY
; ACO AND CARRY
; BUMP ERROR COUNT
; NEW ERROR?
; NO, RETURN.
; YES,
; SAVE MACHINE STATE
; PRINT "ERROR PASS" MSG.
;
; PRINT HEADER
; SET LEADING ZERO SUPR.
; SET LEADING ZERO PRINT
; PRINT MACHINE STATE
; PRINT LOGICAL &
; LISTING PC'S OF ERROR
; (LISTING PC)
; LOGICAL PC)
; FLUSH DEVICE BUFFERS
; RESTORE AC'S 1 & 2
; PROGRAM IN AUTO MODE?
; NOPE: GO BACK TO TEST
; AUTO MODE & SW6=0
; MAKE ERROR RETURN TO DT05
; FINISH RESTORING AC'S
; GO BACK TO TEST

```

```

0024 ECL10
01
02
03
04 00751 020236 .RAND: LDA 0,RAN
05 00752 024210 LDA 1,ITRER
06 00753 125004 MOV 1,1,SNR
07 00754 001400 JMP 0,3
08 00755 105000 MOV 0,1
09 00756 125410 HXL 2,1
10 00757 107000 ADD 0,1
11 00760 125120 MOVZL 1,1
12 00761 125120 MOVZL 1,1
13 00762 123000 ADD 1,0
14 00763 024025- LDA 1,-33031
15 00764 123000 ADD 1,0
16 00765 040236 STA 0,RAN
17 00766 001400 JMP 0,3
18
19
20
21
22 00767 040237 CAL: STA 0,CALO
23 00770 044240 STA 1,CAL1
24 00771 020214 LDA 0,RELOC
25 00772 025400 LDA 1,0,3
26 00773 123000 ADD 1,0,3
27 00774 040241 STA 0,CAL2
28 00775 175000 INC 3,3
29 00776 020237 LDA 0,CALO
30 00777 024240 LDA 1,CAL1
31 01000 002241 JMP @CAL2
32
33
34
35
36 01001 030216 SIZE: LDA 2,MINLOC
37 01002 151400 INC 2,2
38 01003 151112 MOVZL# 2,2,SNCR
39 01004 000406 JMP *,6
40 01005 021000 LDA 0,0,2
41 01006 051000 STA 2,0,2
42 01007 025000 LDA 1,0,2
43 01010 041000 STA 0,0,2
44 01011 132414 SUB# 1,2,SNCR
45 01012 001400 JMP 0,3
46 01013 050011- STA 2,MENTOP
47 01014 000766 JMP SIZE*1

```

```

; RANDOM NUMBER GENERATOR SUBROUTINE
; GENERATE A NEW RANDOM
; NUMBER IN C(ACO) AND C(RAN),
; IF C(ITRER)=0. OTHERWISE
; LOAD C(ACO) WITH OLD #.
; CALL ROUTINE TO REACH RELOCATED SUBROUTINES
; CALL A RELOCATED
; SUBROUTINE.....
; JSR (CAL)
; ADDRESS
; SIZE THE LOGICAL MEMORY
; MEM IS 32K WORDS.
; SAVE MEMORY TOP ADDR
; SIZE*1

```

```

10026 ECL10
01 ;*****PRINT ROUTINES*****
02
03 ;SAVE CARRY
04 ;DECIMAL PRINT C(AC1).
05 STA 3,POERET
06 MOVR 3,3
07 ;RESET C(CARRY) FOR ZERO SUPPRESSION
08 ;SET C(CARRY) IF NOT
09 MOVL 10000.
10 ;
11 ;
12 ;
13 ;
14 ;
15 ;
16 ;
17 ;
18 ;
19 ;
20 ;
21 ;
22 ;
23 ;
24 ;
25 ;
26 ;
27 ;
28 ;
29 ;
30 ;
31 ;
32 ;
33 ;
34 ;
35 ;
36 ;
37 ;
38 ;
39 ;
40 ;
41 ;
42 ;
43 ;
44 ;
45 ;
46 ;
47 ;
48 ;
49 ;
50 ;
51 ;
52 ;
53 ;
54 ;
55 ;
56 ;
57 ;
58 ;
59 ;
60 ;

```

```

10025 ECL10
01 ; RELOCATE SUBROUTINE:
02 ; ALLOCATES MEMORY FOR COPIES
03 ; OF TEST PROGRAM, WORKS IN CONJUNCTION
04 ; WITH BAM ROUTINE IN PAGE ZERO, WHICH
05 ; ACTUALLY COPIES THE TEST PROGRAM TO
06 ; THE NEW LOCATION.
07
08 REL: EISZ CATSW
09 JMP +*2
10 @ICAT
11 ;
12 ;
13 ;
14 ;
15 ;
16 ;
17 ;
18 ;
19 ;
20 ;
21 ;
22 ;
23 ;
24 ;
25 ;
26 ;
27 ;
28 ;
29 ;
30 ;
31 ;
32 ;
33 ;
34 ;
35 ;
36 ;
37 ;
38 ;
39 ;
40 ;
41 ;
42 ;
43 ;
44 ;
45 ;
46 ;
47 ;
48 ;
49 ;
50 ;
51 ;
52 ;
53 ;
54 ;
55 ;
56 ;
57 ;
58 ;
59 ;
60 ;

```

0027 ECL10  
01 01183 000767  
02

JMP MESS1

```
10028 ECL10
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

: LPT/TTO INTERFACE ROUTINE: CHARACTER PASSED IN AC0
CHAR:
MOVL 3,3
STA 3,CHARET
STA 2,CHRSV
:SAVE RETURN ADDR.
:SAVE AC2
:READ SWITCHES INTO AC3
:TEST SWITCH 0
:SWO SET
:SWO CLEAR, DEFAULT
:MASK SW2 & SW5 INTO
:AC2 FROM AC3
:LEFT JUSTIFY SW2
:COMPLEMENT SW2
:NO OUTPUT, RETURN
:MASK CHARACTER INTO L-BYTE
:OF AC3, CLEAR CARRY.
:IF NULL CHAR. RETURN
:DETERMINE REQUIRED
:STATE OF PARITY BIT &
:INSERT IT
:TEST FOR TAB
:TRUE: SETUP TAB SIMULATION
:RESTORE CHARACTER TO R-BYTE AC0,
:SET CARRY, BUMP LINE COUNT.
:SEND TO LPT?
:NOPE, MUST BE TTO
:O.K. PRINT CHARACTER
:WAIT FOR DONE
:CLEAR DEVICE
:SEND TO TTO?
:
:O.K. SEND CHARACTER
:WAIT FOR DONE
:CLEAR DEVICE
:IF TABBING, AND NOT
:FINISHED, LOOP
:TEST FOR CARRIAGE RETURN
:TRUE: ZERO LINE COUNT
:OTHERWISE RETURN
:SET UP TO TAB
:O(AC3) IS TWO'S COMPLEMENT
:OF # OF SPACES NEEDED
```



0029 ECL10  
01 01255 000746  
02  
03 01226 000000 CHRSV: 0

CHAR1

JMP

ITEMP SAVE FOR AC2

```
!0030 ECL10  
01 ;  
02 ;*****MESSAGE DATA BLOCK*****  
03 ;  
04 01227 005215 PASMES: .TXTE !<15><12>PASS !  
05 040520  
06 051523  
07 120240  
08 000000  
09 01234 005215 PERCEN: .TXTE !<15><12>X FAIL!  
10 120245  
11 040706  
12 146311  
13 000275  
14 01241 005215 ERMSG: .TXTE !<15><12><15><12>ERROR IN PASS: !  
15 005215  
16 151505  
17 147722  
18 120322  
19 047311  
20 050240  
21 051501  
22 035123  
23 000240  
24 01253 005215 HEADER: .TXTE !<15><12><15><12>  
25 005215  
26 01255 151503 CRY ACO AC1 AC2 AC3 LISTING LOGICAL<15><12>!  
27 004531  
28 141501  
29 120060  
30 040411  
31 130703  
32 004640  
33 141501  
34 120262  
35 050411  
36 051703  
37 146011  
38 051711  
39 144724  
40 043516  
41 146011  
42 043717  
43 141711  
44 146101  
45 005215  
46 000000  
47 01302 005215 MESIZ: .TXTE !<15><12>LAST LOGICAL ADDRESS=!  
48 040714  
49 152123  
50 146240  
51 043717  
52 141711  
53 146101  
54 006640  
55 042104  
56 142782  
57 051523  
58 000275  
59 01316 005215 MCRLF: .TXTE !<15><12>!  
60 000000
```

```

0031 ECL10
01 01320 142523 SETSW: .TXTE !SET DATA SWITCHS AND PRESS CONTINUE!
02 120324
03 040504
04 040724
05 051640
06 144727
07 141724
08 051510
09 040640
10 042116
11 050240
12 142722
13 051523
14 141640
15 047517
16 144724
17 052516
18 000305

10032 ECL10
01
02
03
04 01342 054242 RCS: STA 3,.BC3 !BCD SUBTRACTION....
05 01343 050243 STA 2,.BC2 !C(AC0)-C(AC1), RESULT IN
06 01344 044244 STA 1,.BC1 !C(AC0). C(AC1-2) UNCHANGED
07 01345 040245 STA 0,.BC0 !A BORROW WILL SET C(CARRY).
08 01346 034014- LDA 3,=17
09 01347 030013- LDA 2,=12
10
11 01350 126401 SUB 1,1,SKP
12 01351 124400 RCSI: NEG 1,1 !C(AC1) DETERMINES BORROW
13 01352 167000 ADD 3,1
14 01353 124000 COM 1,1
15 01354 167400 AND 3,1
16 01355 020244 LDA 0,.BC1
17 01356 107000 ADD 0,1
18 01357 020245 LDA 0,.BC0
19 01360 163400 AND 3,0
20 01361 157400 AND 3,1
21 01362 124243 SUBZ 1,0,SMC
22 01363 143060 ADDC 2,0
23
24 01364 024245 RCS2: LDA 1,.BC0
25 01365 164610 ANC 3,1
26 01366 123000 ADD 1,0
27 01367 040245 STA 0,.BC0
28 01370 126500 SUPL 1,1
29 01371 111610 DXL 1,2
30 01372 175004 MOV 3,Z,SZR
31 01373 000756 JMP RCSI
32 01374 125200 MOVR 1,1
33 01375 024244 LDA 1,.BC1
34 01376 050243 LDA 2,.BC2
35 01377 002242 JMP 0,.BC3
36
37
38

```

```

10033 ECL10
01 01400 024013--HEXN: LDA 1,=12
02 01401 030014--:C(CAO) TO A DECIMAL
03 01402 052246 STA 3,HEXRET
04 01403 155000 HEXN.: :NUMBER IN THE RANGE OF
05 01404 117400 AND 0,9.
06 01405 136432 SUBZ# 1,3,SZC
07 01406 120510 XOR 1,0.
08 01407 105610 DMXL 1,1
09 01410 151004 MOV 2,2,SZR
10 01411 000772 JMP HEXN.
11 01412 002246 JMP @HEXRET
12
13 01413 052242 BCA:
14 01414 050243 STA 2,=BC2
15 01415 042444 STA 1,=BC1
16 01416 034014-- LDA 3,=17
17 01417 030013-- LDA 2,=12
18 01420 040244 BCA1: STA 0,=BC0
19 01421 024244 AND 1,=BC1
20 01422 167402 AND 3,1,SZC
21 01423 162400 SUB 3,0
22 01424 163420 ANDZ 3,0
23 01425 123000 ADD 1,0
24 01426 142412 SUBZ 2,0,SKP
25 01427 142421 MOVZ 0,0
26 01430 101020 AND 3,0
27 01431 163400 LDA 1,=BC0
28 01432 024245 ANC 3,1
29 01433 164610 IOR 1,0
30 01434 120410 DMXL 1,2
31 01435 111610 MOV 3,3,SZR
32 01436 175004 JMP @CA1
33 01437 000761 LDA 1,=BC1
34 01440 024244 LDA 2,=BC2
35 01441 030243 JMP @BC3
36 01442 002242

```

```

10034 ECL10
01
02
03
04
05
06
07
08
09 01443 101000 BEGIN: MOV 0,0
10 SETUP 100
11 JSR @ENTLN
12 100
13 01445 000100
14 01446 102420 SUBZ 0,0
15 01447 100010 ADI 1,0
16 01450 101007 MOV 0,0,=SBN
17 ERROR
18 JSR @ENTLN
19 01455 006264
20
21 01456 006263 EX1: SETUP 100
22 01457 000100 JSR @ENTLN
23 01460 102020 ADGZ 0,0
24 01461 100010 ADI 1,0
25 01462 101002 MOV 0,0,=SZC
26 ERROR
27 01467 101004 MOV 0,0,=SZR
28 ERROR
29 JSR @ENTLN
30 01474 006264
31
32 01475 102400 EX2: SUB 0,0
33 SETUP 10
34 01476 006263 JSR @ENTLN
35 01477 000010
36 01500 105040 EX2.1: MOV# 0,1
37 01501 104010 ADI 1,1
38 01502 106014 ADC# 0,1,=SZR
39 ERROR
40 01507 101003 MOV 0,0,=SNC
41 ERROR
42 JMBR EX2.2
43 01514 030210 LDA 3,ITSR
44 01515 175004 MOV 3,3,SZR
45 01516 000403 JMP EX2.2
46 01517 101704 INC# 0,0,=SZR
47 01520 000760 JMP EX2.1
48 LOOP
49 01521 006264 EX2.2: JSR @ENTLN

```

```

:CONVERT THE NUMBER IN
:C(CAO) TO A DECIMAL
:NUMBER IN THE RANGE OF
:0 TO 9.
:
:
:
:
:BCD ADDITION WITHOUT THE
:USE OF "DAD" INSTRUCTION
:C(CAO)+C(CA1)
:RESULT TO C(CAO)
:C(CARRY) FROM PREVIOUS OPERATION
:IS USED. C(CARRY) WILL
:ALSO HAVE CORRECT STATE
:AT EXIT TIME
:SUBTRACT 12,SET
:C(CARRY)
:SHIFT TO NEXT DIGIT
:RESTORE C(CA1-2)

```

```

:INITIALIZE TEST.
:SET C(CARRY)=1,C(CAO)=0
:INCREMENT TO +1, DON'T
:DISTURB C(CARRY)
:EITHER CARRY OR C(CAO)
:IS 0, AND IN ERROR.
:ITERATE TEST ROUTINE
:INITIALIZE TEST.
:SET C(CAO)=1, C(CARRY)=0
:INCREMENT TO +0
:ADI SHOULD NOT SET CARRY.
:ADI SHOULD ADD +1, TO
:FORM ZERO RESULT.
:ITERATE TEST ROUTINE
:ADI TEST
:INITIALIZE TEST.
:SET C(CA1)=C(CAO)
:ADD IMMEDIATE (1) TO C(CA1)
:TEST FOR C(CA1) ONE GREATER
:THEN C(CAO).
:ADI SHOULD NOT
:CHANGE C(CARRY)
:IF A ERROR PRESENT
:JMP TO EX2.2
:ITERATE TEST ROUTINE

```

0036 ECL10  
 01  
 02 01616 006264  
 03

EXS.2: LOOP  
 JSR @ENTLO ;ITERATE TEST ROUTINE

10035 ECL10

```

01 SUB 0,0
02 SETUP 10
03 JSR @ENTIN
04 JSR @ENTIN
05 JSR @ENTIN
06 MOV 0,2
07 ADD #2 TO C(AC2)
08 MOV 2,2,SZC
09 ERROR
10 INC 0,1
11 INC 1,1 ;C(AC0)=ORIG
12 SUB# 1,2,SZR
13 JSR @ENTLO
14 JMPER EX3.2
15 LDA 3,ITRER
16 MOV 3,3,SZR
17 JMP EX3.2
18 INCS 0,0,SZR
19 JMP EX3.1
20 LOOP EX3.2
21 JSR @ENTLO
22
23
24 SUB 0,0
25 SETUP 10
26 JSR @ENTIN
27 JSR @ENTIN
28 MOV 0,3
29 ADD 3,3
30 INCR 0,1
31 INCL 1,1
32 MOV 3,2
33 SUB# 1,2,SZR
34 ERROR
35 JMPER EX4.2
36 LDA 3,ITRER
37 MOV 3,3,SZR
38 JMP EX4.2
39 INCS 0,0,SZR
40 JMP EX4.1
41 LOOP EX4.2
42 JSR @ENTLO
43
44 SUB 1,1
45 SETUP 10
46 JSR @ENTIN
47 JSR @ENTIN
48 MOV 1,0
49 ADI 4,0
50 INCR 1,2
51 INC 2,2
52 SUB# 2,0,SZR
53 ERROR
54 JMPER EX5.2
55 LDA 3,ITRER
56 MOV 3,3,SZR
57 JMP EX5.2
58 INCS 1,1,SZR
59 LOOP EX5.1
60

```

```

; "ADI" TEST
; INITIALIZE TEST.
; SET C(AC2)=C(AC0)
; ADD #2 TO C(AC2)
; ADI INSTRUCTION
; SHOULD NOT CHANGE C(CARRY)
;C(AC1)=CORRECT
;C(AC2)=ADI RESULT
; IF A ERROR PRESENT
; JMP TO EX3.2
; ITERATE TEST ROUTINE
; "ADI" TEST
; INITIALIZE TEST.
; SET C(AC3)=C(AC0)
; C(AC3)=C(AC3)+3
; THE C(AC1) IS MADE
; 3 GREATER THAN C(AC0).
;C(AC0)=ORIG C(AC3)
;C(AC1)=CORRECT
;C(AC2)=ADI RESULT
; IF A ERROR PRESENT
; JMP TO EX4.2
; ITERATE TEST ROUTINE
; "ADI" TEST
; INITIALIZE TEST.
; SET C(AC0)=C(AC1)
; SHOULD ADD (4) TO C(AC0)
;C(AC2) IS MADE 4 GREATER
; THAN C(AC1).
;C(AC2)=CORRECT
;C(AC1)=ORIG
;C(AC0)=ADI RESULT
; IF A ERROR PRESENT
; JMP TO EX5.2

```

```

10037 ECL10
01 01617 125400 EX6:
02
03 01620 006263
04 01621 000010
05 01622 131000 EX6.1:
06 000020
07 01623 170010
08 01624 170010
09 01625 170010
10 01626 170010
11 01627 170010
12 01630 170010
13 01631 170010
14 01632 170010
15 01633 170010
16 01634 170010
17 01635 170010
18 01636 170010
19 01637 170010
20 01640 170010
21 01641 170010
22 01642 170010
23 01643 020402
24 01644 123001
25 01645 000100
26 01646 112414
27
28
29 01653 034210
30 01654 175004
31 01655 000403
32 01656 125704
33 01657 000743
34
35 01660 006264
36

SUB 1.1
SETUP 10
JSR @ENTIN
10
MOV 1.2
ADI 4.2
ADI 4.2
ADI 4.2
ADI 4.2
ADI 4.2
ADI 4.2
ADI 4.2
ADI 4.2
ADI 4.2
ADI 4.2
ADI 4.2
ADI 4.2
ADI 4.2
ADI 4.2
ADI 4.2
ADI 4.2
ADI 4.2
ADI 4.2
ADI 4.2
LDA 0.1+2
100
SUB# 0.2,SZR
IMPER EX6.2
LDA 3.1TBR
MOV 3.1,SZR
JMP EX6.2
INCS 1.1,SZR
JMP EX6.1
LOOP
EX6.2: JSR @ENTLO
36

10038 ECL10
01
02
03
04 01661 006263
05 01662 000100
06 01663 102400
07 01664 105400
08 01665 131400
09 01666 155400
10
11 01667 100010
12 01670 104010
13 01671 110010
14 01672 114010
15
16 01673 120010
17 01674 124010
18 01675 130010
19 01676 134010
20
21 01677 140010
22 01700 146010
23 01701 150010
24 01702 154010
25
26 01703 160010
27 01704 164010
28 01705 170010
29 01706 174010
30 01707 106014 EX7.1:
31
32 01714 132014
33
34 01721 161000
35 01722 142014
36
37
38 01727 006264
39

; "ADI" TEST
; INITIALIZE TEST.
; ADD ALL VALUES TO ALL
; ACCUMULATORS
; C(AC1) IS 1 GREATER THAN
; C(AC0), C(AC2) IS 1
; GREATER THAN C(AC1), ETC.

SUB# 0.0
INC 0.1
INC 1.2
INC 2.3
ADDA 1
ADI 1.0 ; ADD TO ALL AC'S
ADI 1.1
ADI 1.2
ADI 1.3
ADDA 2
ADI 2.0 ; ADD TO ALL AC'S
ADI 2.1
ADI 2.2
ADI 2.3
ADDA 3
ADI 3.0 ; ADD TO ALL AC'S
ADI 3.1
ADI 3.2
ADI 3.3
ADDA 4
ADI 4.0 ; ADD TO ALL AC'S
ADI 4.1
ADI 4.2
ADI 4.3
ADCA# 0.1,SZR
ERROR
ADCA# 1.2,SZR
ERROR
MOV 3.0
ADCA# 2.0,SZR
ERROR
LOOP
JSR @ENTLO
; "ADI" TO AC3 FAILED
; ITERATE TEST ROUTINE

```

10039 ECL10  
01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56

1  
7

10040 ECL10  
01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41

\*\*\* SUBTRACT IMMEDIATE \*\*\*

EX8:  
SUBTRACT IMMEDIATE \*\*\*  
INITIALIZE TEST.  
DO 8.  
SUBTRACT 1 FROM C(CA0).  
C(CA0) SHOULD=(-1)  
SBI FAILED.  
OTHER AC'S AND CARRY  
SHOULD NOT BE CHANGED.  
ITERATE TEST ROUTINE

EX9:  
SUBTRACT IMMEDIATE \*\*\*  
INITIALIZE TEST.  
DO 8.  
SUBTRACT 1 FROM C(AC3).  
C(AC3) SHOULD=(-1)  
SBI FAILED.  
OTHER AC'S AND CARRY  
SHOULD NOT BE CHANGED.  
ITERATE TEST ROUTINE

EXA.1:  
SUBTRACT IMMEDIATE \*\*\*  
INITIALIZE TEST.  
DO 8.  
SUBTRACT 1 FROM C(AC1).  
C(AC1) SHOULD=(-1)  
SBI FAILED.  
OTHER AC'S AND CARRY  
SHOULD NOT BE CHANGED.  
ITERATE TEST ROUTINE

EXA.2:  
SUBTRACT IMMEDIATE \*\*\*  
INITIALIZE TEST.  
DO 8.  
SUBTRACT 1 FROM C(AC2).  
C(AC2) SHOULD=(-1)  
SBI FAILED.  
OTHER AC'S AND CARRY  
SHOULD NOT BE CHANGED.  
ITERATE TEST ROUTINE

EXB:  
SUBTRACT IMMEDIATE \*\*\*  
INITIALIZE TEST.  
DO 8.  
SUBTRACT 1 FROM C(AC0).  
C(AC0) SHOULD=(-1)  
SBI FAILED.  
OTHER AC'S AND CARRY  
SHOULD NOT BE CHANGED.  
ITERATE TEST ROUTINE

EXB.1:  
SUBTRACT IMMEDIATE \*\*\*  
INITIALIZE TEST.  
DO 8.  
SUBTRACT 1 FROM C(AC0).  
C(AC0) SHOULD=(-1)  
SBI FAILED.  
OTHER AC'S AND CARRY  
SHOULD NOT BE CHANGED.  
ITERATE TEST ROUTINE

EXB.2:  
SUBTRACT IMMEDIATE \*\*\*  
INITIALIZE TEST.  
DO 8.  
SUBTRACT 1 FROM C(AC2).  
C(AC2) SHOULD=(-1)  
SBI FAILED.  
OTHER AC'S AND CARRY  
SHOULD NOT BE CHANGED.  
ITERATE TEST ROUTINE

EXC:  
SUBTRACT IMMEDIATE \*\*\*  
INITIALIZE TEST.  
DO 8.  
SUBTRACT 1 FROM C(AC2).  
C(AC2) SHOULD=(-1)  
SBI FAILED.  
OTHER AC'S AND CARRY  
SHOULD NOT BE CHANGED.  
ITERATE TEST ROUTINE

EXD:  
SUBTRACT IMMEDIATE \*\*\*  
INITIALIZE TEST.  
DO 8.  
SUBTRACT 1 FROM C(AC2).  
C(AC2) SHOULD=(-1)  
SBI FAILED.  
OTHER AC'S AND CARRY  
SHOULD NOT BE CHANGED.  
ITERATE TEST ROUTINE



```

10043 ECL10
01 02262 175004
02 02263 000406
03 02264 020712
04 02265 101700
05 02266 040710
06 02267 101004
07 02270 000712
08
09 02271 006264
10

MOV 3,3,SZR
JMP EXEC.2
LDA EXEC+1
INCS 0,0
STA EXEC+1
MOV 0,0,SZR
JMP EXEC.1

:ITERATE 512 TIMES.

:ITERATE TEST ROUTINE

0044 ECL10
01 02262 175004
02 02263 000406
03 02264 020712
04 02265 101700
05 02266 040710
06 02267 101004
07 02270 000712
08
09 02271 006264
10

:IMMEDIATE ADD/SUBTRACT
:OF NUMBERS 1-4 ARE
:ARE USED ON ALL AC'S
:INITIALIZE TEST.
JSR @ENTIN
5
LDA 0,EXEC+1
:(AC0)=STARTING NUMBER
ASSEN0
INC 0,1
:SET C(AC1) TO 1 GREATER
:THEN C(AC0), ETC.
INC 1,2
INC 2,3
SUBAL 4
SBI 4,0 :SUBTRACT FROM ALL AC'S
SBI 4,1
SBI 4,2
SBI 4,3
ADDA 4
ADI 4,0 :ADD TO ALL AC'S
ADI 4,1
ADI 4,2
ADI 4,3
SUBAL 1
SBI 1,0 :SUBTRACT FROM ALL AC'S
SBI 1,1
SBI 1,2
SBI 1,3
ADDA 1
ADI 1,0 :ADD TO ALL AC'S
ADI 1,1
ADI 1,2
ADI 1,3
SUBAL 2
SBI 2,0 :SUBTRACT FROM ALL AC'S
SBI 2,1
SBI 2,2
SBI 2,3
ADDA 2
ADI 2,0 :ADD TO ALL AC'S
ADI 2,1
ADI 2,2
ADI 2,3
SUBAL 3
SBI 3,0 :SUBTRACT FROM ALL AC'S
SBI 3,1
SBI 3,2
SBI 3,3
ADDA 3
ADI 3,0 :ADD TO ALL AC'S
ADI 3,1
ADI 3,2
ADI 3,3
ADC# 2,3,SNR
:(AC3) SHOULD BE 1 GREATER
:(THAN C(AC2), C(AC3) IS 1 GREATER
ERROR
LDC# 0,1,SZR
JMPER EXEC.2
LDA 3,ITRER

```





```

10047 ECL10
01
02 02426 006263
03 02427 000010
04
05 02430 102400
06 02431 126400
07 02432 152400
08 02433 176440
09 02434 152000
10 02435 140410
11 02436 144410
12 02437 154410
13 02440 116414
14 02441 000403
15 02442 124415
16 02443 142414
17
18
19 02450 006264
20

      EXL:
      SETUP 10
      JSR @ENTIN
      10
      ZEROAC
      SUB 0,0
      SUB 1,1
      SUB 2,2
      SUBO 3,3
      ADC 2,2
      IOR 2,0
      IOR 2,1
      IOR 2,3
      SUB# 0,3,SZR
      JMP ,*5
      SUB# 1,0,SNR
      SUB# 2,0,SZR
      ERROR
      LOOP
      JSR @ENTLO

      TEST "IOR"
      INITIALIZE TEST.

      CLEAR ALL AC'S
      AND C(CARRY)

      IC(AC0,1,3)=0
      IC(AC2)=(-1)
      INCLUSIVE OR FROM
      FACE TO EVERY OTHER AC.
      AFTER THE IOR INSTRUCTIONS.

      ITERATE TEST ROUTINE

10048 ECL10
01
02
03
04 02451 006263
05 02452 000010
06
07 02453 102400
08 02454 126400
09 02455 152400
10 02456 176440
11 02457 176040
12 02460 160410
13 02461 164410
14 02462 170410
15 02463 101002
16 02464 164414
17 02465 000403
18 02466 124415
19 02467 142414
20
21
22 02474 006264
23
24
25 02475 006263
26 02476 000005
27
28
29 02477 102400
30 02500 126400
31 02501 152400
32 02502 176440
33 02503 102000
34 02504 100410
35 02505 133015
36 02506 175014
37
38
39 02513 006264
40

      EXM:
      SETUP 10
      JSR @ENTIN
      10
      ZEROAC
      SUB 0,0
      SUB 1,1
      SUB 2,2
      SUBO 3,3
      ADC 3,3
      IOR 3,0
      IOR 3,1
      IOR 3,2
      MOV 0,0,SZR
      SUB# 0,3,SZR
      SUB# 1,0,SNR
      SUB# 2,0,SZR
      ERROR
      LOOP
      JSR @ENTLO

      ITERATE TEST ROUTINE

      TEST "IOR"
      INITIALIZE TEST.

      CLEAR ALL AC'S
      AND C(CARRY)

      IC(AC0,1,2)=0
      IC(AC3)=(-1)
      INCLUSIVE OR AC3 TO
      EVERY OTHER AC.
      AFTER IOR INSTRUCTIONS
      FALL AC'S SHOULD=-1
      WITH C(CARRY) UNCHANGED
      AT (1).

      ITERATE TEST ROUTINE

      SETUP 5
      JSR @ENTIN
      5
      AC0 IOR,0
      ZEROAC
      SUB 0,0
      SUB 1,1
      SUB 2,2
      SUBO 3,3
      ADC 0,0
      TOR 0,0
      ADD# 0*183,0*283,SNR
      MOV# 0*383,0*383,SZR
      ERROR
      LOOP
      JSR @ENTLO

      ITERATE TEST ROUTINE

      TEST "IOR"
      INITIALIZE TEST.

      CLEAR ALL AC'S
      AND C(CARRY)

      IC(AC0,1,2)=0
      IC(AC3)=(-1)
      INCLUSIVE OR AC3 TO
      EVERY OTHER AC.
      AFTER IOR INSTRUCTIONS
      FALL AC'S SHOULD=-1
      WITH C(CARRY) UNCHANGED
      AT (1).

      ITERATE TEST ROUTINE

      SETUP 5
      JSR @ENTIN
      5
      AC0 IOR,0
      ZEROAC
      SUB 0,0
      SUB 1,1
      SUB 2,2
      SUBO 3,3
      ADC 0,0
      TOR 0,0
      ADD# 0*183,0*283,SNR
      MOV# 0*383,0*383,SZR
      ERROR
      LOOP
      JSR @ENTLO

      ITERATE TEST ROUTINE

```

```

10049 ECL10
01
02 02514 006263
03 02515 000005
04
05 02516 102400
06 02517 126400
07 02518 152400
08 02519 176400
09 02520 152400
10 02521 176400
11 02522 126000
12 02523 124410
13 02524 157015
14 02525 101014
15
16 02532 006264
17
18
19
20 02533 006263
21 02534 000005
22
23 02535 102400
24 02536 126400
25 02537 152400
26 02540 176400
27 02541 152000
28 02542 150410
29 02543 153015
30 02544 125014
31
32 02551 006264
33
34
35 02552 006263
36 02553 000005
37
38
39 02554 102400
40 02555 126400
41 02556 152400
42 02557 176440
43 02558 176440
44 02561 174410
45 02562 107015
46 02563 151014
47
48
49
50 02570 006264
51

10050 ECL10
01
02 02571 006263
03 02572 000400
04
05 02573 006266
06 02574 115300
07 02575 111300
08 02576 105300
09 02577 100000
10 02600 117400
11 02601 116000
12 02602 100000
13 02603 110410
14 02604 156414
15
16 02611 006264
17
18
19
20
21
22 02612 006263
23 02613 000050
24
25 02614 006266
26 02615 152400
27 02616 176400
28 02617 126420
29 02620 104410
30 02621 130410
31 02622 154410
32 02623 164410
33 02624 130410
34 02625 154410
35 02626 106415
36 02627 112414
37 02630 000403
38 02631 101002
39 02632 116414
40
41
42 02637 006264
43
44
45 02640 006263
46 02641 000400
47
48 02642 006266
49 02643 105700
50 02644 131000
51 02645 114000
52 02646 173400
53 02647 172000
54 02650 104410
55 02651 132414
56
57
58 02656 006264
59

```

```

:TEST IOR
:INITIALIZE TEST.

SETUP 400
JSR @ENTIN
400
RAND
:C(AC0)=RANDOM #
:TRUTH TABLE FOR IOR
:0011
:0101
:(0111)
:GENERATE INCLUSIVE OR
:VIA SOFTWARE AND THE
:IOR INSTRUCTION
:IC(AC0)=ORIG ACS
:IC(AC1)=ORIG ACC
:IC(AC2)=IOR RESULT
:ITERATE TEST ROUTINE
LOOP
JSR @ENTLO

:TEST *IOR*
:INITIALIZE TEST.

SETUP 50
JSR @ENTIN
50
RAND
JSR @ENTRA
SUB 2,2
SUB 3,3
SUB7 1,1
IOR 0,1
IOR 1,2
IOR 2,3
IOR 3,1
IOR 1,2
IOR 2,3
SUB# 0,1,SNR
SUB# 0,2,SZR
JMP *3
MOV 0,0,SZC
SUB# 0,3,SZR
ERROR
LOOP
JSR @ENTLO

:TEST *IOR*
:INITIALIZE TEST.

SETUP 400
JSR @ENTIN
400
RAND
JSR @ENTRA
INCS 0,1
MOV 1,2
COM 0,3
AND 3,2
ADC 3,2
IOR 0,1
SUB# 1,2,SZR
ERROR
LOOP
JSR @ENTLO

:TEST *IOR*
:INITIALIZE TEST.

SETUP 400
JSR @ENTIN
400
RAND
JSR @ENTRA
INCS 0,1
MOV 1,2
COM 0,3
AND 3,2
ADC 3,2
IOR 0,1
SUB# 1,2,SZR
ERROR
LOOP
JSR @ENTLO

```

```

:TEST *IOR*
:INITIALIZE TEST.

SETUP 5
JSR @ENTIN
5
ACD IOR,1
ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUBO 3,3
ADC 1,1
:THE VALUE IN AC1
:SHOULD NOT EFFECT
IOR 1,1
ADD# 1+183,1+283,SNR :OTHER AC'S. CHECK
MOV# 1+383,1+383,SZR :AC DESTINATION FOR
:THE IOR INSTRUCTION.
ERROR
LOOP
JSR @ENTLO

:ITERATE TEST ROUTINE
LOOP
JSR @ENTLO

:TEST *IOR*
:INITIALIZE TEST.

SETUP 5
JSR @ENTIN
5
ACD IOR,2
ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUBO 3,3
ADC 2,2
:THE VALUE IN AC2
:SHOULD NOT EFFECT
IOR 2,2
ADD# 2+183,2+283,SNR :OTHER AC'S. CHECK
MOV# 2+383,2+383,SZR :AC DESTINATION FOR
:THE IOR INSTRUCTION.
ERROR
LOOP
JSR @ENTLO

:ITERATE TEST ROUTINE
LOOP
JSR @ENTLO

:TEST *IOR*
:INITIALIZE TEST.

SETUP 5
JSR @ENTIN
5
ACD IOR,3
ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUBO 3,3
ADC 3,3
:THE VALUE IN AC3
:SHOULD NOT EFFECT
IOR 3,3
ADD# 3+183,3+283,SNR :OTHER AC'S. CHECK
MOV# 3+383,3+383,SZR :AC DESTINATION FOR
:THE IOR INSTRUCTION.
ERROR
LOOP
JSR @ENTLO

:ITERATE TEST ROUTINE
LOOP
JSR @ENTLO

```

10051 ECL10

```

01
02
03
04
05
06
07 02657 006263
08 02660 000010
09 02661 102400
10 02662 126440
11 02663 104510
12 02664 101003
13 02665 125004
14
15
16 02672 006264
17
18
19 02673 006263
20 02674 000020
21
22
23 02675 102400
24 02676 126400
25 02677 152400
26 02700 176440
27 02701 126000
28 02702 104510
29 02703 157015
30 02704 101014
31
32
33
34 02711 102400
35 02712 126400
36 02713 152400
37 02714 176440
38 02715 152000
39 02716 110510
40 02717 163015
41 02720 125014
42
43
44
45 02725 102400
46 02726 126400
47 02727 152400
48 02730 176440
49 02731 176000
50 02732 114510
51 02733 107015
52 02734 151014
53
54
55 02741 006264

```

\*\*\* EXCLUSIVE OR \*\*\*

EXU:

SETUP 10  
JSR @ENTIN

```

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

```

EXX:

SETUP 5  
JSR @ENTIN

```

5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

```

10052 ECL10

```

01
02
03
04 02742 006263
05 02743 000005
06 02744 102040
07 02745 100510
08 02746 101002
09 02747 101004
10
11
12 02754 006264
13
14
15 02755 006263
16 02756 000005
17 02757 102400
18 02760 110040
19 02761 110510
20 02762 150014
21
22
23 02767 006264
24

```

EXX:

SETUP 5  
JSR @ENTIN

```

5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

```

10053 ECL10

```

01
02
03
04 02770 006263
05 02771 009400
06
07 02772 006266
08 02773 105300
09 02774 131000
10 02775 113520
11 02776 107000
12 02777 146400
13 03060 111300
14 03001 110510
15 03002 132414
16
17
18 03007 006264
19
20
21 03010 006263
22 03011 009005
23
24 03012 006266
25 03013 104700
26 03014 131300
27 03015 114000
28 03016 109310
29 03017 124510
30 03020 150510
31 03021 174510
32 03022 107015
33 03023 157014
34
35
36 03030 006264
37

```

```

EYX:
      SETUP 400
      JSR @ENTIN
      400
      RAND
      JSR @ENTRA
      MOVS 0,1
      MOV 1,2
      ANDZL 0,2
      ADD 0,1
      SUB 2,1
      MOVS 0,2
      XOR 0,2
      SUB# 1,2,SZR
      ERROR
      LOOP
      JSR @ENTLO

      SETUP 5
      JSR @ENTIN
      5
      RAND
      JSR @ENTRA
      NEG# 0,1
      MOVS 1,2
      COM 0,3
      XOR 0,0
      XOR 1,1
      XOR 2,2
      XOR 3,3
      ADD# 0,1,SNR
      ADD# 2,3,SZR
      ERROR
      LOOP
      JSR @ENTLO

```

```

:INITIALIZE TEST.
:TEST XOR
: C(A C0)=RANDOM #
:RAND XOR RAND
:SOFTWARE SIMULATION
: C(A C2)=XOR RESULT
: C(A C1)=CORRECT
: C(A C0)=ORIG OPERAND
: C(A C0) SWAPPED=OTHER OPERAND
:ITERATE TEST ROUTINE

:INITIALIZE TEST.
:USE A RANDOM NUMBER
: C(A C0)=RANDOM #
:IN EACH AC. XOR AC TO AC
:SHOULD PRODUCE 0 RESULT
:IN EACH AC.

:ALL ACS SHOULD BE 0.
:IF 0-2=0 THEN ACS FAILED.

:ITERATE TEST ROUTINE

```

10054 ECL10

```

01
02
03 03031 006263
04 03032 000100
05
06 03033 006266
07 03034 124510
08 03035 150510
09 03036 174510
10 03037 114510
11 03040 144510
12 03041 170510
13 03042 176000
14 03043 164510
15 03044 170510
16 03045 164510
17 03046 170510
18 03047 106415
19 03050 112414
20
21
22 03055 006264
23
24 03056 006263
25 03057 000100
26
27
28 03060 006266
29 03061 105300
30 03062 111000
31 -DD 16.
32 03063 130510
33 03064 130510
34 03065 130510
35 03066 130510
36 03067 130510
37 03070 130510
38 03071 130510
39 03072 130510
40 03073 130510
41 03074 130510
42 03075 130510
43 03076 130510
44 03077 130510
45 03100 130510
46 03101 130510
47 03102 130510
48 03103 112414
49
50
51 03110 006264
52

```

```

EX00:
      SETUP 100
      JSR @ENTIN
      100
      RAND
      JSR @ENTRA
      XOR 1,1
      XOR 2,2
      XOR 3,3
      XOR 0,3
      XOR 3,1
      XOR 3,2
      ADC 3,3
      XOR 3,1
      XOR 3,2
      SUB# 0,1,SNR
      SUB# 0,2,SZR
      ERROR
      LOOP
      JSR @ENTLO

      SETUP 100
      JSR @ENTIN
      100
      RAND
      JSR @ENTRA
      MOVS 0,1
      -DD 16.
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      SUB# 0,2,SZR
      ERROR
      LOOP
      JSR @ENTLO

      SETUP 100
      JSR @ENTIN
      100
      RAND
      JSR @ENTRA
      MOVS 0,1
      -DD 16.
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      SUB# 0,2,SZR
      ERROR
      LOOP
      JSR @ENTLO

      SETUP 100
      JSR @ENTIN
      100
      RAND
      JSR @ENTRA
      MOVS 0,1
      -DD 16.
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      XOR 1,2
      SUB# 0,2,SZR
      ERROR
      LOOP
      JSR @ENTLO

```

```

:TEST "XOR"
:INITIALIZE TEST.

: C(A C0)=RANDOM #
: C(A C0)=RANDOM
: C(A C0)=RANDOM
: C(A C1)=RANDOM
: C(A C1)=RANDOM
: C(A C2)=RANDOM
: C(A C3)=17777
: COMPLIMENT AND RECOMPLIMENT
: THE CONTENTS OF AC1 AND AC2.

: C(A C0)=ORIG NUMBER (CORRECT).
: XOR FAILED.

:ITERATE TEST ROUTINE

:TEST "XOR"
:INITIALIZE TEST.

: C(A C0)=RANDOM #

: SIXTEEN EXCLUSIVE OR INSTRUCTIONS
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: SHOULD NOT CHANGE THE RESULT
: C(A C2)=XOR RESULT
: C(A C1)=XOR ACS (ORIGINAL)
: ITERATE TEST ROUTINE

```

```

10055 ECL10
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

*** AND WITH COMPLEMENTED SOURCE ***

:TEST "XOR"
:INITIALIZE TEST.

SETUP 5
JSR @ENTIN
S
RAND
JSR @ENTRA
MOV 0,1
XOR 2,2
XOR 0,2
XOR 2,1
MOV 1,1,SZR
ERROR
LOOP
JSR @ENTLO

:ITERATE TEST ROUTINE

:TEST "XOR"
:INITIALIZE TEST.

SETUP 5
JSR @ENTIN
S
ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUBO 3,3
ANC 0,0
ANC 1,1
ANC 2,2
ANC 3,3
ADD# 0,1,SNR
ADD# 2,3,SZR
ERROR
LOOP
JSR @ENTLO

:ITERATE TEST ROUTINE

:TEST "XOR"
:INITIALIZE TEST.

SETUP 5
JSR @ENTIN
S
ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUBO 3,3
ADC 1,1
ANC 0,1
ADD# 1+1&3,1+2&3,SNR
MOV# 1+3&3,1+3&3,SZR
ERROR
ANCACD 2
ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUBO 3,3
ADC 2,2
ANC 0,2
ADD# 2+1&3,2+2&3,SNR
MOV# 2+3&3,2+3&3,SZR
ERROR
ANCACD 3
ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUBO 3,3
ADC 3,3
ANC 0,3
ADD# 3+1&3,3+2&3,SNR
MOV# 3+3&3,3+3&3,SZR
ERROR
LOOP
JSR @ENTLO

:ITERATE TEST ROUTINE

:TEST "XOR"
:INITIALIZE TEST.

SETUP 5
JSR @ENTIN
S
ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUBO 3,3
ADC 1,1
ANC 0,1
ADD# 1+1&3,1+2&3,SNR
MOV# 1+3&3,1+3&3,SZR
ERROR
ANCACD 2
ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUBO 3,3
ADC 2,2
ANC 0,2
ADD# 2+1&3,2+2&3,SNR
MOV# 2+3&3,2+3&3,SZR
ERROR
ANCACD 3
ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUBO 3,3
ADC 3,3
ANC 0,3
ADD# 3+1&3,3+2&3,SNR
MOV# 3+3&3,3+3&3,SZR
ERROR
LOOP
JSR @ENTLO

:ITERATE TEST ROUTINE

```

```

0057 ECL10
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

:TEST "ANC"
:INITIALIZE TEST.

SETUP 5
JSR @ENTIN
5
I=0
DO 4
ADC I,I
:ANC SHOULD PRODUCE 0
ADC I,I
:SET AC I TO (-1)
:SET AC I TO (-1)
:ANC SHOULD PRODUCE 0
ADC I,I
:SET AC I TO (-1)
:ANC SHOULD PRODUCE 0
ADC I,I
:SET AC I TO (-1)
:ANC SHOULD PRODUCE 0
ADC I,I
:SET AC I TO (-1)
:ANC SHOULD PRODUCE 0
ADD# 0,1,SNR
:ALL AC'S SHOULD BE 0.
ADD# 2,3,SZR
ERROR
LOOP
JSR @ENTLO

:ITERATE TEST ROUTINE

:TEST "ANC"
:INITIALIZE TEST.

SETUP 5
JSR @ENTIN
5
SUBZ 0,0
:1 TO 0 SHOULD PRODUCE
:RESULT WITH C(CARRY)
:UNCHANGED AT (1).
MOV 0,0,SZC
:IF A ERROR PRESENT
:JMP TO EX06.
ERROR
LOOP
JSR @ENTLO

:ITERATE TEST ROUTINE

EX05:
02 03216 006263
03 03217 000005
04 000000
05 000004
06 03220 102000
07 03231 100610
08 03252 126000
09 03223 124610
10 03224 152000
11 03225 150610
12 03226 176000
13 03227 174610
14 03230 107015
15 03231 157014
16
17 03236 006264
18
19
20
21 03237 006263
22 03240 000005
23 03241 176000
24 03242 102420
25 03243 160610
26 03244 101002
27 03245 101004
28
29 03252 006264
30
31
32
33

:0058 ECL10
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

:TEST "ANC"
:INITIALIZE TEST.

:0 TO 1 SHOULD
:PRODUCE RESULT OF 1.
: (CARRY) SHOULD REMAIN
:UNCHANGED AT (0).
:ITERATE TEST ROUTINE

SETUP 5
JSR @ENTIN
5
SUB 3,3
ADC 7,1
ANC 3,1
MOV# 1,1,SNR
COM 1,1,SZR
ERROR
LOOP
JSR @ENTLO

SUB 0,0
SETUP 10
JSR @ENTIN
MOVZ 0,3
ADC 2,2
ANC 3,2
COM 0,1
SUB# 2,1,SZR
ERROR
JMPER EX08.
LDA 3,1,TRR
MOV 3,3,SZR
JMP EX08.
INCS 0,0,SZR
JMP EX08
LOOP
JSR @ENTLO

:ITERATE TEST ROUTINE

:TEST "ANC"
:INITIALIZE TEST.

:C(AC3)=C(AC0)
:C(AC2)=17777
:AND COMPLEMENT C(AC3) TO C(AC2)
:C(AC0)=ORIGINAL
:C(AC1)=CORRECT
:C(AC2)=ANC RESULT
:IF A ERROR PRESENT
:JMP TO EX08.
:ITERATE TEST ROUTINE

EX07:
04 03253 006263
05 03254 000005
06 03255 176400
07 03256 126020
08 03257 164610
09 03260 125013
10 03261 124004
11
12
13 03266 006264
14
15 03267 102400
16
17 03270 006263
18 03271 000010
19 03272 115020
20 03273 152000
21 03274 170610
22 03275 104000
23 03276 146414
24
25
26 03303 034210
27 03304 175004
28 03305 000403
29 03306 101704
30 03307 000763
31
32 03310 006264
33

EX08:
04 03273 152000
05 03274 170610
06 03275 104000
07 03276 146414
08
09
10 03303 034210
11 03304 175004
12 03305 000403
13 03306 101704
14 03307 000763
15
16 03310 006264
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33

```

```

10059 ECL10
01
02 03311 006263
03 03312 000400
04
05 03313 006266
06 03314 104300
07 03315 114000
08 03316 131000
09 03317 173400
10 03320 104610
11 03321 132414
12
13
14 03326 006264
15
16 03327 126400
17
18 03330 006263
19 03331 000010
20 03332 150610
21 03333 130610
22 03334 151004
23
24
25 03341 030210
26 03342 173004
27 03343 000403
28 03344 125704
29 03345 000765
30
31 03346 006264
32

10060 ECL10
01
02
03 03347 006263
04 03350 000400
05
06
07 03351 006266
08 03352 115040
09 03353 124610
10 03354 164610
11 03355 134610
12 03356 152000
13 03357 170610
14 03360 154610
15 03361 126000
16 03362 164610
17 03363 152000
18 03364 130610
19 03365 101002
20 03366 112414
21
22
23 03373 006264
24
25
26 03374 006263
27 03375 000100
28
29 03376 006266
30 03377 104000
31 03400 110000
32 03401 120610
33 03402 144610
34 03403 124000
35 03404 112415
36 03405 132414
37
38
39 03412 006264
40

EX09:
SETUP 400
JSR @ENTIN
400
RAND
JSR @ENTRA
COM# 0,1
COM# 0,3
MOV 1,2
AND 3,2
ANC 0,1
SUB# 1,2,SZR
ERROR
LOOP
JSR @ENTLO
14
SUB 1,1
SETUP 10
JSR @ENTIN
10
EX0A:
ANC 2,2
MOV 2,2,SZR
ERROR
JMPER EX0A.
LDA 3,1,TRR
MOV 3,3,SZR
JMP EX0A.
INCS 1,1,SZR
JMP EX0A
EX0A.1
LOOP
JSR @ENTLO
14
SUB 1,1
SETUP 10
JSR @ENTIN
10
EX0B:
SETUP 400
JSR @ENTIN
400
RAND
JSR @ENTRA
MOV# 0,3
ANC 1,1
ANC 3,1
ANC 1,3
ANC 2,2
ANC 3,2
ANC 2,3
ANC 1,1
ANC 3,1
ANC 2,2
ANC 1,2
MOV# 0,2,SZR
SUB# 0,2,SZR
ERROR
LOOP
JSR @ENTLO
24
SETUP 100
JSR @ENTIN
100
RAND
JSR @ENTRA
COM# 0,1
MOV# 0,2
ANC 1,0
ANC 2,1
COM# 1,1
SUB# 0,2,SNR
SUB# 1,2,SZR
ERROR
LOOP
JSR @ENTLO
24
SETUP 100
JSR @ENTIN
100
RAND
JSR @ENTRA
COM# 0,1
MOV# 0,2
ANC 1,0
ANC 2,1
COM# 1,1
SUB# 0,2,SNR
SUB# 1,2,SZR
ERROR
LOOP
JSR @ENTLO
24

!TEST "ANC"
!INITIALIZE TEST.
!C(AC0)=RANDOM #
!C(AC1)=0
!C(AC1)=0
!C(AC3)=UNCHANGED=C(AC0)
!C(AC2)=17777
!C(AC2)=C(AC0) COMPLEMENTED
!C(AC3)=UNCHANGED=C(AC0)
!C(AC1)=17777
!C(AC1)=C(AC0) COMPLEMENTED
!C(AC2)=17777
!C(AC2)=C(AC0)
!C(CARRY) SHOULD REMAIN (1).
!C(AC0)=ORIGINAL AND CORRECT
!C(AC2)=FINAL ANC RESULT
!ITERATE TEST ROUTINE

!TEST "ANC"
!INITIALIZE TEST.
!C(AC0)=RANDOM #
!C(AC1)=C(AC0) COMPLEMENT
!C(AC2)=C(AC0) UNCHANGED.
!C(AC1)=UNCHANGED
!C(AC1)=C(AC2)
!C(AC2)=ORIGINAL #
!C(AC0)=ANC RESULT
!C(AC1)=ANC RESULT
!ACO-2 SHOULD BE SAME #
!ITERATE TEST ROUTINE

```



```

10061 ECL10
01 **** EXCHANGE ACCUMULATORS ****
02
03
04
05 EX00: 10061 ECL10
06 JSR @ENTIN
07 ZER0AC
08 SUB 0,0
09 SUB 1,1
10 SUB 2,2
11 SUB0 3,3
12 XCH 0,0
13 XCH 1,1
14 XCH 2,2
15 XCH 3,3
16 ADD# 0,1-SNR
17 ADD# 2,3-SZR
18 ERROR
19 LOOP
20 JSR @ENTLO
21
22 EX0E: 10061 ECL10
23 JSR @ENTIN
24 JSR @ENTIN
25 I=0
26 I=0 4
27 ADC I,1
28 XCH I,1
29 SET AC I TO 17777
30 XCH I,1
31 ADC I,1
32 XCH I,1
33 SET AC I TO 17777
34 ADC I,1
35 XCH I,1
36 XCH I,1
37 COM# 3,3-SNR
38 COM# 1,1-SZR
39 ERROR
40 COM# 2,2-SNR
41 COM# 0,0-SZR
42 ERROR
43 LOOP
44 JSR @ENTLO
45
46 EX0F: 10061 ECL10
47 JSR @ENTIN
48 I=0
49 SUB 0,0
50 ADC 1,1
51 XCH 0,1
52 COM# 0,0-SNR
53 MOV# 1,1-SZR
54 ERROR
55 LOOP
56 JSR @ENTLO
57
10062 ECL10
01
02 EX0G: 10062 ECL10
03 JSR @ENTIN
04 XCHACD 0
05 ZER0AC
06 SUB 0,0
07 SUB 1,1
08 SUB 2,2
09 SUB0 3,3
10 XCH 0,0
11 XCH 1,1
12 XCH 2,2
13 MOV# 0+183,0+283-SNR
14 MOV# 0+383,0+383-SZR
15 ERROR
16 XCHACD 1
17 ZER0AC
18 SUB 0,0
19 SUB 1,1
20 SUB 2,2
21 SUB0 3,3
22 XCH 1,1
23 ADD# 1+183,1+283-SNR
24 MOV# 1+383,1+383-SZR
25 ERROR
26 XCHACD 2
27 ZER0AC
28 SUB 0,0
29 SUB 1,1
30 SUB 2,2
31 SUB0 3,3
32 XCH 2,2
33 ADC 2,2
34 XCH 2,2
35 ADD# 2+183,2+283-SNR
36 MOV# 2+383,2+383-SZR
37 ERROR
38 XCHACD 3
39 ZER0AC
40 SUB 0,0
41 SUB 1,1
42 SUB 2,2
43 SUB0 3,3
44 ADC 3,3
45 XCH 3,3
46 ADD# 3+183,3+283-SNR
47 MOV# 3+383,3+383-SZR
48 ERROR
49 LOOP
50 JSR @ENTLO
51 ITERATE TEST ROUTINE

```









|             |             |       |       |
|-------------|-------------|-------|-------|
| 10071 ECL10 | 10072 ECL10 | Ex02: | Ex02: |
| 01          | 01          | 02    | 02    |
| 02          | 02          | 03    | 03    |
| 03          | 03          | 04    | 04    |
| 04          | 04          | 05    | 05    |
| 05          | 05          | 06    | 06    |
| 06          | 06          | 07    | 07    |
| 07          | 07          | 08    | 08    |
| 08          | 08          | 09    | 09    |
| 09          | 09          | 10    | 10    |
| 10          | 10          | 11    | 11    |
| 11          | 11          | 12    | 12    |
| 12          | 12          | 13    | 13    |
| 13          | 13          | 14    | 14    |
| 14          | 14          | 15    | 15    |
| 15          | 15          | 16    | 16    |
| 16          | 16          | 17    | 17    |
| 17          | 17          | 18    | 18    |
| 18          | 18          | 19    | 19    |
| 19          | 19          | 20    | 20    |
| 20          | 20          | 21    | 21    |
| 21          | 21          | 22    | 22    |
| 22          | 22          | 23    | 23    |
| 23          | 23          | 24    | 24    |
| 24          | 24          | 25    | 25    |
| 25          | 25          | 26    | 26    |
| 26          | 26          | 27    | 27    |
| 27          | 27          | 28    | 28    |
| 28          | 28          | 29    | 29    |
| 29          | 29          | 30    | 30    |
| 30          | 30          | 31    | 31    |
| 31          | 31          | 32    | 32    |
| 32          | 32          | 33    | 33    |
| 33          | 33          | 34    | 34    |
| 34          | 34          | 35    | 35    |
| 35          | 35          | 36    | 36    |
| 36          | 36          | 37    | 37    |
| 37          | 37          | 38    | 38    |
| 38          | 38          | 39    | 39    |
| 39          | 39          | 40    | 40    |
| 40          | 40          | 41    | 41    |
| 41          | 41          | 42    | 42    |
| 42          | 42          | 43    | 43    |
| 43          | 43          | 44    | 44    |
| 44          | 44          | 45    | 45    |
| 45          | 45          | 46    | 46    |
| 46          | 46          | 47    | 47    |
| 47          | 47          | 48    | 48    |
| 48          | 48          | 49    | 49    |
| 49          | 49          |       |       |

|              |              |              |              |
|--------------|--------------|--------------|--------------|
| SETUP 50     | SETUP 50     | SETUP 50     | SETUP 50     |
| JSR @ENTIN   | JSR @ENTIN   | JSR @ENTIN   | JSR @ENTIN   |
| 50           | 50           | 50           | 50           |
| RAND         | RAND         | RAND         | RAND         |
| JSR @ENTRA   | JSR @ENTRA   | JSR @ENTRA   | JSR @ENTRA   |
| MOVZ 0,2     | MOVZ 0,2     | MOVZ 0,2     | MOVZ 0,2     |
| ADC 1,1      | ADC 1,1      | ADC 1,1      | ADC 1,1      |
| LSH 1,0      | LSH 1,0      | LSH 1,0      | LSH 1,0      |
| MOV 0,0,SZC  | MOV 0,0,SZC  | MOV 0,0,SZC  | MOV 0,0,SZC  |
| SUB# 0,2,SZR | SUB# 0,2,SZR | SUB# 0,2,SZR | SUB# 0,2,SZR |
| ERROR        | ERROR        | ERROR        | ERROR        |
| LOOP         | LOOP         | LOOP         | LOOP         |
| JSR @ENTLO   | JSR @ENTLO   | JSR @ENTLO   | JSR @ENTLO   |
| 5            | 5            | 5            | 5            |
| ADCO 0,0,SKP | ADCO 0,0,SKP | ADCO 0,0,SKP | ADCO 0,0,SKP |
| 16,          | 16,          | 16,          | 16,          |
| LDA 2,-1     | LDA 2,-1     | LDA 2,-1     | LDA 2,-1     |
| LSH 2,0      | LSH 2,0      | LSH 2,0      | LSH 2,0      |
| MOV 0,0,SZC  | MOV 0,0,SZC  | MOV 0,0,SZC  | MOV 0,0,SZC  |
| MOV 0,0,SZR  | MOV 0,0,SZR  | MOV 0,0,SZR  | MOV 0,0,SZR  |
| ERROR        | ERROR        | ERROR        | ERROR        |
| LOOP         | LOOP         | LOOP         | LOOP         |
| JSR @ENTLO   | JSR @ENTLO   | JSR @ENTLO   | JSR @ENTLO   |
| 50           | 50           | 50           | 50           |
| RAND         | RAND         | RAND         | RAND         |
| JSR @ENTRA   | JSR @ENTRA   | JSR @ENTRA   | JSR @ENTRA   |
| LDA 2,-2     | LDA 2,-2     | LDA 2,-2     | LDA 2,-2     |
| ANDS 0,2,SKP | ANDS 0,2,SKP | ANDS 0,2,SKP | ANDS 0,2,SKP |
| 177400       | 177400       | 177400       | 177400       |
| MOV 0,1,SKP  | MOV 0,1,SKP  | MOV 0,1,SKP  | MOV 0,1,SKP  |
| -8,          | -8,          | -8,          | -8,          |
| LDA 0,-1     | LDA 0,-1     | LDA 0,-1     | LDA 0,-1     |
| LSH 0,1      | LSH 0,1      | LSH 0,1      | LSH 0,1      |
| SUB# 1,2,SZR | SUB# 1,2,SZR | SUB# 1,2,SZR | SUB# 1,2,SZR |
| ERROR        | ERROR        | ERROR        | ERROR        |
| LOOP         | LOOP         | LOOP         | LOOP         |
| JSR @ENTLO   | JSR @ENTLO   | JSR @ENTLO   | JSR @ENTLO   |
| 06266        | 06266        | 06266        | 06266        |
| 06263        | 06263        | 06263        | 06263        |
| 00005        | 00005        | 00005        | 00005        |
| 102041       | 102041       | 102041       | 102041       |
| 000020       | 000020       | 000020       | 000020       |
| 101210       | 101210       | 101210       | 101210       |
| 101002       | 101002       | 101002       | 101002       |
| 101004       | 101004       | 101004       | 101004       |
| 06264        | 06264        | 06264        | 06264        |
| 06263        | 06263        | 06263        | 06263        |
| 000050       | 000050       | 000050       | 000050       |
| 06266        | 06266        | 06266        | 06266        |
| 030402       | 030402       | 030402       | 030402       |
| 113701       | 113701       | 113701       | 113701       |
| 177400       | 177400       | 177400       | 177400       |
| 105041       | 105041       | 105041       | 105041       |
| 17770        | 17770        | 17770        | 17770        |
| 020777       | 020777       | 020777       | 020777       |
| 105210       | 105210       | 105210       | 105210       |
| 132410       | 132410       | 132410       | 132410       |
| 06264        | 06264        | 06264        | 06264        |

|                              |                              |                              |                              |
|------------------------------|------------------------------|------------------------------|------------------------------|
| :TEST "LSH"                  | :TEST "LSH"                  | :TEST "LSH"                  | :TEST "LSH"                  |
| :INITIALIZE TEST.            | :INITIALIZE TEST.            | :INITIALIZE TEST.            | :INITIALIZE TEST.            |
| PC(AC0)=RANDOM #             | PC(AC0)=RANDOM #             | PC(AC0)=RANDOM #             | PC(AC0)=RANDOM #             |
| :(-1), SHIFT RIGHT 1 PLACE   | :(-1), SHIFT RIGHT 1 PLACE   | :(-1), SHIFT RIGHT 1 PLACE   | :(-1), SHIFT RIGHT 1 PLACE   |
| :FORM CORRECT                | :FORM CORRECT                | :FORM CORRECT                | :FORM CORRECT                |
| :SHIFT                       | :SHIFT                       | :SHIFT                       | :SHIFT                       |
| :C(CARRY) SHOULD NOT CHANGE  | :C(CARRY) SHOULD NOT CHANGE  | :C(CARRY) SHOULD NOT CHANGE  | :C(CARRY) SHOULD NOT CHANGE  |
| :C(AC2)=CORRECT              | :C(AC2)=CORRECT              | :C(AC2)=CORRECT              | :C(AC2)=CORRECT              |
| :C(AC1)=SHIFT COUNT          | :C(AC1)=SHIFT COUNT          | :C(AC1)=SHIFT COUNT          | :C(AC1)=SHIFT COUNT          |
| :C(AC0)=LSH RESULT           | :C(AC0)=LSH RESULT           | :C(AC0)=LSH RESULT           | :C(AC0)=LSH RESULT           |
| :ITERATE TEST ROUTINE        | :ITERATE TEST ROUTINE        | :ITERATE TEST ROUTINE        | :ITERATE TEST ROUTINE        |
| 16                           | 16                           | 16                           | 16                           |
| :TEST "LSH"                  | :TEST "LSH"                  | :TEST "LSH"                  | :TEST "LSH"                  |
| :INITIALIZE TEST.            | :INITIALIZE TEST.            | :INITIALIZE TEST.            | :INITIALIZE TEST.            |
| :SET C(AC0) AND C(CARRY)     | :SET C(AC0) AND C(CARRY)     | :SET C(AC0) AND C(CARRY)     | :SET C(AC0) AND C(CARRY)     |
| PC(AC2)=SHIFT COUNT OF 16    | PC(AC2)=SHIFT COUNT OF 16    | PC(AC2)=SHIFT COUNT OF 16    | PC(AC2)=SHIFT COUNT OF 16    |
| :SHIFTING SHOULD PRODUCE     | :SHIFTING SHOULD PRODUCE     | :SHIFTING SHOULD PRODUCE     | :SHIFTING SHOULD PRODUCE     |
| :A ZERO RESULT WITH C(CARRY) | :A ZERO RESULT WITH C(CARRY) | :A ZERO RESULT WITH C(CARRY) | :A ZERO RESULT WITH C(CARRY) |
| :UNCHANGED AT (1)            | :UNCHANGED AT (1)            | :UNCHANGED AT (1)            | :UNCHANGED AT (1)            |
| :ITERATE TEST ROUTINE        | :ITERATE TEST ROUTINE        | :ITERATE TEST ROUTINE        | :ITERATE TEST ROUTINE        |
| 000040                       | 000040                       | 000040                       | 000040                       |
| :TEST "LSH"                  | :TEST "LSH"                  | :TEST "LSH"                  | :TEST "LSH"                  |
| :INITIALIZE TEST.            | :INITIALIZE TEST.            | :INITIALIZE TEST.            | :INITIALIZE TEST.            |
| PC(AC0)=RANDOM #             | PC(AC0)=RANDOM #             | PC(AC0)=RANDOM #             | PC(AC0)=RANDOM #             |
| :THE LEFT 8 BITS OF          | :THE LEFT 8 BITS OF          | :THE LEFT 8 BITS OF          | :THE LEFT 8 BITS OF          |
| :ACO GO TO THE RIGHT 8       | :ACO GO TO THE RIGHT 8       | :ACO GO TO THE RIGHT 8       | :ACO GO TO THE RIGHT 8       |
| :BITS OF AC2                 | :BITS OF AC2                 | :BITS OF AC2                 | :BITS OF AC2                 |
| :SHIFT RIGHT 8 PLACES        | :SHIFT RIGHT 8 PLACES        | :SHIFT RIGHT 8 PLACES        | :SHIFT RIGHT 8 PLACES        |
| :C(CARRY)=LSH RESULT         | :C(CARRY)=LSH RESULT         | :C(CARRY)=LSH RESULT         | :C(CARRY)=LSH RESULT         |
| :C(AC2)=CORRECT              | :C(AC2)=CORRECT              | :C(AC2)=CORRECT              | :C(AC2)=CORRECT              |
| :ITERATE TEST ROUTINE        | :ITERATE TEST ROUTINE        | :ITERATE TEST ROUTINE        | :ITERATE TEST ROUTINE        |
| 000200                       | 000200                       | 000200                       | 000200                       |
| :TEST "LSH"                  | :TEST "LSH"                  | :TEST "LSH"                  | :TEST "LSH"                  |
| :INITIALIZE TEST.            | :INITIALIZE TEST.            | :INITIALIZE TEST.            | :INITIALIZE TEST.            |
| PC(AC0)=RANDOM #             | PC(AC0)=RANDOM #             | PC(AC0)=RANDOM #             | PC(AC0)=RANDOM #             |
| :IN BITS 0-7 (HIGH ORDER)    | :IN BITS 0-7 (HIGH ORDER)    | :IN BITS 0-7 (HIGH ORDER)    | :IN BITS 0-7 (HIGH ORDER)    |
| :SHIFTED LEFT.               | :SHIFTED LEFT.               | :SHIFTED LEFT.               | :SHIFTED LEFT.               |
| :C(AC3)=CORRECT              | :C(AC3)=CORRECT              | :C(AC3)=CORRECT              | :C(AC3)=CORRECT              |
| :C(AC0)=ORIGINAL NUMBER.     | :C(AC0)=ORIGINAL NUMBER.     | :C(AC0)=ORIGINAL NUMBER.     | :C(AC0)=ORIGINAL NUMBER.     |
| :ITERATE TEST ROUTINE        | :ITERATE TEST ROUTINE        | :ITERATE TEST ROUTINE        | :ITERATE TEST ROUTINE        |
| 006264                       | 006264                       | 006264                       | 006264                       |
| 006263                       | 006263                       | 006263                       | 006263                       |
| 000020                       | 000020                       | 000020                       | 000020                       |
| 102001                       | 102001                       | 102001                       | 102001                       |
| 024777                       | 024777                       | 024777                       | 024777                       |
| 121210                       | 121210                       | 121210                       | 121210                       |
| 101004                       | 101004                       | 101004                       | 101004                       |
| 000100                       | 000100                       | 000100                       | 000100                       |
| 102001                       | 102001                       | 102001                       | 102001                       |
| 000100                       | 000100                       | 000100                       | 000100                       |
| 024777                       | 024777                       | 024777                       | 024777                       |
| 121210                       | 121210                       | 121210                       | 121210                       |
| 101004                       | 101004                       | 101004                       | 101004                       |
| 000200                       | 000200                       | 000200                       | 000200                       |
| 006264                       | 006264                       | 006264                       | 006264                       |

```

10075 ECL10
01
02
03
04 04477 006263
05 04500 000050
06
07 04501 006266
08 04502 176520
09 04503 111001
10 04504 000377
11 04505 024777
12 04506 107700
13 04507 000010
14 04507 171210
15 04510 171210
16 04511 171210
17 04512 171210
18 04513 171210
19 04514 171210
20 04515 171210
21 04516 171210
22 04517 132414
23
24 04524 006264
25
26
27 04525 006263
28 04526 000005
29
30
31
32 04527 102400
33 04530 126400
34 04531 152400
35 04532 176440
36 04533 102000
37 04534 101210
38 04535 133015
39 04536 175014
40
41
42
43 04538 102400
44 04544 126400
45 04545 152400
46 04546 176440
47 04547 126000
48 04550 125210
49 04551 157015
50 04552 101014
51
52
53
54 04557 102400
55 04560 126400
56 04561 152400
57 04562 176440
58 04563 152000
59 04564 151210
60 04565 163015

EX1A:
      SETUP 50
      JSR @ENTIN
      RAND
      JSR @ENTRA
      SUB#L 5,3
      MOV @I,2,S,KP
      LDA 1,-1
      ANDS 0,-1
      .DO 8.
      LSH 3,2
      LSH 3,2
      LSH 3,2
      LSH 3,2
      LSH 3,2
      LSH 3,2
      LSH 3,2
      LSH 3,2
      SUB# 1,-2,-SZR
      ERROR
      JSR @ENTLO
      SETUP 5
      JSR @ENTIN
      LSHACD 0
      ZEROAC
      SUB 0,0
      SUB 1,1
      SUB 2,2
      SUBO 3,3
      ADC 0,0
      LSH 0,0
      ADD# 0+183,0+283,SNR
      MOV# 0+383,0+383,SZR
      ERROR
      LSHACD 1
      ZEROAC
      SUB 0,0
      SUB 1,1
      SUB 2,2
      SUBO 3,3
      ADC 1,1
      LSH 1,1
      ADD# 1+183,1+283,SNR
      MOV# 1+383,1+383,SZR
      ERROR
      LSHACD 2
      ZEROAC
      SUB 0,0
      SUB 1,1
      SUB 2,2
      SUBO 3,3
      ADC 2,2
      LSH 2,2
      ADD# 2+183,2+283,SNR
      ERROR

      !TEST "LSH"
      !INITIALIZE TEST.
      !C(CAC0)=RANDOM #
      !SHIFT COUNT=1 LEFT
      !C(CAC2) GETS RANDOM
      !C(CAC1) WILL GET #
      !LSB OF RANDOM
      !*8=8 SHIFTS
      !*8=8 SHIFTS
      !*8=8 SHIFTS
      !*8=8 SHIFTS
      !*8=8 SHIFTS
      !*8=8 SHIFTS
      !*8=8 SHIFTS
      !*8=8 SHIFTS
      !C(CAC0)=ORIGINAL NUMBER
      !C(CAC1)=CORRECT
      !C(CAC2)=LSH RESULT
      !ITERATE TEST ROUTINE
      !INITIALIZE TEST.
      !CLEAR ALL AC'S
      !AND C(CARRY)
      !TEST THAT LSH RESULT
      !GOES ONLY TO C(CAC 0)
      !C(CAC1)=1+283,SNR
      !MOV# 1+383,1+383,SZR
      !ERROR
      !LSHACD 1
      !ZEROAC
      !SUB 0,0
      !SUB 1,1
      !SUB 2,2
      !SUBO 3,3
      !ADC 1,1
      !LSH 1,1
      !ADD# 1+183,1+283,SNR
      !MOV# 1+383,1+383,SZR
      !ERROR
      !LSHACD 2
      !ZEROAC
      !SUB 0,0
      !SUB 1,1
      !SUB 2,2
      !SUBO 3,3
      !ADC 2,2
      !LSH 2,2
      !ADD# 2+183,2+283,SNR
      !TEST THAT LSH RESULT
      !GOES ONLY TO C(CAC 2)
      !ERROR

0074 ECL10
01 04566 125014
02
03
04
05 04573 102400
06 04574 126400
07 04575 152400
08 04576 176440
09 04577 176000
10 04600 175210
11 04601 107015
12 04602 151014
13
14
15 04607 006264
16
17
18 04610 006263
19 04611 000010
20 04612 101001
21 04613 177760
22 04614 020777
23 04615 028776
24 04616 030775
25 04617 034774
26 04620 101210
27 04621 125210
28 04622 151210
29 04623 175210
30 04624 173015
31 04625 107014
32
33
34 04632 006264
35

EX1C:
      SETUP 10
      JSR @ENTIN
      MOV 0,0,S,KP
      LDA 0,-1
      LDA 1,-2
      LDA 2,-3
      LDA 3,-4
      LSH 0,0
      LSH 1,1
      LSH 2,2
      LSH 3,3
      ADD# 3,2,SNR
      ADD# 0,1,SZR
      ERROR
      JSR @ENTLO
      LOOP
      !ITERATE TEST ROUTINE
      !TEST "LSH"
      !INITIALIZE TEST.
      !MOV 0,0,S,KP
      !-16.
      !SHIFT 16 TIMES RIGHT.
      !LOAD ALL AC'S WITH
      !SHIFT COUNT
      !ANY NUMBER SHIFTED 16 TIMES
      !RIGHT SHOULD PRODUCE A
      !ZERO RESULT. ANY AC NOT
      !ZERO FAILED. WITH ITS
      !LSH
      !ADD# 3,2,SNR
      !ADD# 0,1,SZR
      !ERROR
      JSR @ENTLO
      LOOP
      !ITERATE TEST ROUTINE

```

```

10075 ECL10
01
02
03
04 04633 006263
05 04634 000010
06 04635 020402
07 04636 105041
08 04637 000017
09 04640 131000
10 LDA 0,+2
11 MOV 0,1,SKP
12 MOV 1,2
13 MOV 1,2
14 MOV 0,1,SNR
15 MOV 1,1,SNR
16 04652 176620
17 04653 156415
18 04654 156414
19
20
21 04661 006264
22
23
24
25
26 04662 006263
27 04663 000050
28
29 04660 006266
30 04665 024402
31 04666 107401
32 04667 000037
33 04670 115000
34 04671 135210
35 04672 130405
36 04673 000404
37 04674 103000
38 04675 151404
39 04676 000776
40 04677 171000
41 04700 112414
42
43
44
45

:TEST "LSH"
:INITIALIZE TEST.

LOAD EACH AC (0-2)
WITH A SHIFT COUNT
OF 17 (LEFT 15 DECIMAL)
PLACES.
THE RESULT OF THIS
SHIFT SHOULD LEAVE ONLY
THE SIGN BIT SET

(CARRY) SHOULD REMAIN
UNCHANGED AT (1)
(CORRECT VALUE=100000

ITERATE TEST ROUTINE

:TEST "LSH"
:INITIALIZE TEST.

(CAC0)=RANDOM #
THE LSH COMMAND IS
GIVEN A RANDOM (LEFT)
SHIFT COUNT OF 0 TO 37
OCTAL

LEFT SHIFT IS SIMULATED
BY ADDITION.

(CAC0)=CORRECT RESULT
(CAC2)=LSH RESULT
(CAC1)=SHIFT COUNT
ITERATE TEST ROUTINE

10076 ECL10
01
02 04706 006263
03 04707 000100
04
05 04710 006266
06 04711 030402
07 04712 113701
08 04713 017400
09 04714 176000
10 04715 104405
11 04716 000404
12 04717 175120
13 04720 125404
14 04721 000776
15 04722 117400
16 04723 150400
17 04724 105000
18 04725 145210
19 04726 150400
20 04727 145210
21 04730 166414
22
23
24 04735 006264
25
26
27

SETUP 100
JSR @ENTLO
100
RAND
JSR @ENTRA
LDA 2,+2
ANDS 0,2,SKP
17400
ADC 3,3
NEG 2,1,SNR
JMP ,+4
MOVZL 3,3
INC 1,1,SNR
AND 0,3
NEG 2,2
MOV 0,1
LSH 2,1
NEG 2,2
LSH 2,1
SUB# 3,1,SNR
ERROR
LOOP
JSR @ENTLO

EX10:
SETUP 10
JSR @ENTIN
10
LDA 0,+2
MOV 0,1,SKP
17
MOV 1,2
LSH 0,0
LSH 1,1
LSH 1,1
LSH 2,2
SUB# 0,1,SNR
MOV 0,1,SNR
ERROR
SUBZR 3,3
SUB# 1,3,SNR
SUB# 2,3,SNR
ERROR
LOOP
JSR @ENTLO

EX1E:
SETUP 50
JSR @ENTIN
50
RAND
JSR @ENTRA
LDA 1,+2
AND 0,1,SKP
37
MOV 0,3
LSH 1,3
NEG 1,2,SNR
JMP ,+4
ADD 0,0
INC 2,2,SNR
JMP ,+2
MOV 3,2
SUB# 0,2,SNR
ERROR
LOOP
JSR @ENTLO

:TEST "LSH"
:INITIALIZE TEST.

(CAC0)=RANDOM #
(CAC1)=LSH RESULT
(CAC2)=SHIFT COUNT
(CAC3)=CORRECT
ITERATE TEST ROUTINE

```



```

10077 ECL10
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

!TEST "LSH"
!INITIALIZE TEST.
:C(AC0)=RANDOM #
:RAND
JSR @ENTRA
MOV 0,1,SKP
LDA 0,-1
MOV 1,2
MOV 2,3
LSH 0,2
LSH 0,3
ADDZ 1,1
ADDZ 1,1
SUB# 1,3,SNR
SUB# 1,2,SZR
ERROR
LOOP
JSR @ENTLO

!ITERATE TEST ROUTINE
:TEST "DLSH"
!INITIALIZE TEST.
:C(AC0) ARE BOTH
!Cleared
!SHIFT 0 PLACES
!EXPECT RESULT
ADD# 0,1,SZR
ERROR !OF BOTH AC'S TO BE ZERO
LOOP
JSR @ENTLO

!ITERATE TEST ROUTINE
:TEST "DLSH"
!INITIALIZE TEST.
:C(AC0)=RANDOM #
:RAND
JSR @ENTRA
MOV 0,2
MOVO 0,1,SKP
LDA 0,-1
!C(AC0)=LEFT SHIFT 16
!C(AC1)=RANDOM DATA
!SHOULD PRODUCE 0 RESULT
!IN C(AC1) AND ORIG RANDOM
!IN C(AC0)
SUB# 0,2,SZR
ERROR
LOOP
JSR @ENTLO

!ITERATE TEST ROUTINE

10078 ECL10
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

!TEST "LSH"
!INITIALIZE TEST.
:C(AC0)=RANDOM #
:RAND
JSR @ENTRA
MOV 0,1,SKP
LDA 0,-1
MOV 1,2
MOV 2,3
LSH 0,2
LSH 0,3
ADDZ 1,1
ADDZ 1,1
SUB# 1,3,SNR
SUB# 1,2,SZR
ERROR
LOOP
JSR @ENTLO

!ITERATE TEST ROUTINE
:TEST "DLSH"
!INITIALIZE TEST.
:C(AC0) ARE BOTH
!Cleared
!SHIFT 0 PLACES
!EXPECT RESULT
ADD# 0,1,SZR
ERROR !OF BOTH AC'S TO BE ZERO
LOOP
JSR @ENTLO

!ITERATE TEST ROUTINE
:TEST "DLSH"
!INITIALIZE TEST.
:C(AC0)=RANDOM #
:RAND
JSR @ENTRA
MOV 0,2
MOVO 0,1,SKP
LDA 0,-1
!C(AC0)=LEFT SHIFT 16
!C(AC1)=RANDOM DATA
!SHOULD PRODUCE 0 RESULT
!IN C(AC1) AND ORIG RANDOM
!IN C(AC0)
SUB# 0,2,SZR
ERROR
LOOP
JSR @ENTLO

!ITERATE TEST ROUTINE

10079 ECL10
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

!TEST "LSH"
!INITIALIZE TEST.
:C(AC0)=RANDOM #
:RAND
JSR @ENTRA
MOV 0,1,SKP
LDA 0,-1
MOV 1,2
MOV 2,3
LSH 0,2
LSH 0,3
ADDZ 1,1
ADDZ 1,1
SUB# 1,3,SNR
SUB# 1,2,SZR
ERROR
LOOP
JSR @ENTLO

!ITERATE TEST ROUTINE
:TEST "DLSH"
!INITIALIZE TEST.
:C(AC0) ARE BOTH
!Cleared
!SHIFT 0 PLACES
!EXPECT RESULT
ADD# 0,1,SZR
ERROR !OF BOTH AC'S TO BE ZERO
LOOP
JSR @ENTLO

!ITERATE TEST ROUTINE
:TEST "DLSH"
!INITIALIZE TEST.
:C(AC0)=RANDOM #
:RAND
JSR @ENTRA
MOV 0,2
MOVO 0,1,SKP
LDA 0,-1
!C(AC0)=LEFT SHIFT 16
!C(AC1)=RANDOM DATA
!SHOULD PRODUCE 0 RESULT
!IN C(AC1) AND ORIG RANDOM
!IN C(AC0)
SUB# 0,2,SZR
ERROR
LOOP
JSR @ENTLO

!ITERATE TEST ROUTINE

10080 ECL10
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

!TEST "LSH"
!INITIALIZE TEST.
:C(AC0)=RANDOM #
:RAND
JSR @ENTRA
MOV 0,1,SKP
LDA 0,-1
MOV 1,2
MOV 2,3
LSH 0,2
LSH 0,3
ADDZ 1,1
ADDZ 1,1
SUB# 1,3,SNR
SUB# 1,2,SZR
ERROR
LOOP
JSR @ENTLO

!ITERATE TEST ROUTINE
:TEST "DLSH"
!INITIALIZE TEST.
:C(AC0) ARE BOTH
!Cleared
!SHIFT 0 PLACES
!EXPECT RESULT
ADD# 0,1,SZR
ERROR !OF BOTH AC'S TO BE ZERO
LOOP
JSR @ENTLO

!ITERATE TEST ROUTINE
:TEST "DLSH"
!INITIALIZE TEST.
:C(AC0)=RANDOM #
:RAND
JSR @ENTRA
MOV 0,2
MOVO 0,1,SKP
LDA 0,-1
!C(AC0)=LEFT SHIFT 16
!C(AC1)=RANDOM DATA
!SHOULD PRODUCE 0 RESULT
!IN C(AC1) AND ORIG RANDOM
!IN C(AC0)
SUB# 0,2,SZR
ERROR
LOOP
JSR @ENTLO

!ITERATE TEST ROUTINE

10081 ECL10
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

!TEST "LSH"
!INITIALIZE TEST.
:C(AC0)=RANDOM #
:RAND
JSR @ENTRA
MOV 0,1,SKP
LDA 0,-1
MOV 1,2
MOV 2,3
LSH 0,2
LSH 0,3
ADDZ 1,1
ADDZ 1,1
SUB# 1,3,SNR
SUB# 1,2,SZR
ERROR
LOOP
JSR @ENTLO

!ITERATE TEST ROUTINE
:TEST "DLSH"
!INITIALIZE TEST.
:C(AC0) ARE BOTH
!Cleared
!SHIFT 0 PLACES
!EXPECT RESULT
ADD# 0,1,SZR
ERROR !OF BOTH AC'S TO BE ZERO
LOOP
JSR @ENTLO

!ITERATE TEST ROUTINE
:TEST "DLSH"
!INITIALIZE TEST.
:C(AC0)=RANDOM #
:RAND
JSR @ENTRA
MOV 0,2
MOVO 0,1,SKP
LDA 0,-1
!C(AC0)=LEFT SHIFT 16
!C(AC1)=RANDOM DATA
!SHOULD PRODUCE 0 RESULT
!IN C(AC1) AND ORIG RANDOM
!IN C(AC0)
SUB# 0,2,SZR
ERROR
LOOP
JSR @ENTLO

!ITERATE TEST ROUTINE

```

```

10079 ECL10
01
02
03
04 05060 006263
05 05061 000100
06
07 05062 006266
08 05063 115001
09 05064 000050
10 05065 024777
11
12 05066 131310
13 05067 154710
14 05070 131310
15 05071 154710
16 05072 131310
17 05073 154710
18 05074 131310
19 05075 154710
20 05076 131310
21 05077 154710
22 05100 131310
23 05101 154710
24 05102 131310
25 05103 154710
26 05104 131310
27 05105 154710
28 05106 131005
29 05107 162414
30
31
32 05114 006264
33
34
35 05115 006263
36 05116 000100
37 05117 101001
38 05120 000020
39 05121 034777
40 05122 171310
41 05123 145310
42 05124 121310
43 05125 115310
44 05126 107015
45 05127 133014
46
47 05134 030764
48 05135 156414
49
50
51 05142 006264
52

```

```

10080 ECL10
01
02
03
04 05143 006263
05 05144 000100
06
07 05145 006266
08 05146 105001
09 05147 177760
10 05150 034777
11 05151 165310
12 05152 125005
13 05153 142414
14
15
16 05160 006264
17
18 05161 006263
19 05162 000100
20 05163 101001
21 05164 177760
22 05165 020777
23 05166 101310
24 05167 125310
25 05170 151310
26 05171 175310
27 05172 133014
28
29 05177 034765
30 05178 034765
31 05200 116414
32
33
34 05205 006264
35
36
37 05206 006263
38 05207 000040
39
40 05210 006266
41 05211 195001
42 05212 000050
43 05213 030777
44 05214 155310
45 05215 101005
46 05216 136414
47
48
49 05223 006264

```

```

;TEST "DLSH"
;INITIALIZE TEST.

SETUP 100
JSR @ENTIN
RAND
JSR @ENTRA
MOV 0,1,SKP
-16.
LDA 3,-1
;C(AC3)=SHIFT COUNT
;C(AC1) TO C(AC2), ZEROS TO C(AC1)

;C(AC0)=CORRECT
;C(AC1)=DLSH RESULT
;C(AC2)=DLSH RESULT
;ITERATE TEST ROUTINE

;TEST "DLSH"
;INITIALIZE TEST.

;C(AC0)=SHIFT RIGHT 16 TIMES
;C(AC0) TO C(AC1), ZERO TO C(AC0)
;C(AC1) TO C(AC2), ZERO TO C(AC1)
;C(AC2) TO C(AC3), ZERO TO C(AC2)
;C(AC3) TO C(AC0), ZERO TO C(AC3)

;ITERATE TEST ROUTINE

;TEST "DLSH"
;INITIALIZE TEST.

;C(AC0)=RANDOM #
;RANDOM NUMBER IS
;SHIFTED FROM AC0 TO AC3

;C(AC0) SHOULD=0
;C(AC1)=ORIGINAL #
;C(AC2)=SHIFT COUNT
;C(AC3)=LSH RESULT
;ITERATE TEST ROUTINE

```

```

;TEST "DLSH"
;INITIALIZE TEST.

SETUP 40
JSR @ENTIN
RAND
JSR @ENTRA
MOV 0,1,SKP
-16.
LDA 2,-1
;C(AC3)=SHIFT COUNT
;C(AC1) TO C(AC2), ZEROS TO C(AC1)

;C(AC0)=CORRECT
;C(AC1)=DLSH RESULT
;C(AC2)=DLSH RESULT
;ITERATE TEST ROUTINE

;TEST "DLSH"
;INITIALIZE TEST.

;C(AC0)=RANDOM #
;RANDOM NUMBER IS
;SHIFTED FROM AC0 TO AC3

;C(AC0) SHOULD=0
;C(AC1)=ORIGINAL #
;C(AC2)=SHIFT COUNT
;C(AC3)=LSH RESULT
;ITERATE TEST ROUTINE

```







```

10087 ECL10
01
02 05763 006263
03 05764 000005
04
05
06
07 05700 006263
08 05701 000005
09
10 05702 100610
11 05703 124610
12 05704 150610
13 05705 174610
14 05706 102000
15 05707 102410
16 05710 133015
17 05711 175014
18
19
20 05716 100610
21 05717 124610
22 05720 150610
23 05721 174610
24 05722 126000
25 05723 126410
26 05724 157015
27 05725 101014
28
29
30 05732 100610
31 05733 124610
32 05734 150610
33 05735 174610
34 05736 152000
35 05737 152410
36 05740 163015
37 05741 125014
38
39
40 05746 100610
41 05747 124610
42 05750 150610
43 05751 174610
44 05752 176000
45 05753 176410
46 05754 107015
47 05755 151014
48
49
50 05762 006264
51

```

```

*** LOB, LOCATE LEAD BIT ***
:
EX1M:
SETUP 5
JSR @ENTIN
I=0
-00 4
ANC I,I
LOB I,I
SET AC I TO ZERO
SHOULD COUNT TO OCTAL 20
SET AC I TO ZERO
SHOULD COUNT TO OCTAL 20
SET AC I TO ZERO
SHOULD COUNT TO OCTAL 20
SET AC I TO ZERO
SHOULD COUNT TO OCTAL 20
SET AC I TO ZERO
SHOULD COUNT TO OCTAL 20
THE SAME
ERROR
MOV# 0,0,SKP
20
LDA 3,-1
SUB# 3,1,SNR
SUB# 3,2, SZR
ERROR
LOOP
JSR @ENTLO
ITERATE TEST ROUTINE

```

```

10088 ECL10
01
02 05763 006263
03 05764 000005
04
05
06
07 05765 100610
08 05766 102410
09 05767 124610
10 05770 126410
11 05771 150610
12 05772 152410
13 05773 174610
14 05774 176410
15 05775 106415
16 05777 000406
17 06000 101001
18 06001 000020
19 06002 034777
20 06003 166415
21 06004 172414
22
23
24 06011 006264
25

```

```

EX1X:
SETUP 5
JSR @ENTIN
I=0
-00 4
ANC I,I
LOB I,I
SET AC I TO ZERO
SHOULD COUNT TO OCTAL 20
SET AC I TO ZERO
SHOULD COUNT TO OCTAL 20
SET AC I TO ZERO
SHOULD COUNT TO OCTAL 20
SET AC I TO ZERO
SHOULD COUNT TO OCTAL 20
THE SAME
ERROR
MOV# 0,0,SKP
20
LDA 3,-1
SUB# 3,1,SNR
SUB# 3,2, SZR
ERROR
LOOP
JSR @ENTLO
ITERATE TEST ROUTINE

```



10091 ECL10

```

01
02
03
04 06105 006263      :TEST "LOB"
05 06106 000100      :INITIALIZE TEST.
06
07 06107 006266      :C(AC0)=RANDOM #
08 06110 115000      :C(AC1)=077777
09 06111 126220      ADCR 1,1
10 06112 111040      MOVO 0,2
11 06113 000010      DO 8.
12 06114 136410      LOB 1,3
13 06115 136410      LOB 1,3
14 06116 136410      LOB 1,3
15 06117 136410      LOB 1,3
16 06118 136410      LOB 1,3
17 06120 136410      LOB 1,3
18 06121 136410      LOB 1,3
19 06122 136410      LOB 1,3
20 06123 170010      ADI 4,2
21 06124 170010      SUB# 3,2,SNR
22 06125 172415      MOV 0,0,SNC
23 06126 101003      ERROR
24
25
26 06133 006264      :ITERATE TEST ROUTINE
27
28
29 06134 006263      :TEST "LOB"
30 06135 000100      :INITIALIZE TEST.
31 06136 102000      :NO COUNTS SHOULD OCCUR
32 06137 105000      :NO COUNTS SHOULD OCCUR
33 06138 000010      :NO COUNTS SHOULD OCCUR
34 06140 106410      LOB 0,1
35 06141 106410      LOB 0,1
36 06142 106410      LOB 0,1
37 06143 106410      LOB 0,1
38 06144 106410      LOB 0,1
39 06145 106410      LOB 0,1
40 06146 106410      LOB 0,1
41 06147 106410      LOB 0,1
42 06150 106415      SUB# 0,1,SNR
43 06151 124014      COM# 1,1,SZR
44      ERROR
45      LOOP
46 06156 006264      :ITERATE TEST ROUTINE
47

```

EX2A:

EX2B:

10092 ECL10

```

01
02
03
04 06157 006263      :TEST "LOB"
05 06160 000100      :INITIALIZE TEST.
06 06161 102520      SUBZL 0,0
07 06162 152401      :THE "LOB" INSTRUCTION WILL
08 06163 000170      :FIRST FIND 15 ZEROS, THEN
09 06164 112410      :14,13,12,ETC. THESE VARIOUS
10 06165 101124      :VALUES SHOULD FORM A SUM
11 06166 000776      JMP *-2
12 06167 034774      LDA 3, *-4
13 06170 156414      SUB# 2,3,SZR
14      ERROR
15 06175 006264      :ITERATE TEST ROUTINE
16
17
18 06176 006263      :TEST "LOB"
19 06177 000020      :INITIALIZE TEST.
20
21 06200 126320      LOBAL 1
22 06201 176401      SUBZL 1,1
23 06202 000170      SUB 1,2,3,1+2,3,SKP
24 06203 136410      LOB 1,1+2,3
25 06204 125124      MOVZL 1,1,SZR
26 06205 000776      JMP *-2
27 06206 020774      LDA 1,3,3, *-4
28 06207 162414      SUB# 1,2,3,1+3,3,SZR
29      ERROR
30      LOOP
31 06214 006264      :ITERATE TEST ROUTINE
32
33
34 06215 006263      :TEST "LOB"
35 06216 000020      :INITIALIZE TEST.
36
37 06217 152520      LOBAL 2
38 06220 102401      SUBZL 2,2
39 06221 000170      SUB 2,2,3,2+2,3,SKP
40 06222 142410      LOB 2,2+2,3
41 06223 151124      MOVZL 2,2,SZR
42 06224 151124      JMP *-2
43 06225 024774      LDA 2,3,3, *-4
44 06226 106414      SUB# 2,2,3,2+3,3,SZR
45      ERROR
46      LOOP
47 06233 006264      :ITERATE TEST ROUTINE
48

```

EX2C:

EX2E:

10093 ECL10

```

01
02
03
04 06234 006263      :TEST "LOB"
05 06235 000100      :INITIALIZE TEST.
06 06236 102520      SUBZL 0,0
07 06237 152401      :THE "LOB" INSTRUCTION WILL
08 06238 000170      :FIRST FIND 15 ZEROS, THEN
09 06239 112410      :14,13,12,ETC. THESE VARIOUS
10 06240 101124      :VALUES SHOULD FORM A SUM
11 06241 000776      JMP *-2
12 06242 034774      LDA 3, *-4
13 06243 156414      SUB# 2,3,SZR
14      ERROR
15 06244 006264      :ITERATE TEST ROUTINE
16
17
18 06245 006263      :TEST "LOB"
19 06246 000020      :INITIALIZE TEST.
20
21 06247 126320      LOBAL 1
22 06248 176401      SUBZL 1,1
23 06249 000170      SUB 1,2,3,1+2,3,SKP
24 06250 136410      LOB 1,1+2,3
25 06251 125124      MOVZL 1,1,SZR
26 06252 000776      JMP *-2
27 06253 020774      LDA 1,3,3, *-4
28 06254 162414      SUB# 1,2,3,1+3,3,SZR
29      ERROR
30      LOOP
31 06255 006264      :ITERATE TEST ROUTINE
32
33
34 06256 006263      :TEST "LOB"
35 06257 000020      :INITIALIZE TEST.
36
37 06258 152520      LOBAL 2
38 06261 102401      SUBZL 2,2
39 06262 000170      SUB 2,2,3,2+2,3,SKP
40 06263 142410      LOB 2,2+2,3
41 06264 151124      MOVZL 2,2,SZR
42 06265 151124      JMP *-2
43 06266 024774      LDA 2,3,3, *-4
44 06267 106414      SUB# 2,2,3,2+3,3,SZR
45      ERROR
46      LOOP
47 06268 006264      :ITERATE TEST ROUTINE
48

```

EX2D:

EX2F:



```

10093 ECL10
01
02
03 06234 006263      :TEST "LOB"
04 06235 000020      :INITIALIZE TEST.
05
06 06236 176520      :LOB RESULT IN C(AC1)
07 06237 126401      :THE "LOB INSTRUCTION WILL
08 06240 000170      :FIND 15 ZEROS THEN 14
09 06241 166410      :ETC. THESE VALUES SHOULD
10 06242 175124      :FORM THE SUM 120 IN C(AC0)
11 06243 030776      :FORM THE SUM 120 IN C(AC0)
12 06244 030774      :FORM THE SUM 120 IN C(AC0)
13 06245 132414      :FORM THE SUM 120 IN C(AC0)
14
15
16 06252 006264      :ITERATE TEST ROUTINE
17
18
19 06253 006263      :TEST "LOB"
20 06254 000020      :INITIALIZE TEST.
21
22 06255 102520      :LOB RESULT IN C(AC2)
23 06256 152401      :THE "LOB INSTRUCTION WILL
24 06257 000170      :FIND 15 ZEROS THEN 14
25 06260 112410      :ETC. THESE VALUES SHOULD
26 06261 101124      :FORM THE SUM 120 IN C(ACD)
27 06262 000776      :FORM THE SUM 120 IN C(ACD)
28 06263 030774      :FORM THE SUM 120 IN C(ACD)
29 06264 156414      :FORM THE SUM 120 IN C(ACD)
30
31
32 06271 006264      :ITERATE TEST ROUTINE
33

```

```

10094 ECL10
01
02
03
04 06272 006263      :TEST "LOB"
05 06273 000100      :INITIALIZE TEST.
06
07 06274 006266      :C(AC0)=RANDOM #
08 06275 105300      :RANDOM NUMBER IS MOSTLY
09 06276 123405      :ZEROS, BUT REJECT ALL ZEROS
10 06277 000775      :CLEAR AC2 AND AC3
11 06300 150510      :WITH HIGH ORDER BIT(1). NO COUNTS
12 06301 170510      :SIMULATE THE "LOB" INSTRUCTION
13 06302 105102      :SIMULATE THE "LOB" INSTRUCTION
14 06303 000404      :SIMULATE THE "LOB" INSTRUCTION
15 06304 151400      :SIMULATE THE "LOB" INSTRUCTION
16 06305 125123      :SIMULATE THE "LOB" INSTRUCTION
17 06306 000776      :SIMULATE THE "LOB" INSTRUCTION
18 06307 116410      :SIMULATE THE "LOB" INSTRUCTION
19 06310 156415      :SIMULATE THE "LOB" INSTRUCTION
20 06311 125003      :SIMULATE THE "LOB" INSTRUCTION
21
22
23 06316 006264      :ITERATE TEST ROUTINE
24
25
26 06317 006263      :TEST "LOB"
27
28
29 06320 006266      :C(AC0)=RANDOM #
30 06321 115000      :C(AC3)=RANDOM
31 06322 150510      :SET C(AC2)=0
32 06323 172410      :COUNT THE NUMBER OF ZEROS (LEADING)
33 06324 155210      :SHIFTING SHOULD LOSE NO BITS
34 06325 150400      :NOW SHIFT RIGHT WHICH SHOULD
35 06326 155210      :RESTORE IT TO ORIGINAL
36 06327 116414      :C(AC0)=ORIGINAL AND CORRECT
37
38
39 06330 006264      :C(AC2)=LOB RESULT NEGATED
40

```

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40

```

```

10095 ECL10
01
02
03 06335 006263
04 06336 000100
05 06337 105300
06 06338 123400
07 06339 141500
08 06340 159600
09 06341 177700
10 06342 195800
11 06343 214000
12 06344 232200
13 06345 250400
14 06346 268600
15 06347 286800
16 06350 305000
17 06351 323200
18 06352 341400
19 06353 359600
20 06354 377800
21 06355 396000
22
23
24 06362 006264
25
26
27 06363 006263
28 06364 000400
29
30 06365 006266
31 06366 105300
32 06367 123400
33 06370 105005
34 06371 000774
35 06372 115122
36 06373 000405
37 06374 164010
38 06375 164010
39 06376 175103
40 06377 000775
41 06400 115000
42
43 06401 162410
44 06402 162410
45 06403 162410
46 06404 162410
47 06405 162410
48 06406 162410
49 06407 162410
50 06410 162410
51 06411 106414
52
53
54 06416 006264

```

```

:TEST "LOB"
:INITIALIZE TEST.
EX2L:
SETUP 10
JSR @ENTIN
10
ADCR 2,2
ADCR 1,1
LOB 2,1
MOV 1,1,SNR
MOV 1,1,SNC
ERROR
LOOP
JSR @ENTLO
12 06432 006264
13

```

```

:TEST "LOB"
:INITIALIZE TEST.
EX2J:
SETUP 100
JSR @ENTIN
100
RAND
JSR @ENTRA
MOV 0,1
AND 1,0
MOV 0,2
XOR 3,5
LOB 2,3
LOB 2,3
LOB 2,3
LOB 2,3
MOVZR 3,1
MOVZR 1,1
LSH 1,2
NEG 1,1
LSH 1,2
SUB# 2,0,SZR
ERROR
LOOP
JSR @ENTLO
24 06362 006264
25
26
27 06363 006263
28 06364 000400
29
30 06365 006266
31 06366 105300
32 06367 123400
33 06370 105005
34 06371 000774
35 06372 115122
36 06373 000405
37 06374 164010
38 06375 164010
39 06376 175103
40 06377 000775
41 06400 115000
42
43 06401 162410
44 06402 162410
45 06403 162410
46 06404 162410
47 06405 162410
48 06406 162410
49 06407 162410
50 06410 162410
51 06411 106414
52
53
54 06416 006264

```

```

:TEST "LOB"
:INITIALIZE TEST.
EX2L:
SETUP 10
JSR @ENTIN
10
ADCR 2,2
ADCR 1,1
LOB 2,1
MOV 1,1,SNR
MOV 1,1,SNC
ERROR
LOOP
JSR @ENTLO
12 06432 006264
13

```

```

:TEST "LOB"
:INITIALIZE TEST.
EX2J:
SETUP 100
JSR @ENTIN
100
RAND
JSR @ENTRA
MOV 0,1
AND 1,0
MOV 0,2
XOR 3,5
LOB 2,3
LOB 2,3
LOB 2,3
LOB 2,3
MOVZR 3,1
MOVZR 1,1
LSH 1,2
NEG 1,1
LSH 1,2
SUB# 2,0,SZR
ERROR
LOOP
JSR @ENTLO
24 06362 006264
25
26
27 06363 006263
28 06364 000400
29
30 06365 006266
31 06366 105300
32 06367 123400
33 06370 105005
34 06371 000774
35 06372 115122
36 06373 000405
37 06374 164010
38 06375 164010
39 06376 175103
40 06377 000775
41 06400 115000
42
43 06401 162410
44 06402 162410
45 06403 162410
46 06404 162410
47 06405 162410
48 06406 162410
49 06407 162410
50 06410 162410
51 06411 106414
52
53
54 06416 006264

```









```

10105 ECL10
01
02 07170 014204 EXTR: DSZ PASSIN
03 07171 000437 JMP DOMOR
04 07172 010203 ISZ PASS
05 07173 010100 MOV 0,0,SKP
06 07174 063077 HALT
07 07175 020205 LDA 0,PASSVL
08 07176 040204 STA 0,PASSIN
09
10 07177 060477 READS 0
11 07200 101112 MOVL# 0,0,SZC
12 07201 000403 JMP +3
13 07202 122470 ELDA 0,SRREG
14 000034
15 07204 143770 ANDI 184,0
16 004000
17 07206 101004 MOV 0,0,SZR
18 07207 000406 JMP PSCK1
19
20 07210 006221 JSR @IMESS
21 07211 001227 PASHES
22 07212 125020 MOVZ 1,1
23 07213 024203 LDA 1,PASS
24 07214 006224 JSR @IPECC
25
26 07215 034010-PSCK1: LDA 3,,EGGS
27 07216 021400 LDA 0,0,3
28 07217 101005 MOV 0,0,SNR
29 07220 000410 JMP +8
30 07221 015403 DSZ 3,3
31 07222 000406 JMP +6
32 07223 062677 IORST
33 07224 021403 LDA 0,3,3
34 07225 035404 LDA 3,4,3
35 07226 041776 STA 0,-2,3
36 07227 001400 JMP 0,3
37
38 07230 006220 DOMOR: CALL
39 JSR @ICAL
40 REL
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

```

```

10106 ECL10
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

```

```

:*****EGGS & DIRT DATA BLOCKS*****
EGGS:
AUTO: 0
DEV: 0
CATSW: 0
PCNT: 0
RTN: 0
SAREG: 0
PRGEND: PRGEND
:TXI /COPYRIGHT(C)06C,1974,75,76
:DTOS AUTO MODE SWITCH
:PRIMARY DEVICE CODE TO BE TESTED
:CAT SWITCH, SET IF CAT LOADED
:PASS COUNT, # OF TIMES TO RUN
:RETURN POINT TO RESTORE DTOS
:DEFAULT SWITCH REGISTER

```

```

05 07232 000000 AUTO: 0
06 07233 000000 DEV: 0
07 07234 000000 CATSW: 0
08 07235 000000 PCNT: 0
09 07236 000000 RTN: 0
10 07237 000000 SAREG: 0
11
12 07240 007240 PRGEND: PRGEND
13
14 07241 047503 .TXI /COPYRIGHT(C)06C,1974,75,76
15 054520
16 044522
17 049107
18 024124
19 026503
20 043504
21 026103
22 034461
23 032067
24 033454
25 026065
26 033067
27 07256 046101 ALL RIGHTS RESERVED/
28 020114
29 044522
30 044107
31 051524
32 051040
33 051505
34 051105
35 042526
36 000104
37
38 07270 141705 DIRT: .TXI IECLIPSE103I
39 144714
40 051520
41 130705
42 031460
43 000000
44 07276 000000
45 07277 000200 DTOSR
46 07300 175772 175772
47 07301 000000
48 07302 000000
49 07303 000000
50 07304 000000
51 07305 000000
52
53
54 .END DTOSR
55 00013-000012
56 000017
57 000240
58 177770
59 000212
60 104400

```

```

: *** END OF TEST ROUTINES ***

```

0107 ECL10  
 01 022000  
 02 000377  
 03 000660  
 04 000011  
 05 035031  
 06 000144  
 07 000200  
 08 000400  
 09 001777  
 10 000263  
 11 000257  
 12 000004

0108 ECL10

| AC0   | 000225    | 18/16  | 21/08  | 21/30  | 22/21  | 22/34  | 23/05  | 23/28  |
|-------|-----------|--------|--------|--------|--------|--------|--------|--------|
| AC0   | 000225    | 18/16  | 21/08  | 21/30  | 22/21  | 22/34  | 23/05  | 23/28  |
| AC1   | 000226    | 23/58  | 22/22  | 22/35  | 23/15  | 23/30  | 23/46  |        |
| AC2   | 000227    | 18/17  | 22/23  | 22/36  | 23/16  | 23/32  | 23/47  |        |
| AC3   | 000230    | 18/18  | 23/34  | 23/59  | 34/17  | 34/27  | 34/29  | 34/40  |
|       |           | 18/19  | 35/10  | 35/35  | 35/35  | 35/55  | 37/26  | 38/32  |
|       |           | 38/42  | 39/17  | 39/17  | 39/20  | 39/34  | 39/37  | 39/49  |
|       |           | 38/34  | 38/37  | 39/17  | 42/37  | 42/39  | 43/57  | 43/59  |
|       |           | 40/30  | 41/29  | 41/32  | 45/50  | 46/17  | 46/19  | 46/45  |
|       |           | 45/15  | 45/27  | 45/39  | 49/15  | 49/32  | 49/49  | 50/16  |
|       |           | 47/18  | 48/21  | 48/38  | 51/15  | 51/43  | 51/54  | 52/11  |
|       |           | 50/41  | 50/57  | 51/15  | 51/32  | 51/43  | 51/54  | 56/19  |
|       |           | 52/22  | 53/17  | 53/35  | 54/21  | 54/50  | 55/13  | 58/12  |
|       |           | 56/36  | 56/47  | 56/58  | 57/17  | 57/29  | 58/12  | 58/25  |
|       |           | 59/13  | 59/24  | 60/22  | 60/38  | 61/19  | 61/38  | 61/41  |
|       |           | 61/55  | 62/15  | 62/26  | 62/37  | 62/48  | 63/45  | 64/45  |
|       |           | 65/15  | 66/45  | 66/49  | 67/19  | 67/22  | 67/25  | 67/28  |
|       |           | 67/48  | 67/51  | 67/55  | 67/58  | 68/18  | 68/21  | 68/39  |
|       |           | 68/41  | 68/55  | 69/12  | 69/15  | 69/34  | 69/37  | 69/54  |
|       |           | 70/35  | 70/48  | 71/12  | 71/26  | 71/43  | 72/15  | 72/29  |
|       |           | 72/37  | 72/45  | 73/24  | 73/41  | 73/52  | 74/03  | 74/14  |
|       |           | 74/33  | 75/16  | 75/20  | 76/23  | 76/23  | 77/25  | 77/37  |
|       |           | 78/21  | 78/32  | 78/48  | 79/31  | 79/47  | 79/50  | 80/15  |
|       |           | 80/30  | 80/33  | 80/48  | 81/27  | 81/56  | 83/21  | 84/43  |
|       |           | 85/56  | 86/28  | 86/40  | 87/19  | 87/29  | 87/39  | 87/49  |
|       |           | 88/23  | 89/33  | 90/28  | 91/25  | 91/45  | 92/15  | 92/31  |
|       |           | 92/47  | 93/15  | 93/31  | 94/22  | 94/38  | 95/23  | 95/33  |
|       |           | 96/11  | 97/19  | 97/29  | 97/39  | 97/49  | 98/12  | 98/17  |
|       |           | 98/22  | 98/27  | 99/12  | 99/18  | 99/24  | 99/30  | 99/47  |
|       |           | 100/12 | 100/57 | 101/32 | 101/54 | 102/37 | 102/54 | 103/17 |
|       |           | 103/34 | 104/15 | 104/35 |        |        |        |        |
| ACD   | 000104 MC | 8/32   | 48/27  | 49/04  | 49/21  | 49/38  | 43/19  | 43/29  |
| ADDAL | 000063 MC | 8/19   | 38/10  | 38/15  | 38/20  | 38/25  |        |        |
|       |           | 43/39  | 43/49  |        |        |        |        |        |
| ANAC  | 000152 MC | 9/12   | 56/25  | 56/36  | 56/47  |        |        |        |
| ASSEM | 000072 MC | 8/26   | 43/10  | 46/07  |        |        |        |        |
| AUTO  | 007332    | 20/15  | 23/48  | 106/05 |        |        |        |        |
| BAMBL | 000257    | 18/41  | 18/42  | 18/50  |        |        |        |        |
| BAMBY | 000247    | 18/37  | 20/34  | 25/37  |        |        |        |        |
| BCA   | 001413    | 33/13  | 33/33  |        |        |        |        |        |
| BCA1  | 001420    | 33/18  |        |        |        |        |        |        |
| BCS   | 001342    | 32/04  |        |        |        |        |        |        |
| RCS1  | 001351    | 32/12  | 32/31  |        |        |        |        |        |
| RCS2  | 001364    | 32/24  |        |        |        |        |        |        |
| REGIN | 001443    | 18/47  | 25/40  | 34/09  |        |        |        |        |
| RCNAD | 000202    | 17/45  | 17/47  |        |        |        |        |        |
| CAL   | 000767    | 18/08  | 24/22  |        |        |        |        |        |
| CAL0  | 000237    | 18/26  | 24/22  | 24/29  |        |        |        |        |
| CAL1  | 000240    | 18/27  | 24/23  | 24/30  |        |        |        |        |
| CAL2  | 000241    | 18/28  | 24/27  | 24/31  |        |        |        |        |
| CALL  | 000016 MC | 7/17   | 18/44  | 105/37 | 105/37 | 106/07 |        |        |
| CATSM | 007234    | 17/31  | 19/11  | 19/32  | 25/08  | 26/03  |        |        |
| CHAR  | 001144    | 26/42  | 26/57  | 26/60  | 28/03  |        |        |        |
| CHAR1 | 001173    | 28/30  | 28/47  | 29/01  |        |        |        |        |
| CHAR2 | 001202    | 28/33  | 28/39  |        |        |        |        |        |
| CHAR3 | 001210    | 28/40  | 28/46  |        |        |        |        |        |
| CHAR4 | 001231    | 28/27  | 28/57  |        |        |        |        |        |
| CHARE | 000232    | 18/21  | 28/04  |        |        |        |        |        |
| CHORZ | 000233    | 18/21  | 28/30  | 28/50  | 28/57  |        |        |        |
| CHRSV | 001226    | 28/05  | 28/52  | 29/05  |        |        |        |        |



0109 ECL10

0110 FCL10

| Code  | MC     | 11/03 | 97/09  | 97/19  | 97/29  | 97/39  | 82/04  | 83/25  | 85/04  | 86/02  | 86/32  | 87/07  | 88/02  |
|-------|--------|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| CORAC | 000353 | MC    | 17/22  | 106/04 | 34/40  | 34/40  | 82/04  | 83/25  | 85/04  | 86/02  | 86/32  | 87/07  | 88/02  |
| CRY   | 000231 |       | 34/27  | 34/27  | 34/29  | 34/40  | 89/04  | 90/02  | 91/04  | 91/29  | 92/04  | 92/19  | 92/35  |
| DADAC | 001122 | MC    | 35/35  | 35/35  | 38/32  | 38/32  | 93/03  | 93/19  | 94/04  | 94/26  | 95/04  | 95/27  | 96/03  |
| DEV   | 007233 |       | 39/17  | 39/20  | 39/34  | 39/37  | 97/07  | 98/03  | 99/03  | 99/36  | 100/02 | 100/16 | 101/03 |
| DIRT  | 007270 |       | 41/32  | 42/37  | 43/57  | 43/59  | 101/36 | 102/04 | 102/41 | 103/04 | 103/21 | 104/03 | 104/20 |
| DLR   | 000553 | MC    | 45/39  | 45/50  | 46/17  | 46/19  | 18/57  | 34/18  | 36/30  | 38/49  | 39/21  | 39/42  | 36/20  |
| DOMOR | 007230 |       | 49/32  | 49/32  | 49/49  | 46/45  | 37/25  | 38/38  | 39/21  | 39/56  | 40/37  | 41/53  | 42/46  |
| DRB   | 000607 | MC    | 51/43  | 51/54  | 52/11  | 52/16  | 44/09  | 45/16  | 45/28  | 45/40  | 45/51  | 46/26  | 46/46  |
| DS12  | 001026 | MC    | 54/21  | 54/29  | 55/13  | 56/19  | 47/19  | 48/22  | 48/39  | 49/16  | 49/33  | 49/50  | 50/17  |
| DSR1  | 001183 | MC    | 57/17  | 57/24  | 58/12  | 58/25  | 50/42  | 50/58  | 51/16  | 51/55  | 52/12  | 52/23  | 53/18  |
| DSBAC | 001143 | MC    | 60/22  | 60/38  | 61/19  | 61/58  | 53/36  | 54/22  | 54/51  | 55/14  | 56/20  | 56/29  | 57/18  |
| DSL12 | 001066 | MC    | 62/26  | 62/37  | 63/45  | 64/45  | 57/30  | 58/13  | 58/32  | 59/14  | 59/31  | 60/23  | 60/39  |
| DSL4  | 000737 | MC    | 66/49  | 67/19  | 67/22  | 67/28  | 61/20  | 61/42  | 61/56  | 62/49  | 63/46  | 64/46  | 65/16  |
| DSL8  | 000703 | MC    | 67/55  | 68/18  | 68/21  | 68/39  | 66/50  | 67/29  | 67/59  | 68/22  | 68/42  | 68/56  | 69/23  |
| DSR4  | 000772 | MC    | 71/12  | 71/43  | 72/15  | 72/29  | 69/44  | 69/55  | 70/34  | 70/49  | 71/13  | 71/27  | 71/44  |
| DSR8  | 000642 | MC    | 73/24  | 73/41  | 74/03  | 74/29  | 72/16  | 72/08  | 73/25  | 74/15  | 74/34  | 75/21  | 75/44  |
| DT088 | 000200 |       | 75/20  | 75/43  | 76/23  | 77/37  | 76/24  | 77/26  | 77/38  | 78/22  | 78/33  | 78/49  | 79/32  |
| DUBL  | 000484 | MC    | 78/48  | 79/31  | 79/47  | 79/50  | 79/51  | 80/16  | 80/34  | 80/49  | 81/28  | 81/57  | 83/22  |
| EGGS  | 007232 |       | 80/48  | 81/27  | 81/56  | 83/21  | 84/44  | 85/57  | 86/29  | 86/41  | 87/50  | 88/24  | 89/34  |
| ENTER | 000285 |       | 86/40  | 87/19  | 87/59  | 87/39  | 90/29  | 91/26  | 91/46  | 92/16  | 92/32  | 92/48  | 93/16  |
|       |        |       | 90/28  | 91/25  | 91/45  | 92/15  | 93/32  | 94/23  | 94/39  | 95/24  | 95/54  | 96/12  | 97/50  |
|       |        |       | 93/31  | 94/22  | 94/38  | 95/23  | 98/28  | 99/31  | 99/48  | 100/13 | 100/58 | 101/33 | 101/55 |
|       |        |       | 97/29  | 97/39  | 97/49  | 98/12  | 102/38 | 102/55 | 103/18 | 103/35 | 104/16 | 104/34 |        |
|       |        |       | 99/12  | 99/18  | 99/24  | 99/30  | 18/59  | 25/13  | 50/05  | 50/25  | 50/48  | 53/07  |        |
|       |        |       | 101/32 | 101/54 | 102/37 | 103/17 | 54/06  | 54/28  | 55/06  | 59/05  | 60/07  | 60/29  |        |
|       |        |       | 104/35 |        |        |        | 64/07  | 65/05  | 66/07  | 67/10  | 67/35  | 68/07  |        |
|       |        |       |        |        |        |        | 68/48  | 70/10  | 70/40  | 71/05  | 71/33  | 72/05  |        |
|       |        |       |        |        |        |        | 75/29  | 76/05  | 77/06  | 78/07  | 78/39  | 79/07  |        |
|       |        |       |        |        |        |        | 80/40  | 81/07  | 81/34  | 82/07  | 83/28  | 85/07  |        |
|       |        |       |        |        |        |        | 90/05  | 91/07  | 92/04  | 94/29  | 95/07  | 95/30  |        |
|       |        |       |        |        |        |        | 101/39 | 102/07 | 102/47 | 103/07 | 103/24 | 104/06 |        |
|       |        |       |        |        |        |        | 23/18  | 30/14  | 23/04  |        |        |        |        |
|       |        |       |        |        |        |        | 18/52  | 18/58  | 23/04  |        |        |        |        |
|       |        |       |        |        |        |        | 23/12  | 23/51  | 23/58  |        |        |        |        |
|       |        |       |        |        |        |        | 17/56  | 23/04  | 23/37  |        |        |        |        |
|       |        |       |        |        |        |        | 7/22   | 34/16  | 34/26  |        |        |        |        |
|       |        |       |        |        |        |        | 35/13  | 35/34  | 38/26  |        |        |        |        |
|       |        |       |        |        |        |        | 39/16  | 39/19  | 39/34  |        |        |        |        |
|       |        |       |        |        |        |        | 41/31  | 42/36  | 42/38  |        |        |        |        |
|       |        |       |        |        |        |        | 45/38  | 45/49  | 46/18  |        |        |        |        |
|       |        |       |        |        |        |        | 48/37  | 49/14  | 49/31  |        |        |        |        |
|       |        |       |        |        |        |        | 51/14  | 51/31  | 51/42  |        |        |        |        |
|       |        |       |        |        |        |        | 53/34  | 54/20  | 54/49  |        |        |        |        |
|       |        |       |        |        |        |        | 56/57  | 57/16  | 57/28  |        |        |        |        |
|       |        |       |        |        |        |        | 60/21  | 60/37  | 61/18  |        |        |        |        |
|       |        |       |        |        |        |        | 62/25  | 62/36  | 63/07  |        |        |        |        |
|       |        |       |        |        |        |        | 66/48  | 67/18  | 67/21  |        |        |        |        |
|       |        |       |        |        |        |        | 67/54  | 67/57  | 68/20  |        |        |        |        |
|       |        |       |        |        |        |        | 69/11  | 69/14  | 69/33  |        |        |        |        |
|       |        |       |        |        |        |        | 71/11  | 71/25  | 71/42  |        |        |        |        |
|       |        |       |        |        |        |        | 73/23  | 73/40  | 73/51  |        |        |        |        |
|       |        |       |        |        |        |        | 75/19  | 75/42  | 76/22  |        |        |        |        |
|       |        |       |        |        |        |        | 78/47  | 79/46  | 79/46  |        |        |        |        |
|       |        |       |        |        |        |        | 80/47  | 81/26  | 81/55  |        |        |        |        |
|       |        |       |        |        |        |        | 86/39  | 87/28  | 87/38  |        |        |        |        |
|       |        |       |        |        |        |        | 90/27  | 91/24  | 91/40  |        |        |        |        |
|       |        |       |        |        |        |        | 93/30  | 94/31  | 94/37  |        |        |        |        |
|       |        |       |        |        |        |        | 97/28  | 97/36  | 97/48  |        |        |        |        |

ENTIN 000263

MC

ENTRA 000266

ENTLO 000264

ERM8G 001241

ERR 000666

ERR1 000746

ERR2 000212

ERROR 000023

0111 ECL10

|      |        |        |        |        |        |        |        |        |
|------|--------|--------|--------|--------|--------|--------|--------|--------|
| EX0  | 001444 | 99/11  | 99/17  | 99/23  | 99/29  | 99/46  | 100/11 | 100/56 |
| EX00 | 003031 | 101/31 | 101/53 | 102/36 | 102/53 | 103/16 | 103/53 | 104/14 |
| EX01 | 003056 | 54/02  |        |        |        |        |        |        |
| EX02 | 003111 | 55/02  |        |        |        |        |        |        |
| EX03 | 003126 | 56/04  |        |        |        |        |        |        |
| EX04 | 003147 | 56/22  |        |        |        |        |        |        |
| EX05 | 003216 | 57/01  |        |        |        |        |        |        |
| EX06 | 003237 | 57/20  |        |        |        |        |        |        |
| EX07 | 003253 | 58/03  |        |        |        |        |        |        |
| EX08 | 003272 | 58/19  |        |        |        |        |        |        |
| EX0R | 003310 | 58/28  | 58/30  |        |        |        |        |        |
| EX09 | 003311 | 59/01  | 59/29  |        |        |        |        |        |
| EX0A | 003332 | 59/20  | 59/30  |        |        |        |        |        |
| EX0A | 003346 | 59/27  |        |        |        |        |        |        |
| EX0B | 003347 | 60/03  |        |        |        |        |        |        |
| EX0C | 003374 | 60/25  |        |        |        |        |        |        |
| EX0D | 003413 | 61/04  |        |        |        |        |        |        |
| EX0E | 003430 | 61/22  |        |        |        |        |        |        |
| EX0E | 003482 | 61/41  |        |        |        |        |        |        |
| EX0F | 003485 | 61/06  |        |        |        |        |        |        |
| EX0G | 003477 | 62/01  |        |        |        |        |        |        |
| EX0H | 003562 | 63/01  |        |        |        |        |        |        |
| EX0I | 003637 | 64/03  |        |        |        |        |        |        |
| EX0J | 003702 | 65/01  |        |        |        |        |        |        |
| EX0K | 003722 | 66/03  |        |        |        |        |        |        |
| EX0K | 004003 | 66/49  |        |        |        |        |        |        |
| EX0L | 004094 | 67/06  |        |        |        |        |        |        |
| EX0M | 004083 | 67/31  |        |        |        |        |        |        |
| EX0M | 004095 | 67/31  | 67/38  |        |        |        |        |        |
| EX0N | 004074 | 67/64  | 67/52  |        |        |        |        |        |
| EX0N | 004110 | 67/51  | 67/58  |        |        |        |        |        |
| EX0O | 004111 | 68/03  |        |        |        |        |        |        |
| EX0P | 004140 | 68/24  |        |        |        |        |        |        |
| EX0Q | 004166 | 68/44  |        |        |        |        |        |        |
| EX0R | 004206 | 69/07  | 69/21  |        |        |        |        |        |
| EX0R | 004232 | 69/18  | 69/22  |        |        |        |        |        |
| EX0S | 004236 | 69/29  | 69/42  |        |        |        |        |        |
| EX0S | 004261 | 69/40  | 69/43  |        |        |        |        |        |
| EX0T | 004262 | 69/46  |        |        |        |        |        |        |
| EX0U | 004275 | 70/06  |        |        |        |        |        |        |
| EX0V | 004331 | 70/36  |        |        |        |        |        |        |
| EX0W | 004347 | 71/01  |        |        |        |        |        |        |
| EX0X | 004364 | 71/16  |        |        |        |        |        |        |
| EX0Y | 004401 | 71/29  |        |        |        |        |        |        |
| EX0Z | 004421 | 72/01  |        |        |        |        |        |        |
| EX0- | 004441 | 72/18  |        |        |        |        |        |        |
| EX1  | 001456 | 34/20  |        |        |        |        |        |        |
| EX1A | 004477 | 73/07  |        |        |        |        |        |        |
| EX1B | 004525 | 73/27  |        |        |        |        |        |        |
| EX1C | 004610 | 74/17  |        |        |        |        |        |        |
| EX1D | 004633 | 75/03  |        |        |        |        |        |        |
| EX1E | 004662 | 75/25  |        |        |        |        |        |        |
| EX1F | 004706 | 76/01  |        |        |        |        |        |        |
| EX1G | 004736 | 77/02  |        |        |        |        |        |        |
| EX1H | 005003 | 78/03  |        |        |        |        |        |        |

0112 ECL10

|       |        |        |       |  |  |  |  |  |
|-------|--------|--------|-------|--|--|--|--|--|
| EX1I  | 005027 | 78/24  |       |  |  |  |  |  |
| EX1J  | 005042 | 78/35  |       |  |  |  |  |  |
| EX1K  | 005060 | 79/03  |       |  |  |  |  |  |
| EX1L  | 005115 | 79/34  | 79/47 |  |  |  |  |  |
| EX1L  | 005120 | 80/03  |       |  |  |  |  |  |
| EX1M  | 005143 | 80/18  |       |  |  |  |  |  |
| EX1M  | 005161 | 80/22  | 80/30 |  |  |  |  |  |
| EX1N  | 005164 | 80/36  |       |  |  |  |  |  |
| EX1O  | 005206 | 81/06  | 81/12 |  |  |  |  |  |
| EX1P  | 005226 | 81/16  | 81/22 |  |  |  |  |  |
| EX1P  | 005237 | 81/33  | 81/39 |  |  |  |  |  |
| EX1Q  | 005260 | 81/44  | 81/51 |  |  |  |  |  |
| EX1Q  | 005272 | 82/03  |       |  |  |  |  |  |
| EX1R  | 005312 | 83/24  |       |  |  |  |  |  |
| EX1S  | 005431 | 85/03  |       |  |  |  |  |  |
| EX1T  | 005551 | 86/01  |       |  |  |  |  |  |
| EX1U  | 005627 | 86/31  |       |  |  |  |  |  |
| EX1V  | 005664 | 87/01  |       |  |  |  |  |  |
| EX1W  | 005700 | 87/06  |       |  |  |  |  |  |
| EX1X  | 005743 | 88/01  |       |  |  |  |  |  |
| EX1Y  | 006012 | 89/03  |       |  |  |  |  |  |
| EX1Z  | 006051 | 90/01  |       |  |  |  |  |  |
| EX1-  | 004167 | 77/28  |       |  |  |  |  |  |
| EX2   | 001475 | 34/32  |       |  |  |  |  |  |
| EX20  | 006647 | 100/01 |       |  |  |  |  |  |
| EX2A  | 006105 | 91/03  |       |  |  |  |  |  |
| EX2B  | 006134 | 91/28  |       |  |  |  |  |  |
| EX2C  | 006157 | 92/03  |       |  |  |  |  |  |
| EX2D  | 006176 | 92/18  |       |  |  |  |  |  |
| EX2E  | 006215 | 92/34  |       |  |  |  |  |  |
| EX2F  | 006234 | 93/02  |       |  |  |  |  |  |
| EX2G  | 006253 | 93/18  |       |  |  |  |  |  |
| EX2H  | 006274 | 94/06  |       |  |  |  |  |  |
| EX2I  | 006317 | 94/25  |       |  |  |  |  |  |
| EX2J  | 006335 | 95/03  |       |  |  |  |  |  |
| EX2K  | 006365 | 95/29  |       |  |  |  |  |  |
| EX2L  | 006417 | 96/02  |       |  |  |  |  |  |
| EX2M  | 006433 | 97/06  |       |  |  |  |  |  |
| EX2N  | 006516 | 98/02  |       |  |  |  |  |  |
| EX2O  | 006561 | 99/02  |       |  |  |  |  |  |
| EX2P  | 006630 | 99/35  |       |  |  |  |  |  |
| EX2R  | 006665 | 100/15 |       |  |  |  |  |  |
| EX2S  | 006721 | 101/02 |       |  |  |  |  |  |
| EX2T  | 006760 | 101/35 |       |  |  |  |  |  |
| EX2U  | 007004 | 102/03 |       |  |  |  |  |  |
| EX2V  | 007047 | 102/40 |       |  |  |  |  |  |
| EX2W  | 007067 | 103/03 |       |  |  |  |  |  |
| EX2X  | 007107 | 103/20 |       |  |  |  |  |  |
| EX2Y  | 007127 | 104/02 |       |  |  |  |  |  |
| EX2Z  | 007146 | 104/19 |       |  |  |  |  |  |
| EX2-  | 001500 | 34/36  | 34/47 |  |  |  |  |  |
| EX2.1 | 001500 | 34/45  | 34/48 |  |  |  |  |  |
| EX2.2 | 001521 | 35/02  |       |  |  |  |  |  |
| EX3   | 001522 | 35/06  | 35/19 |  |  |  |  |  |
| EX3.1 | 001525 | 35/17  | 35/20 |  |  |  |  |  |
| EX3.2 | 001550 | 35/24  |       |  |  |  |  |  |
| EX4   | 001551 | 35/28  |       |  |  |  |  |  |
| EX4.1 | 001554 | 35/40  |       |  |  |  |  |  |
| EX4.2 | 001573 | 35/58  |       |  |  |  |  |  |



0116 ECL10

0115 ECL10

|               |        |        |        |        |        |        |        |        |        |        |        |        |
|---------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| MAXLO 000217  | 18/04  | 19/37  | 25/14  | 25/29  |        |        | 93/02  | 93/18  | 94/25  | 95/03  | 95/26  | 96/02  |
| MEMTO 000011- | 17/23  | 19/09  | 20/06  | 24/46  |        |        | 97/06  | 98/02  | 99/02  | 100/01 | 100/15 | 101/02 |
| MESIZ 001302  | 20/05  | 30/47  | 26/59  |        |        |        | 101/35 | 102/03 | 102/40 | 103/20 | 104/02 | 104/19 |
| MESRE 000235  | 18/24  | 26/49  |        |        |        |        | 18/10  | 24/36  | 24/47  |        |        |        |
| MESS 001127   | 18/09  | 26/49  |        |        |        |        | 20/18  | 20/29  | 20/31  |        |        |        |
| MESS1 001132  | 26/52  | 27/01  |        |        |        |        | 19/27  | 19/35  |        |        |        |        |
| MINLO 000216  | 18/03  | 19/35  | 24/36  | 25/15  |        |        | 8/12   | 41/06  | 41/11  | 41/21  | 43/14  | 43/24  |
| NDCAT 00524   | 19/14  | 19/21  | 19/29  |        |        |        | 43/34  | 43/44  |        |        |        |        |
| NSTR 000500   | 17/47  | 19/05  |        |        |        |        | 17/34  | 20/04  |        |        |        |        |
| OFF 000170    | 17/29  |        |        |        |        |        | 20/27  | 22/16  |        |        |        |        |
| ON 000171     | 17/30  |        |        |        |        |        | 9/21   | 62/04  | 28/10  | 105/13 | 106/10 |        |
| PASME 001227  | 30/04  | 105/21 | 21/22  | 23/19  | 105/04 | 105/23 | 20/27  | 22/16  | 62/26  | 62/37  |        |        |
| PASS 000203   | 17/48  | 20/13  | 105/08 |        |        |        | 51/32  | 51/32  | 51/43  | 46/31  | 47/04  | 48/06  |
| PASS1 000204  | 17/49  | 105/02 |        |        |        |        | 8/05   | 39/09  | 41/01  | 51/22  | 51/33  | 51/44  |
| PASSV 000205  | 106/08 | 105/07 |        |        |        |        | 48/28  | 49/05  | 49/39  | 61/07  | 62/05  | 62/16  |
| PAST 007235   | 18/12  | 26/04  |        |        |        |        | 56/07  | 56/26  | 56/37  | 73/53  | 74/04  | 74/04  |
| PDEC 001054   | 26/26  | 26/44  |        |        |        |        | 62/27  | 62/38  | 73/31  | 32/18  | 32/27  | 33/18  |
| PDEL1 001101  | 26/32  | 26/36  |        |        |        |        | 18/32  | 32/07  | 32/16  | 32/33  | 33/15  | 33/28  |
| PDEL2 001113  | 26/07  | 26/17  | 26/29  | 26/41  |        |        | 18/30  | 32/06  | 32/34  | 33/14  | 33/35  | 33/34  |
| PDECS 001120  | 18/23  | 26/05  | 26/15  | 26/45  |        |        | 18/30  | 32/06  | 32/34  | 33/14  | 33/35  | 33/34  |
| PDERE 000234  | 22/25  | 30/09  |        |        |        |        | 17/22  | 23/54  | 105/26 |        |        |        |
| PERE 001234   | 18/11  | 26/14  |        |        |        |        | 18/53  | 18/59  | 125/04 |        |        |        |
| POCT 001066   | 18/03  | 19/16  | 106/12 |        |        |        |        |        |        |        |        |        |
| PRGEN 007240  | 105/18 | 105/26 |        |        |        |        |        |        |        |        |        |        |
| PSCK1 007215  | 18/25  | 24/04  |        |        |        |        |        |        |        |        |        |        |
| RAN 000236    | 7/13   | 25/12  | 24/16  |        |        |        |        |        |        |        |        |        |
| RAND 000012   | 54/05  | 54/27  | 50/04  | 50/24  | 50/47  | 53/06  | 60/28  | 60/06  | 60/28  | 63/04  | 63/04  | 63/04  |
|               | 64/06  | 65/04  | 66/06  | 67/09  | 67/34  | 68/06  | 68/27  | 71/04  | 71/32  | 72/04  | 73/06  | 73/06  |
|               | 68/27  | 70/09  | 71/04  | 71/32  | 72/04  | 73/06  | 73/06  | 78/06  | 79/06  | 80/06  | 80/06  | 80/06  |
|               | 75/28  | 76/04  | 77/05  | 78/06  | 78/38  | 80/06  | 83/27  | 85/06  | 89/06  | 89/06  | 89/06  | 89/06  |
|               | 80/39  | 81/06  | 81/33  | 82/06  | 83/27  | 85/06  | 94/28  | 95/29  | 101/05 | 101/05 | 101/05 | 101/05 |
|               | 90/04  | 91/06  | 94/06  | 94/28  | 95/29  | 101/05 | 101/38 | 102/06 | 102/43 | 103/23 | 104/05 | 104/22 |
|               | 101/38 | 102/06 | 102/43 | 103/26 | 103/23 | 104/05 |        |        |        |        |        |        |
| REL 001015    | 25/08  | 25/22  | 105/40 |        |        |        |        |        |        |        |        |        |
| RELO 001031   | 25/21  |        |        |        |        |        |        |        |        |        |        |        |
| REL1 001044   | 25/25  |        |        |        |        |        |        |        |        |        |        |        |
| REL2 001052   | 25/17  | 25/38  |        |        |        |        |        |        |        |        |        |        |
| RELDC 000214  | 17/58  | 18/39  | 20/33  | 21/14  | 23/38  | 24/24  | 25/23  |        |        |        |        |        |
|               | 25/32  | 28/20  | 28/52  |        |        |        |        |        |        |        |        |        |
| REST 001215   | 28/16  |        |        |        |        |        |        |        |        |        |        |        |
| RTN 007256    | 106/09 |        |        |        |        |        |        |        |        |        |        |        |
| SETSM 001320  | 20/21  | 31/01  | 34/20  | 34/33  | 35/03  | 35/25  | 35/45  |        |        |        |        |        |
| SETUP 000005  | 7/08   | 34/10  | 39/06  | 39/23  | 39/40  | 40/04  | 40/39  |        |        |        |        |        |
|               | 42/02  | 43/06  | 45/06  | 45/18  | 45/30  | 45/42  | 46/04  |        |        |        |        |        |
|               | 46/28  | 47/01  | 48/03  | 48/24  | 49/01  | 49/18  | 49/35  |        |        |        |        |        |
|               | 50/01  | 50/21  | 50/44  | 51/06  | 51/18  | 52/03  | 52/14  |        |        |        |        |        |
|               | 53/03  | 53/20  | 54/02  | 54/24  | 55/02  | 56/04  | 56/22  |        |        |        |        |        |
|               | 57/01  | 57/20  | 58/03  | 58/16  | 59/01  | 59/17  | 60/03  |        |        |        |        |        |
|               | 60/25  | 61/04  | 61/22  | 61/48  | 62/01  | 63/01  | 64/03  |        |        |        |        |        |
|               | 65/01  | 66/03  | 67/06  | 67/31  | 68/03  | 68/24  | 68/44  |        |        |        |        |        |
|               | 69/04  | 69/26  | 69/46  | 70/06  | 70/36  | 71/01  | 71/16  |        |        |        |        |        |
|               | 71/29  | 72/01  | 72/18  | 73/03  | 73/27  | 74/17  | 75/03  |        |        |        |        |        |
|               | 75/25  | 76/01  | 77/02  | 77/28  | 78/03  | 78/24  | 78/35  |        |        |        |        |        |
|               | 79/03  | 79/34  | 80/03  | 80/18  | 80/36  | 81/03  | 81/30  |        |        |        |        |        |
|               | 82/03  | 83/24  | 85/03  | 86/01  | 86/31  | 87/06  | 88/01  |        |        |        |        |        |
|               | 89/03  | 90/01  | 91/03  | 91/28  | 92/03  | 92/18  | 92/34  |        |        |        |        |        |

SIZE 001001

START 000563

STLOC 000531

SUBAL 000053 MC

SWRES 000534

SMREG 007237 MC

XCHAC 000172 MC

XORAC 000132 MC

ZERDA 000042 MC

.8C0 000245

.8C1 000244

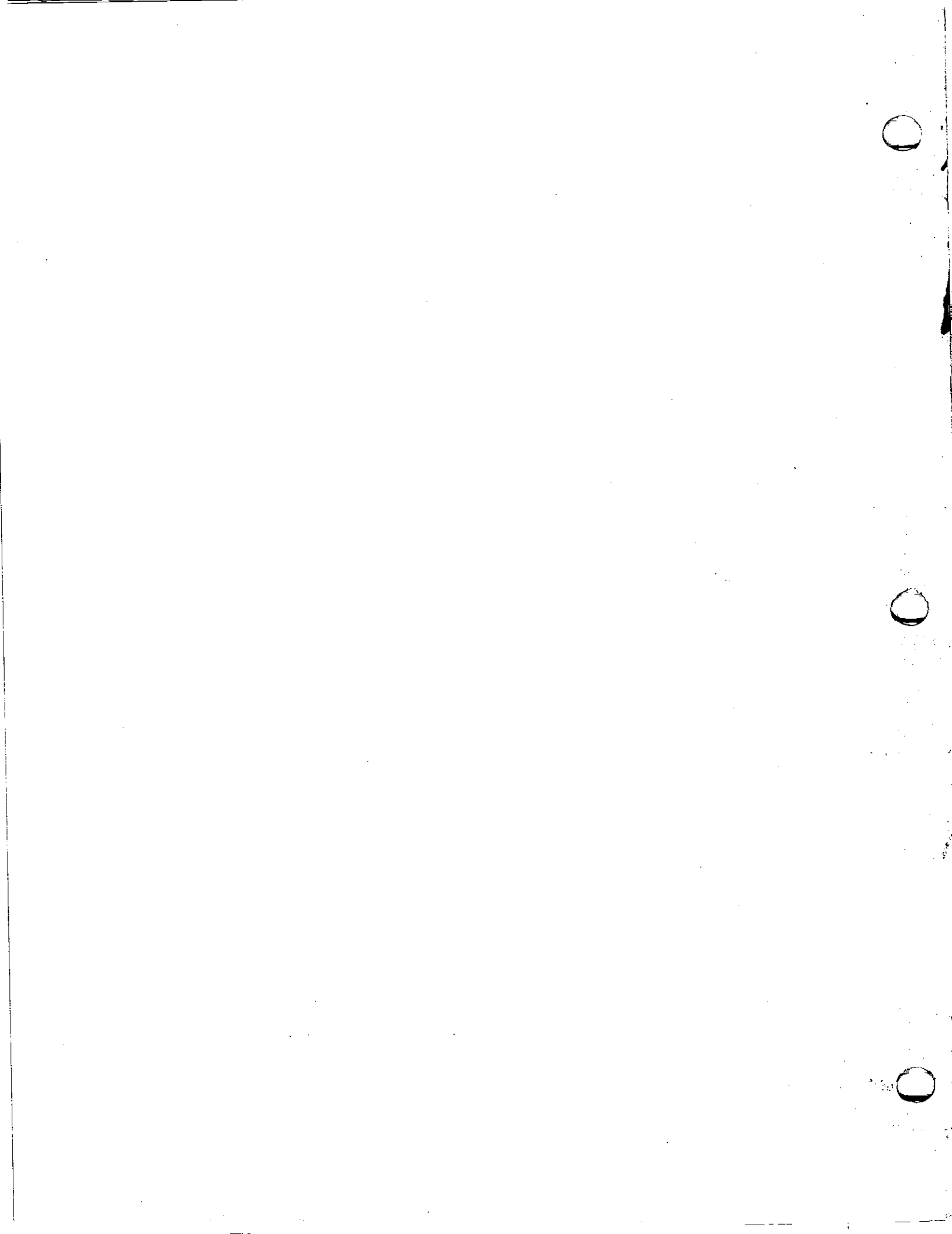
.8C2 000243

.8C3 000242

.EGGS 000010-

.RAND 000751





**DataGeneral**

---

---

**DIAGNOSTIC  
LISTING**

---

---

LISTING

096-000240-03

PROGRAM

EXERCISER FOR ECLIPSE  
PART 2

TAPE

095-000225-03

ABSTRACT

'ECLIPSE11' IS AN EXERCISER PROGRAM USED TO TEST THE RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF THE ECLIPSE COMPUTER. 'ECLIPSE11' EXERCISES THE EXTENDED INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES OF ITS RELIABLE OPERATION.

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





01 0001 ECL111 MACRO REV 03.00 14:34:19 08/08/76  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

TITLE ECL11  
ECLIPSE11  
ECLIPSE11 - CONTINUATION OF ECLIPSE11  
PART 2 OF EXERCISER FOR ECLIPSE

\*\*\*\*\*  
NAME: ECLIPSE11.SR PART NUMBER: 094-080624  
\*\*\*\*\*  
DESCRIPTION: ECLIPSE EXERCISER, PART 2  
REVISION HISTORY:  
REV. DATE  
00 08/02/74  
01 12/20/74  
02 04/11/75  
03 08/06/76  
\*\*\*\*\*  
COPYRIGHT (C) DATA GENERAL CORPORATION, 1974, 1975, 1976  
ALL RIGHTS RESERVED.  
\*\*\*\*\*

```

10003 ECL11
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

EXERCISER FOR ECLIPSE: PART 2
PROGRAM NAME
-----
ECLIPSE11
GENERAL DESCRIPTION
-----
ECLIPSE11 IS AN EXERCISER PROGRAM USED TO TEST THE RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF THE ECLIPSE COMPUTER. ECLIPSE11 EXERCISES THE EXTENDED INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES OF ITS RELIABLE OPERATION.

THE INSTRUCTIONS EXERCISED ARE AS FOLLOWS:
LXB,HXL,HXR,DHXL,DHXR,DAD AND DSB
LOCATIONS 200 TO 205 IN PAGE 0 ARE FIXED FOR ECLIPSE11 PROGRAM.
LOCATION 203 KEEPS TRACK OF NUMBER OF PASSES RUN THROUGH ECLIPSE11 PROGRAM.
LOCATION 201 KEEPS TRACK OF THE TEST RUNNING AT PRESENT AND IS USEFUL FOR DEBUG WHEN LOOPING OCCURS IN THE PROGRAM.
LOCATION 202 CONTAINS THE STARTING ADDRESS OF ECLIPSE11 PROGRAM.
LOCATION 200 IS USED BY DTOS.
LOCATION 204 KEEPS TRACK OF INTERNAL PASS COUNT WHICH IS FIXED BY LOCATION 205.

FIRST PASS THROUGH ECLIPSE11 TEST WILL RUN SUPERFAST. NEXT PASSES WILL RUN SLOWER AS EACH TEST IS RUN SEVERAL TIMES TO RUN ALL RANDOM NUMBER COMBINATIONS.

MACHINE REQUIREMENTS
-----
ECLIPSE PROCESSOR
4K READ-WRITE MEMORY
CONSOLE EQUIPMENT

10004 ECL11
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

SWITCH SETTINGS
-----
THIS PROGRAM USES DATA SWITCHES AS FOLLOWS
SW"0" - USE COMMENTS OF "SMREG" IF 0
SW"1" - USE DATA SWITCHES IF 1
SW"2" - LOOP ON FAILING TEST IF 0
SW"3" - PROCEED TO NEXT TEST IF 1
SW"4" - INHIBIT PRINTING TO TTY IF 0
SW"5" - DO NOT PRINT X ERRORS IF 0
SW"6" - PRINT FAILURE RATE IF 1
SW"7" - INHIBIT PRINTING PASS COUNT IF 1
SW"8" - INHIBIT OUTPUT TO LINE PRINTER IF 0
SW"9" - OUTPUT TO LINE PRINTER IF 1

EDTOS PRESUMES THAT ALL SWITCHES ARE 0. RETURN IS MADE TO THE OPER SYS UPON AN ERROR AFTER FIRST PRINTING THE FAILURE REPORT. IN STAND ALONE MODE, THE OPERATOR IS ASKED TO PRESET THESE SWITCHES (THE MACHINE HALTS FOR HIM TO DC SU) AND PRESS CONTINUE. THIS PRESET VALUE CAN BE CHANGED AT ANY TIME BY SETTING DATA SWITCH "0" TO A ONE. THE PROGRAM WILL THEN FOLLOW ACTUAL SETTINGS.

STAND ALONE STARTING ADDRESS = 200
IF 'ICAT1' OR 'KITTEN' WAS LOADED FROM DTOS AND RESTART WAS NEEDED, THEN USE AS FOLLOWS:
STARTING ADDR = 170 (FOR START WITH NO 'ICAT1')
STARTING ADDR = 171 (FOR START WITH 'ICAT1')

MONITOR LOCATION 200 TO CHECK THE CURRENT PASS COUNT
MONITOR LOCATION X60000 TO MAKE SURE THAT 'ICAT1' OR 'KITTEN' IS RUNNING. IN CASES WHERE PROGRAM IS STARTED WITH 'ICAT1' OR 'KITTEN' LOCATION X60000 WILL SHOW A PATTERN CHANGING FROM ZEROS TO ALL ONES TO AN INC/SWAP PATTERN.

(X= THE NUMBER OF THE HIGHEST MEMORY MODULE IN THE SYSTEM AND MAY BE A VALUE 0 - 7)

```

```

10003 ECL11
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

EXERCISER FOR ECLIPSE: PART 2
PROGRAM NAME
-----
ECLIPSE11
GENERAL DESCRIPTION
-----
ECLIPSE11 IS AN EXERCISER PROGRAM USED TO TEST THE RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF THE ECLIPSE COMPUTER. ECLIPSE11 EXERCISES THE EXTENDED INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES OF ITS RELIABLE OPERATION.

THE INSTRUCTIONS EXERCISED ARE AS FOLLOWS:
LXB,HXL,HXR,DHXL,DHXR,DAD AND DSB
LOCATIONS 200 TO 205 IN PAGE 0 ARE FIXED FOR ECLIPSE11 PROGRAM.
LOCATION 203 KEEPS TRACK OF NUMBER OF PASSES RUN THROUGH ECLIPSE11 PROGRAM.
LOCATION 201 KEEPS TRACK OF THE TEST RUNNING AT PRESENT AND IS USEFUL FOR DEBUG WHEN LOOPING OCCURS IN THE PROGRAM.
LOCATION 202 CONTAINS THE STARTING ADDRESS OF ECLIPSE11 PROGRAM.
LOCATION 200 IS USED BY DTOS.
LOCATION 204 KEEPS TRACK OF INTERNAL PASS COUNT WHICH IS FIXED BY LOCATION 205.

FIRST PASS THROUGH ECLIPSE11 TEST WILL RUN SUPERFAST. NEXT PASSES WILL RUN SLOWER AS EACH TEST IS RUN SEVERAL TIMES TO RUN ALL RANDOM NUMBER COMBINATIONS.

MACHINE REQUIREMENTS
-----
ECLIPSE PROCESSOR
4K READ-WRITE MEMORY
CONSOLE EQUIPMENT

```

10000 ECL11  
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

OPERATING PROCEDURE/OPERATOR INPUT  
-----  
LOAD THE PROGRAM VIA THE BINARY LOADER OR INSERT A  
PRELOADED MEMORY MODULE.  
SET SWITCHES TO 200.  
PROGRAM WILL HALT AFTER PRINTING THE MESSAGE  
"SET DATA SWITCHES AND PRESS CONTINUE".  
SET REQUIRED SWITCH SETTINGS AND PRESS CONTINUE.  
THE PROGRAM WILL RUN UNTIL MANUALLY STOPPED. IN CASE  
OF MALFUNCTIONING, THE PROGRAM WILL PRINT ERROR  
MESSAGE AND TAKE APPROPRIATE ACTION AS PER THE SW  
SETTINGS.  
PROGRAM OUTPUT/ERROR DESCRIPTION  
-----  
FOR ANY ERRORS DETECTED, THE PROGRAM WILL PRINT ERROR  
REPORT OR % FAILURES DEPENDING UPON THE SW SETTINGS.  
ERROR REPORT CONSISTS OF ALL ACCUMULATORS, CARRY,  
RELOCATED PROGRAM COUNTER OF THE TEST THAT IS FAILING  
AND PC IN THE LISTING AT THE TIME OF FAILURE.  
THE PROGRAM WILL LOOP IN THE TEST THAT IS FAILING IF  
SW"1" IS 0.  
SW"2" TO 1.  
IF LOOPING OCCURS IN THE PROGRAM, STOP THE COMPUTER  
AND CHECK LOCATION 201 TO FIND OUT THE TEST THAT WAS  
RUNNING BEFORE THE LOOPING OCCURRED.

10000 ECL11  
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

PROGRAM DESCRIPTION/THEORY OF OPERATION  
-----  
EACH TEST IS COMPLETE IN ITSELF, SO THE PROGRAM CAN  
BE STARTED FROM ANY TEST WITHOUT CAUSING ANY  
INITIALIZATION ERRORS.  
WHEN "ECLIPSE11" IS STARTED AT LOCATION 200 OR BY  
DTOS, IT WILL SIZE UP THE MEMORY AND WILL PRINT  
THE TOP OF THE MEMORY.  
AFTER SETTING UP THE SWITCHES AND PRESSING CONTINUE,  
THE EXERCISE WILL RUN THE FIRST PASS VERY FAST. IN  
THE FIRST PASS EACH TEST IS RUN ONLY ONCE. ALL OTHER  
PASSES WILL TAKE MORE TIME AS EACH TEST IS RUN  
ACCORDING TO THE TEST COUNTY SPECIFIED IN EACH TEST.  
AFTER THE 1ST PASS, "ECLIPSE11" IS RELOCATED IN THE  
AVAILABLE MEMORY FOR ALL NEXT PASSES AND THE AREA  
FOR SCRATCH BUFFER AREA. REFER TO THE LISTING TO  
FIND OUT THE INFORMATION ABOUT EACH TEST.  
RESTRICTIONS/MISC  
-----  
CERTAIN INSTRUCTIONS LIKE BLM, XCT, RAM, ETC.,  
DO NOT INTERRUPT TO OCCUR DURING THEIR  
EXECUTION. THIS FEATURE OF THOSE INSTRUCTIONS IS  
NOT CHECKED IN THIS TEST.

```

10007 ECL11
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

?
? *****GLOBAL MACROS*****
.MACRO LOOP
JSR #ENTLO
ITERATE TEST ROUTINE
X

.MACRO SETUP
JSR #ENTIN
INITIALIZE TEST.
A1
X

.MACRO RAND
JSR #ENTRA
R(A0)=RANDOM N
X

.MACRO CALL
JSP #ICAL
ICALL SUBROUTINE A1
A1
X

.MACRO ENKOR
JMP A+2
**
JMP A+3
**
STA B,AC3
**
JSR #ENTEX
X

.MACRO JAPER
LVA S,ITRER
MOV B,3,SZR
JMP A1 IJMP TO A1
X

?
? *****LOCAL MACROS*****
.MACRO ZEROAC
SUB 0,0
ICLEAR ALL AC'S
AND C(CARRY)
X

.MACRO SUBAL
SBI A1,0
SBI A1,1
SBI A1,2
SBI A1,3
SUBTRACT FROM ALL AC'S
X

.MACRO ADDAL
ADI A1,0
ADI A1,1
ADI A1,2
ADI A1,3
ADD TO ALL AC'S
X

.MACRO ASSEND
INC 0,1
INC 1,2
INC 2,3
ISEY C(AC1) TO 1 GREATER
ITHE C(AC0), ETC.
X

.MACRO ACO
ZEROAC
ADC A2,A2
A1 A2,A2
ITHE VALUE IN AC A2
ISHOULD NOT EFFECT
ADDM A2+1&3,A2+2&3,SMW
MOVW A2+3&3,A2+3&3,SZR
IOTHER AC'S - CHECK
IAC DESTINATION FOR
ERROR
X

```



10009 ECL11

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38

```

.MACRO XORACD
ZERDAD
ADC A1,A1
XOR A1,A1
ADD# A1+183,A1+283,SNK
MOV# A1+383,A1+383,SKZ
ERROR
X

.MACRO ANEACD
ZERDAD
ADC A1,A1
ADD# A1+183,A1+283,SNK
MOV# A1+383,A1+383,SKZ
ERROR
X

.MACRO XCRACD
ZERDAD
XCH A1,A1
ADD# A1+183,A1+283,SNK
MOV# A1+383,A1+383,SKZ
ERROR
X

.MACRO LSHACD
ZERDAD
LSH A1,A1
ADD# A1+183,A1+283,SNK
MOV# A1+383,A1+383,SKZ
ERROR
X

```

```

;TEST THAT XOR RESULT
;GOES ONLY TO C(ACD)

;TEST THAT ANE RESULT
;GOES ONLY TO C(AC A1)

;TEST THAT XCH RESULT
;GOES ONLY TO C(AC A1)

;TEST THAT LSH RESULT
;GOES ONLY TO C(AC A1)

```

10010 ECL11

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38

```

.MACRO LORACD
ANC 0,P
ANC 1,1
ANC 2,2
ANC 3,3
ADC A1,A1
LOR A1,A1
ADD# A1+183,A1+283,SNK
MOV# A1+383,A1+383,SKZ
ERROR
X

.MACRO LORAL
SUBZL A1,A1
SUB A1+283,A1+283,SKP
120,A1,A1,SKZ
LOR A1,A1+283
MOVZL A1,A1,SKZ
JMP *-2
LDA A1+383,-4
SUB# A1+283,A1+383,SKZ
EMRUR
X

.MACRO LORACD
XOR 0,0
XOR 1,1
XOR 2,2
XOR 3,3
ADC A1,A1
LOR A1,A1
ADD# A1+183,A1+283,SNK
MOV# A1+383,A1+383,SKZ
ERROR
X

```

```

;TEST THAT LOR RESULT
;GOES ONLY TO AC A1

;RESET ACIS TO 0

;RESULT TO AC A1

;THE "LOR INSTRUCTION WILL
;FIND 15 ZEROS THEN 14
;ETC. THESE VALUES SHOULD
;FORM THE SUM 120 IN C(ACO)

;(AC A1+2)=LOR RESULT
;LOR FAILED

;TEST THAT LOR RESULT
;GOES ONLY TO AC A1

;RESET AC0-3 TO ZERO

;RESULT TO AC A1

```

10011 ECL11

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39

```
.MACRO
COBACD
ANC 0,0
ANC 1,1
ANC 2,2
ANC 3,3
COM 1,1
COB 1,1
ADD# 1,183,1+283,SNR
MOV# 1,383,1+383,SZR
ERROR
X
```

```
.MACRO
HSLACD
ANC 0,0
ANC 1,1
ANC 2,2
ANC 3,3
ADC 1,1
HXL 1,1,1
ADD# 1,183,1+283,SNR
MOV# 1,383,1+383,SZR
ERROR
X
```

```
.MACRO
HSRACD
XOR 0,0
XOR 1,1
XOR 2,2
XOR 3,3
ADC 1,1,1
HXR 1,1,1
ADD# 1,183,1+283,SNR
MOV# 1,383,1+383,SZR
ERROR
X
```

```
.MACRO
JTEST THAT COB RESULT
JGOES ONLY TO AC 1
JRESET AC0-3 TO ZERO
JRESULT TO AC 1
X
```

```
.MACRO
JTEST THAT HSL RESULT
JGOES ONLY TO AC 1
JRESET AC0-3 TO ZERO
JRESULT TO AC 1
X
```

```
.MACRO
JTEST THAT HSR RESULT
JGOES ONLY TO AC 1
JRESET AC0-3 TO ZERO
JRESULT TO AC 1
X
```

10012 ECL11

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42

```
.MACRO
DUBL
SETUP 5
ADC 1,1
SUB 1,183,1+183
XOR 1,283,1+283
ANC 1,383,1+383
A2 4,1
ADD# 0,1,SNR
ADD# 2,3,SZR
ERROR
LOOP
X
```

```
.MACRO
LRSHIFT
SETUP 40
RAND
MOV 0,1
MOV 0,2
MOV 0,3
DXX 4,1
DXX 4,1
MOV 1,183,1+183,SNR
SUB# 1,1,283,SZR
ERROR
LOOP
X
```

```
.MACRO
DLB
SETUP 40
RAND
MOV 0,1
MOV 1,183
MOV0S 1,1,383
DXX 2,1
SUB# 1,1,383,SNR
MOV 0,0,SNR
ERROR
LOOP
X
```

```
JTEST "A2"
JSET AC 1 TO 17777
JOTHER AC'S SET TO 0
J#2 SHOULD CLEAR AC 1
JALL AC'S SHOULD NOW
JBE CLEARED
```

```
JTEST "DMSL/DMSR"
J(CAC 1) SHOULD NOT CHANGE
JDOUBLE SHIFT 16 PLACES
JRIGHT THEN LEFT
```

```
JSHIFT DOUBLE LEFT 8
JTEST THE HIGH ORDER PART
JOF A DOUBLE SHIFT LEFT
J8 PLACES TO AC 1
J(CARRY) SHOULD NOT CHANGE
JMSL FAILED
```

10013 ECL11

```

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

```

.MACRO DSR6 SETUP 40  
 RANU  
 MOV 0,A1  
 MOV A1,A1+183  
 MOVZS A1,A1+383  
 DMXR 2,A1  
 SUB# A1+183,A1+383,SNR  
 MOV 0,0,SZC  
 ERROR  
 LOOP

.MACRO DSR6 SETUP 40  
 RANU  
 MOV 0,A1  
 LDA A1+283,+2  
 ANDS A1,A1+283,SKP  
 17760  
 MOV A1,A1+383  
 MOV A1,A1+183  
 DMXR 2,A1  
 SUB# A1+183,A1+383,SNR  
 ERROR  
 LOOP

.MACRO DSR6 SETUP 40  
 RANU  
 MOV 0,A1  
 LDA A1+283,+2  
 ANDS A1,A1+283,SKP  
 17760  
 MOV A1,A1+383  
 MOV A1,A1+183  
 DMXR 2,A1  
 SUB# A1+183,A1+383,SNR  
 ERROR  
 LOOP

!DOUBLE SHIFT RIGHT 8

!TEST THE LOW ORDER PART  
!OF THE DOUBLE SHIFT

!8 PLACE DOUBLE SHIFT

!A1=8  
!A1+2=0,A  
!A1+3=8A  
!A1+4=8  
!SHIFT A1 RIGHT 8 TIMES

!TEST HIGH PART

10014 ECL11

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

```

.MACRO DSR4 SETUP 20  
 RANU  
 MOV 0,A1  
 LDA A1+283,+2  
 ANDS A1,A1+283,SKP  
 377  
 MOV A1,A1+383  
 MOV A1,A1+183  
 DMXR 2,A1  
 SUB# A1+183,A1+383,SNR  
 ERROR  
 LOOP

.MACRO DSR4 SETUP 10  
 ADDU A1+183,A1+183,SKP  
 17760  
 LDA A1,+1  
 DMXR 1,A1  
 SUB# A1,A1+183,SNR  
 MOV 0,0,SZC  
 ERROR  
 LOOP

.MACRO DSR4 SETUP 10  
 ADDU A1+183,A1+183,SKP  
 17760  
 LDA A1,+1  
 DMXR 1,A1  
 SUB# A1,A1+183,SNR  
 MOV 0,0,SZC  
 ERROR  
 LOOP

!TEST "DMSL" 8 TIMES LEFT

!A1=8  
!A1+2=8,0  
!A1+3=8,A  
!A1+4=8,8  
!A1+5=8,0  
!A1+6=8,A,+1+1=8,0

!DMSL FAILED

!TEST "DMSL" 4 TIMES LEFT

!7777 SHIFTED LEFT WILL  
!EQUAL 1777400, SAME AS  
!HIGH ORDER PART

!C(CARRY) SHOULD NOT CHANGE  
!DMSL FAILED

!TEST "DMSR" 4 TIMES RIGHT

!177760 SHIFTED RIGHT=7777  
!ALL ONES SHIFTED RIGHT  
!ALSO EQUALS 7777

!C(CARRY) SHOULD NOT CHANGE  
!DMSR FAILED

```

10W15 ECL11
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

;TEST DMSK/DMSL
;SHIFT RIGHT 12 PLACES
;THEN SHIFT LEFT 12

;CHECK HIGH PART
;CHECK LOW PART

;TEST DMSL/DMSR
;CHECK HIGH PART
;CHECK LOW PART
;SHIFTING 12 FAILED

.MACRO DS12
  SETUP 2#
  RAND
  MOV 0,1
  MOV 1,1+183
  MOV 1+183,1+283
  DMX 3,1
  LDA 1+383,+2
  AND 1+283,1+383,SKP
  1/RAND
  SUB# 1,1+283,SNR
  SUB# 1+183,1+383,SZR
  ERROR
  LOOP
*

.MACRO DSL12
  SETUP 2#
  RAND
  MOV 0,1
  MOV 1,1+183
  MOV 1+183,1+283
  DMX 3,1
  LDA 1+383,+2
  AND 1+283,1+383,SKP
  1/RAND
  SUB# 1,1+383,SNR
  SUB# 1+183,1+283,SZR
  ERROR
  LOOP
*

.MACRO DS11
  SETUP 5
  ZEROAC
  ADCZL 1,1
  DAD 1,1
  ADD 1+183,1+283,SNR
  MOV# 1+383,1+383,SZR
  ERROR
  LOOP
*

.MACRO DSBACD
  ZEROAC
  SUBZL 1,1
  DSH 1,1
  ADD# 1+183,1+283,SNR
  MOV# 1+383,1+383,SZR
  ERROR
*

.MACRO DS81
  SETUP 10
  SUBZ 1,1
  DSB 1,1
  MOV 1,1,SNR
  MOV 1,1,SZR
  ERROR
  LOOP
*

;DECIMAL ADD
;TEST THAT RESULT GOES TO 1, ON

;DECIMAL SUBTRACT
;TEST THAT RESULT GOES
;TO AC 1, ONLY

;CLEAR AC 1, SET C(CARRY).
;DECIMAL SUBTRACT SHOULD
;GIVE 0 RESULT AND SET
;C(CARRY). C(AC 1) NOT
;0 OR C(CARRY) IS 0

```

```

10W16 ECL11
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

```







```

10021 ECL11
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

; *****TEST UTILITY SUBROUTINES*****
; SUBROUTINE TO INITIALIZE A TEST LOOP
INIT: INC J,3
STA 3,ITRET
STA 0,ALC2
LCA 0,1,3
STA 0,ITK
STA 0,ITRET
LDA 0,RELUC
SUB 0,3
STA 3,LISTNG
SUM J,3
STA 3,ITRER
STA 3,ITREC
LDA 3,PASS
MOV 3,3,SZR
JMP IN11
SUB2L 3,3
STA 3,ITR
STA 3,ITRCT
LDA 0,ACR
JMP 0,ITRET

; SUBROUTINE TO TERMINATE A TEST LOOP
LUP1: STA 3,LOPRET
DSZ ITRET
JMP LOP3
LDA 3,ITRER
MOV 3,3,SNR
JMP 0,LOPRET
LDA 3,ITR
STA 3,ITRET
HEADS 3
MOVLW 3,3,SZC
JMP 3,3
ELDA 3,3,SMREG
ADDL 3,3
ADDL 3,3,SNC
JMP LOP2
STA 2,ACR
STA 1,AC1
STA 2,AC2
JSR #INER
PERCENT 0,0
LDA 1,ITREC
LDA 0,ITREC
LDA 2,100,
MUL 3,0
LDA 2,ITR
DIV #1,DEC
LDA 2,ACB
LDA 1,AC1
LDA 2,AC2
LDA 3,3
SUB 3,3
STA 3,ITREC
LUP3: LDA 3,ITRER
MOV 3,3,SZR
HEADS 3
MOVLW 3,3,SZC
JMP 3,3
ELDA 3,3,SMREG
ADDLW 3,3,SNC
JMP 3,3,ITRET
LOPRET

```

```

10022 ECL11
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

; SUBROUTINE TO TERMINATE A TEST LOOP
LUP1: STA 3,LOPRET
DSZ ITRET
JMP LOP3
LDA 3,ITRER
MOV 3,3,SNR
JMP 0,LOPRET
LDA 3,ITR
STA 3,ITRET
HEADS 3
MOVLW 3,3,SZC
JMP 3,3
ELDA 3,3,SMREG
ADDL 3,3,SNC
MOVLW 3,3
ELDA 3,3,SMREG
PERCENT 0,0
LDA 1,ITREC
LDA 0,ITREC
LDA 2,100,
MUL 3,0
LDA 2,ITR
DIV #1,DEC
LDA 2,ACB
LDA 1,AC1
LDA 2,AC2
LDA 3,3
SUB 3,3
STA 3,ITREC
LUP2: LDA 3,ITRER
MOV 3,3,SZR
HEADS 3
MOVLW 3,3,SZC
JMP 3,3
ELDA 3,3,SMREG
ADDLW 3,3,SNC
JMP 3,3,ITRET
LOPRET

```

```

IF NO ERROR, ITERATE
OTHERWISE LOOK AT DATA
IF "1" SWITCH FOR PROCEED,
FOR NOT.

```

10223 ECL11

```

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

; ERROR SUBROUTINE: SETS ITRER FLAG, PRINTS MACHINE STATE.
;SAVE PC OF ERROR
;SAVE TEMPORARY
;ADC0 AND CARRY
;BUMP ERROR COUNT
;NEW ERROR?
;NO, RETURN.
YES,
;SAVE MACHINE STATE
;PRINT "ERRRR PASS" MSG.
;PRINT HEADER
;SET LEADING ZERO SUPR.
;PRINT LOGICAL &
;LISTING PC'S OF ERRDR
;(LISTING PC)
;(LOGICAL PC)
;FLUSH DEVICE BUFFERS
;RESTORE AC'S 1 & 2
;PROGRAM IN AUTO MODE?
;NOPE: GO BACK TO TEST
;AUTO MODE & SM6#0
;MAKE ERROR RETURN TO DT05
;FINISH RESTORING AC'S
;GO BACK TO TEST

STA 3,ERRR1
STA 0,AC2
STA 0,0
STA 0,CRY
ISZ ITRER
LDA 0,ITRER
ADCC# 0,3,SNR
ERR1
ERR1
STA 3,ITRER
STA 1,AC1
STA 2,AL2
JSR 0,EMSG
LDA 1,PASS
INCZ 1,1
JSR 0,IPDEC
JSR 0,EMSG
HEADER
MOVZ 0,0
LDA 1,CRY
JSR 0,IPDEC
MOVU 0,0
LDA 1,ACR
JSR 0,IPDEC
LDA 1,AC1
JSR 0,IPDEC
LDA 1,AC2
JSR 0,IPDEC
LDA 1,AC3
JSR 0,IPDEC
LDA 1,ERRR1
LDA 0,RELOC
SUBZ 0,1
JSR 0,IPDEC
LDA 1,ERRR1
JSR 0,IPDEC
KCLF
LDA 1,ERRR1
LDA 2,AC2
ELOA 3,AUTO
MOV 3,3,SNR
ERR1
JONST
LDA 3,EMSG
LDA 3,4,3
JMP 0,3
LDA 0,AC2
LDA 3,AC3
ERR1
JMP 0,ERRR1

```

10224 ECL11

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

; RANDOM NUMBER GENERATOR SUBROUTINE
;GENERATE A NEW RANDOM
;NUMBER IN C(AC0) AND C(RAN),
;IF C(ITRER)=0, OTHERWISE
;LOAD C(AC0) WITH OLD R.
;HAND: LDA 0,RAN
;ITRER
;1,1,ISZR
MOV 0,3
MOV 0,1
MOV 0,2,1
XL 2,1
ADD 0,1
MOVZ 1,1
MOVZ 1,1
ADD 1,0
LDA 1,33R31
ADD 1,0
STA 0,RAN
JMP 0,3
; CALL ROUTINE TO REACH RELOCATED SUBROUTINES
CAL: STA 0,CAL0
STA 1,CAL1
0,RELOC
LDA 1,0,3
ADD 1,0
STA 0,CAL2
INC 3,3
LDA 0,CAL0
LDA 1,CAL1
JMP 0,CAL2
; SIZING SUBROUTINE: RETURNS SIZE OF LOGICAL MEM IN AC2
;SIZE THE LOGICAL MEMORY
LDA 2,MINLOC
INC 2,2
MOVL# 2,2,ISZ
JMP +5
LDA 0,0,2
STA 2,0,2
LDA 1,0,2
STA 0,0,2
SUM# 1,2,ISZ
JMP 0,3
;SAVE MEMORY TOP ADDR
SIZE+1
JMP

```

10225 ECL11

```

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

; ERROR SUBROUTINE: SETS ITRER FLAG, PRINTS MACHINE STATE.
;SAVE PC OF ERROR
;SAVE TEMPORARY
;ADC0 AND CARRY
;BUMP ERROR COUNT
;NEW ERROR?
;NO, RETURN.
YES,
;SAVE MACHINE STATE
;PRINT "ERRRR PASS" MSG.
;PRINT HEADER
;SET LEADING ZERO SUPR.
;PRINT LOGICAL &
;LISTING PC'S OF ERRDR
;(LISTING PC)
;(LOGICAL PC)
;FLUSH DEVICE BUFFERS
;RESTORE AC'S 1 & 2
;PROGRAM IN AUTO MODE?
;NOPE: GO BACK TO TEST
;AUTO MODE & SM6#0
;MAKE ERROR RETURN TO DT05
;FINISH RESTORING AC'S
;GO BACK TO TEST

STA 3,ERRR1
STA 0,AC2
STA 0,0
STA 0,CRY
ISZ ITRER
LDA 0,ITRER
ADCC# 0,3,SNR
ERR1
ERR1
STA 3,ITRER
STA 1,AC1
STA 2,AL2
JSR 0,EMSG
LDA 1,PASS
INCZ 1,1
JSR 0,IPDEC
JSR 0,EMSG
HEADER
MOVZ 0,0
LDA 1,CRY
JSR 0,IPDEC
MOVU 0,0
LDA 1,ACR
JSR 0,IPDEC
LDA 1,AC1
JSR 0,IPDEC
LDA 1,AC2
JSR 0,IPDEC
LDA 1,AC3
JSR 0,IPDEC
LDA 1,ERRR1
LDA 0,RELOC
SUBZ 0,1
JSR 0,IPDEC
LDA 1,ERRR1
JSR 0,IPDEC
KCLF
LDA 1,ERRR1
LDA 2,AC2
ELOA 3,AUTO
MOV 3,3,SNR
ERR1
JONST
LDA 3,EMSG
LDA 3,4,3
JMP 0,3
LDA 0,AC2
LDA 3,AC3
ERR1
JMP 0,ERRR1

```

10025 ECL11

```

01 / RELOCATE SUBROUTINE: ALLOCATES MEMORY FOR COPIES
02 / OF TEST PROGRAM. WORKS IN CONJUNCTION
03 / WITH BAK ROUTINE IN PAGE ZERO, WHICH
04 / ACTUALLY COPIES THE TEST PROGRAM TO
05 / THE NEW LOCATION.
06 /
07 /
08 01014 112470 REL: 01014 112470
09 004662
10 01016 000402
11 01017 000012
12
13 01020 000266
14 01021 020217
15 01022 030316
16 01023 140433
17 01024 000425
18 01025 122422
19 01026 000777
20 01027 107002
21 01030 140433 REL0:
22 01031 000763
23 01032 020214
24 01033 122432
25 01034 000407
26 01035 100400
27 01036 132433
28 01037 000403
29 01040 024217
30 01041 102401
31 01042 107002
32 01043 044214 REL1:
33 01044 176002
34 01045 114710
35 01046 144710
36 01047 154710
37 01050 000247 REL2:
38
39 01051 000220
40 01052 001442

```

10026 ELL11

```

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

```

10026 ELL11

```

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

```

0027 ECL11  
01 01142 00E767  
02

JMP MESS1

10028 ECL11

```
01 ; LPT/TTO INTERFACE ROUTINE: CHARACTER PASSED IN AC0  
02  
03 01143 175100 CHAK: 3,3  
04 01144 054252 STA 3,CHARCT  
05 01145 050400 STA 2,CHRST  
06  
07 01146 074877 READS 3  
08 01147 175112 PUVL# 3,3,3ZC  
09 01148 000403 JMP *3  
10 01151 130470 ELDA 3,SMREG  
11 01153 004500 LDA 2,=22000  
12 01154 173400 AND 3,2  
13 01155 150120 ADDZL 2,2  
14 01156 153200 ADDCR REST  
15 01157 000430 JMP  
16 01158 034823-  
17  
18 01160 103004 LDA 3,=377  
19 01161 117250 ANDZS 0,3,SNR  
20 01162 000432 JMP REST  
21  
22 01163 103004 0,0,=SZR  
23 01164 000777 JMP *1  
24 01165 177600 ADDCR 3,3  
25 01166 020021- LDA 0,=211+400  
26 01167 102445 SUBD 3,0,SNR  
27 01170 000430 JMP CHAR4  
28  
29 01171 101340 MOVDS 3,0  
30 01172 010233 CHAR1: CHORZ  
31  
32 01173 151145 MGVZL# 2,2,SNR  
33 01174 000405 JMP CHAR2  
34 01175 001117 D0AS 0,LPT  
35 01176 000517 SKPBZ LPT  
36 01177 000777 JMP *-1  
37 01200 000217 NI0C LPT  
38  
39 01201 151133 MGVZL# 2,2,SNR  
40 01202 000405 JMP CHAR3  
41 01203 001111 D0AS 0,TTO  
42 01204 000511 SKPBZ TTO  
43 01205 000777 JMP *-1  
44 01206 000211 NI0C TTO  
45  
46 01207 175403 CHAR3: INC 3,3,SNR  
47 01210 000702 JMP CHAR1  
48 01211 030020- LDA 2,=212  
49 01212 142405 SUB 2,0,SNR  
50 01213 040233 STA 0,CHORZ  
51  
52 01214 030411 REST: LDA 2,CHRST  
53 01215 034252 LDA 3,CHARCT  
54 01216 175200 MOVDR 3,3  
55 01217 001400 JMP 0,3  
56  
57 01220 034233 CHAR4: J,CHOKZ  
58 01221 020017- LDA 0,=8  
59 01222 114410 TOR 0,3  
60 01223 020016- LDA 0,=240
```

0029 ECL11  
01 01224 000746  
02  
03 01225 000000 CHKS: 0

JFP CHAR1

TEMP SAVE FOR AC2

```
0000 ECL11  
01  
02  
03 *****MESSAGE DATA BLCK*****  
04 01226 000215 PASMES: .TXTE !<15><12>PASS I  
05 040520  
06 051523  
07 120240  
08 000000  
09 000000 PERLEN: .TXTE !<15><12>X FAIL:  
10 120240  
11 000000  
12 140311  
13 000275  
14 01240 000215 ERMSG: .TXTE !<15><12><15><12>ERROR IN PASS: I  
15 000215  
16 151365  
17 14722  
18 120322  
19 047311  
20 050240  
21 051501  
22 030123  
23 000240  
24 01252 000215 HEADER: .TXTE !<15><12><15><12>  
25 000215  
26 01254 151303 CHY AC0 AC2 AC3 LISTING LOGICAL<15><12>I  
27 000531  
28 141501  
29 120302  
30 040411  
31 130703  
32 000400  
33 141501  
34 120202  
35 040411  
36 031703  
37 140411  
38 051711  
39 144724  
40 040316  
41 140411  
42 040717  
43 141711  
44 140101  
45 000215  
46 000000  
47 01301 000215 MESIZ: .TXTE !<15><12>LAST LOGICAL ADDRESS:  
48 040714  
49 150123  
50 140240  
51 040717  
52 141711  
53 140101  
54 040640  
55 040104  
56 140722  
57 051523  
58 000275  
59 01315 000215 KCRLF: .TXTE !<15><12>I  
60 000000
```

```

0031 ECL11
01 *1317 142523 SETSW: .TATE ISET DATA SWITCHS AND PRESS CONTINUE1
02 126324
03 *44504
04 *44724
05 *51644
06 143727
07 141724
08 *51514
09 *46644
10 *42116
11 *50244
12 142722
13 *51523
14 141644
15 *47317
16 144724
17 *52516
18 *48365

10032 ECL11
01
02
03
04 *1341 *54242 BCS:
05 *1342 *50543
06 *1343 *44244
07 *1344 *46245
08 *1345 *34415-
09 *1346 *34014-
10
11 *1347 126401
12 *1350 124402 BCS1:
13 *1351 157403
14 *1352 124004
15 *1353 167405
16 *1354 *26244
17 *1355 197406
18 *1356 *24245
19 *1357 163407
20 *1360 167408
21 *1361 122423
22 *1362 143062
23
24 *1363 *24245 BCS2:
25 *1364 164612
26 *1365 123002
27 *1366 *44245
28 *1367 126546
29 *1370 111612
30 *1371 175044
31 *1372 *08756
32 *1373 125246
33 *1374 *24244
34 *1375 *36243
35 *1376 *02242
36
37
38

*****SUBROUTINES LOCAL TO THIS TEST*****
PBCU SUBTRACTION...
P(C(AC0)-C(AC1), RESULT IN
P(C(AC0), C(AC1-2) UNCHANGED
P(A BORROW WILL SET C(CARRY)).

STA 3,BC3
STA 2,BC2
P(AC0)-C(AC1), RESULT IN
P(C(AC0), C(AC1-2) UNCHANGED
P(A BORROW WILL SET C(CARRY)).
LDA 3,*17
LDA 2,*12

SUB 1,1,SKP
NEG 1,1
ADD 3,1
COB 1,1
AND 3,1
LDA 0,BC1
ADD 0,1
LDA 0,BC0
AND 3,0
AND 3,1
SUB2 1,0,SNC
ADD2 2,0

LUA 1,BC0
AND 3,1
ADD 1,0
STA 0,BC0
SUB1 1,1
DAXL 1,2
MOV 3,3,SZR
JMP BCS1
MOVR 1,1
LUA 1,BC1
LDA 2,BC2
JMP 0,BC3

P(C(AC1) DETERMINES BORROW

P(MASK THE DIGITS
P(DIGIT)9, FIX.

P(SAVE BORROW CONDITION.
P(ITERATE
P(EXIT, RESTORE C(AC1-2)
P(AND FIX C(CARRY))

```



```

10033 ECL11
01 01377 024014=HEXN:
02 01400 030015=
03 01401 054246
04 01402 150000
05 01403 117400
06 01404 136432
07 01405 120510
08 01406 105510
09 01407 050772
10 01410 000772
11 01411 002246
12
13 01412 054246
14 01413 050243
15 01414 044244
16 01415 034015=
17 01416 030014=
18 01417 040245
19 01420 024244
20 01421 107424
21 01422 102400
22 01423 103420
23 01424 123000
24 01425 142412
25 01426 142421
26 01427 101020
27 01430 103400
28 01431 024245
29 01432 104610
30 01433 126410
31 01434 111510
32 01435 175004
33 01436 000761
34 01437 024244
35 01440 030243
36 01441 002242

; CONVERT THE NUMBER IN
; C(CARRY) TO A DECIMAL OF
; NUMBER IN THE RANGE OF
; 70 TO 9.
LDA 1,12
LDA 2,17
STA 3,HEXRET
MOV 2,3
AND 0,3
SUB# 1,3,SZC
XOR 1,0
DNL 1,1
MOV 2,2,SZR
JMP HEXN,
JMP 0,HEXRET

;BCD ADDITION WITHOUT THE
;USE OF "DAD" INSTRUCTION
;C(CARRY)+C(AC1)
;RESULT TO C(AC0)
;C(AC1=2) UNCHANGED
;C(CARRY) FROM PREVIOUS OPERATION
;IS USED. C(CARRY) WILL
;ALSO HAVE CORRECT STATE
;AT EXIT TIME
STA 3,BC3
STA 2,BC2
STA 1,BC1
LDA 3,17
LDA 2,12
LDA 1,BC0
AND 3,0
ANDZ 3,0
ADD 1,0
SUB# 2,0,SNP
MOVZ 0,0
AND 3,0
LDA 1,BC0
INC 3,1
IOR 1,0
DNL 1,2
MOV 3,3,SZR
JMP BC1
LDA 1,BC1
LDA 2,BC2
JMP 0,BC3

;SUBTRACT 12,SET
;C(CARRY)
;SHIFT TO NEXT DIGIT
;RESTORE C(AC1=2)
SUBTRACT 12,SET
C(CARRY)
SHIFT TO NEXT DIGIT
RESTORE C(AC1=2)

```

```

10034 ECL11
01
02
03
04
05
06
07 01442 101000 BEGIN:
08
09
10 01443 000203
11 01444 000005
12 LBRACC 0
13 XOR 0,0
14 01445 100510
15 01446 124510
16 01447 150510
17 01450 174510
18 01451 102000
19 01452 102510
20 01453 130015
21 01454 175014
22
23 01461 100510
24 01462 124510
25 01463 150510
26 01464 174510
27 01465 120000
28 01466 120510
29 01467 157015
30 01470 101014
31
32
33 01475 100510
34 01476 124510
35 01477 150510
36 01500 174510
37 01501 152000
38 01502 152510
39 01503 103015
40 01504 125014
41
42
43 01511 100510
44 01512 124510
45 01513 150510
46 01514 174510
47 01515 170000
48 01516 170510
49 01517 107015
50 01520 151014
51
52
53 01525 000204

; ***** FIRST TEST *****
;
; *** LRB (LOCATE AND RESET LEAD BIT) ***
;
; INITIALIZE TEST.
SETUP 5
JSR 0,ENTIN
LBRACC 0
XOR 0,0
XOR 1,1
XOR 2,2
XOR 3,3
AND 0,0
LRB 0,0
ADD# 0+183,1+283,SNR
MOV# 0+383,0+383,SZR
EKOR
LBRACC 1
XOR 0,0
XOR 1,1
XOR 2,2
XOR 3,3
ADC 1,1
LRB 1,1
ADD# 1+183,1+283,SNR
MOV# 1+383,1+383,SZR
EKOR
LBRACC 2
XOR 0,0
XOR 1,1
XOR 2,2
XOR 3,3
AUC 2,2
LRB 2,2
ADD# 2+183,2+283,SNR
MOV# 2+383,2+383,SZR
EKOR
LBRACC 3
XOR 0,0
XOR 1,1
XOR 2,2
XOR 3,3
ADC 3,3
LRB 3,3
ADD# 3+183,3+283,SNR
MOV# 3+383,3+383,SZR
ERROR
LOOP
JSR 0,ENTLU
ITERATE TEST ROUTINE

;TEST THAT LRB RESULT
;Goes ONLY TO AC 0
;PRESET AC0=3 TO ZERO
;TEST THAT LRB RESULT
;Goes ONLY TO AC 1
;PRESET AC0=3 TO ZERO
;TEST THAT LRB RESULT
;Goes ONLY TO AC 2
;PRESET AC0=3 TO ZERO
;TEST THAT LRB RESULT
;Goes ONLY TO AC 3
;PRESET AC0=3 TO ZERO

```

|             |       |             |  |  |  |
|-------------|-------|-------------|--|--|--|
| 10035 ECL11 |       |             |  |  |  |
| 01          | EX3B: | SETUP 10    |  |  |  |
| 02          |       | JSR #ENTIN  |  |  |  |
| 03          |       | 10          |  |  |  |
| 04          |       | SUB# 0,0    |  |  |  |
| 05          |       | LRR 0,1     |  |  |  |
| 06          |       | MOV 0,0,SZC |  |  |  |
| 07          |       | MOV 0,0,SZR |  |  |  |
| 08          |       | ERROR       |  |  |  |
| 09          |       | LOOP        |  |  |  |
| 10          |       | JSR #ENTLO  |  |  |  |
| 11          |       |             |  |  |  |
| 12          |       |             |  |  |  |
| 13          | EX3C: | SETUP 10    |  |  |  |
| 14          |       | JSR #ENTIN  |  |  |  |
| 15          |       | 10          |  |  |  |
| 16          |       | SUB# 1,1    |  |  |  |
| 17          |       | LRR 1,2     |  |  |  |
| 18          |       | MOV 0,0,SNC |  |  |  |
| 19          |       | MOV 1,1,SZR |  |  |  |
| 20          |       | ERROR       |  |  |  |
| 21          |       | LOOP        |  |  |  |
| 22          |       | JSR #ENTLO  |  |  |  |
| 23          |       |             |  |  |  |
| 24          | EX3D: | SETUP 10    |  |  |  |
| 25          |       | JSR #ENTIN  |  |  |  |
| 26          |       | 10          |  |  |  |
| 27          |       | SUB# 2,2    |  |  |  |
| 28          |       | LRR 2,3     |  |  |  |
| 29          |       | MOV 0,0,SZC |  |  |  |
| 30          |       | MOV 2,2,SZR |  |  |  |
| 31          |       | ERROR       |  |  |  |
| 32          |       | LOOP        |  |  |  |
| 33          |       | JSR #ENTLO  |  |  |  |
| 34          |       |             |  |  |  |
| 35          |       |             |  |  |  |
| 10036 ECL11 |       |             |  |  |  |
| 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          |       |             |  |  |  |

/TEST "LRR"  
 /INITIALIZE TEST.  
 /SEE TEST EX3B

SETUP 10  
 JSR #ENTIN  
 10  
 SUB# 3,3  
 LRR 3,0  
 MOV 0,0,SZC  
 MOV 3,3,SZR  
 ERROR  
 LOOP

EX3E:

04 01507 006263  
 05 01570 006017  
 06 01571 170427  
 07 01572 162512  
 08 01573 101002  
 09 01574 175004  
 10  
 11

/ITERATE TEST ROUTINE

JSR #ENTLO  
 SETUP 40  
 JSR #ENTIN  
 40  
 I=0  
 -DO 3  
 SUB# 1,1  
 LRR 1,3  
 MOV 1,1,SZR  
 ERROR  
 LOOP

EX3F:

15 01002 006263  
 16 01003 006047  
 17 01004 006003  
 18 01004 102520  
 19 01004 102520  
 20 01005 110512  
 21 01006 101004  
 22

/SET BIT 15  
 /LRR SHOULD RESET IT.  
 /C(ACS) SHOULD BE=0

SUR# 1,1  
 LRR 1,3  
 MOV 1,1,SZR  
 ERROR  
 LOOP

EX3G:

23 01013 120520  
 24 01014 136510  
 25 01015 125004  
 26  
 27 01022 132520  
 28 01023 100510  
 29 01024 101004  
 30  
 31  
 32 01031 006264  
 33

/SET BIT 15  
 /LRR SHOULD RESET IT.  
 /C(ACS) SHOULD BE=0

SUR# 1,1  
 LRR 1,3  
 MOV 1,1,SZR  
 ERROR  
 LOOP

EX3H:

35 01032 006263  
 36 01033 006047  
 37 01034 102002  
 38 01035 120000  
 39 01036 120000  
 40 01037 102510  
 41 01040 120510  
 42 01041 101003  
 43 01042 006403  
 44 01043 110415  
 45 01044 130414  
 46  
 47  
 48 01051 006264  
 49

/ITERATE TEST ROUTINE

JSR #ENTLO  
 SETUP 40  
 JSR #ENTIN  
 40  
 ADC 0,0  
 ADC 1,1  
 ADC# 3,3  
 LRR 0,0  
 MOV 0,0,SNC  
 JMP #+3  
 SUB# 0,3,SNR  
 SUB# 1,3,SZR  
 ERROR  
 LOOP

EX3I:

41 01040 120510  
 42 01041 101003  
 43 01042 006403  
 44 01043 110415  
 45 01044 130414  
 46  
 47  
 48 01051 006264  
 49

/ITERATE TEST ROUTINE

JSR #ENTLO  
 SETUP 40  
 JSR #ENTIN  
 40  
 ADC 0,0  
 ADC 1,1  
 ADC# 3,3  
 LRR 0,0  
 MOV 0,0,SNC  
 JMP #+3  
 SUB# 0,3,SNR  
 SUB# 1,3,SZR  
 ERROR  
 LOOP

EX3J:

41 01040 120510  
 42 01041 101003  
 43 01042 006403  
 44 01043 110415  
 45 01044 130414  
 46  
 47  
 48 01051 006264  
 49

/ITERATE TEST ROUTINE

JSR #ENTLO  
 SETUP 40  
 JSR #ENTIN  
 40  
 ADC 0,0  
 ADC 1,1  
 ADC# 3,3  
 LRR 0,0  
 MOV 0,0,SNC  
 JMP #+3  
 SUB# 0,3,SNR  
 SUB# 1,3,SZR  
 ERROR  
 LOOP

EX3K:

41 01040 120510  
 42 01041 101003  
 43 01042 006403  
 44 01043 110415  
 45 01044 130414  
 46  
 47  
 48 01051 006264  
 49

/ITERATE TEST ROUTINE

JSR #ENTLO  
 SETUP 40  
 JSR #ENTIN  
 40  
 ADC 0,0  
 ADC 1,1  
 ADC# 3,3  
 LRR 0,0  
 MOV 0,0,SNC  
 JMP #+3  
 SUB# 0,3,SNR  
 SUB# 1,3,SZR  
 ERROR  
 LOOP

EX3L:

41 01040 120510  
 42 01041 101003  
 43 01042 006403  
 44 01043 110415  
 45 01044 130414  
 46  
 47  
 48 01051 006264  
 49





10041 ECL11

```

01
02
03
04 02116 000263
05 02117 000249
06
07 02120 000266
08 02121 105000
09 02122 115000
10 02123 102400
11 02124 129510
12 02125 134610
13 02126 150000
14 02127 110610
15 02130 170143
16 02131 000776
17 02132 142414
18
19
20 02137 000264
21
22
23 02140 000263
24 02141 000040
25
26 02142 000266
27 02143 115005
28 02144 000776
29 02145 120400
30 02146 104010
31 02147 172510
32 02150 175004
33 02151 000775
34 02152 110610
35 02153 130414
36
37
38 02150 000264
39

```

EX3P:

```

SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
MOV 0,1
MOV 0,3
SUB 0,0
LRB 1,0
ANC 1,3
ADC 2,2
ADI 1,2
MOVOL 3,3,SNC
JMP *-2
SUB# 2,0,5ZH
ERROR
LOOP
JSR #ENTLO

```

EX3U:

```

SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
MOV 0,3,SNR
JMP EX3U
SUB 1,1
ADI 1,1
LRB 3,2
MOV 3,3,SZR
JMP *-3
COB 0,3
SUB# 1,3,5ZR
ERROR
LOOP
JSR #ENTLO

```

TEST "LRR"  
INITIALIZE TEST.

```

FC(AC0)=RANDOM #

```

```

PA BIT CHANGE WILL RESULT
IN A SET BIT IN C(AC3)

```

SHIFT C(3) TO GET THE

```

FCOUNT
FC(AC0)=LRR RESULT
FC(AC2)=CORRECT
ITERATE TEST ROUTINE

```

TEST "LRR"  
INITIALIZE TEST.

FC(AC0)=RANDOM #

```

FCOUNT HOW MANY TIMES IT
TAKES "LRR" TO RESET
ALL BITS. THEN USE "LOB"
TO FIND HOW MANY BITS
WERE PRESENT
FC(AC0)=ORIGINAL NUMBER
FC(AC3)=COR RESULT (CORRECT)
FC(AC1)=LRR ITERATIONS.
ITERATE TEST ROUTINE

```

10042 ECL11

```

01
02
03 02161 000263
04 02162 000210
05
06 02163 000266
07 02164 110300
08 02165 114610
09 02166 115000
10 02167 120510
11 02168 120510
12 02170 150000
13 02171 110610
14 02172 101143
15 02173 000776
16 02174 100510
17 02175 132414
18
19
20 02202 000264
21
22
23 02203 000263
24 02204 000040
25
26 02205 150510
27 02206 120510
28 02207 170521
29 02210 000170
30 02211 101000
31 02212 112510
32 02214 000005
33 02215 170172
34 02216 000775
35 02217 030771
36 02220 172414
37
38
39 02225 000264
40

```

EX3K:

```

SETUP 10
JSR #ENTIN
10
RAND
JSR #ENTRA
MOV# 0,3
MAKE MORE THAN THE USUAL
NUMBER OF ZEROS.
FC(AC1)=0
FC(AC2)=1
SIMULATE "LRR" BY COUNTING
THE LEADING ZEROS
MOVOL 0,0,SNC
JMP *-2
LRB 3,1
SUB# 1,2,5ZR
ERROR
LOOP
JSR #ENTLO

```

EX3S:

```

SETUP 40
JSR #ENTIN
40
XOR 2,2
XOR 1,1
SUBL 3,3,SXP
LRB 3,0
MOV 0,2
MOV 0,0,SZR
JMP *-5
MOVZL 3,3,SZR
JMP *-5
LOA 3,1,-7
SUB# 3,2,5ZH
ERROR
LOOP
JSR #ENTLO

```

TEST "LRR"  
INITIALIZE TEST.

```

FC(AC0)=RANDOM #

```

```

FC(AC1)=LRR RESULT
FC(AC2)=CORRECT
ITERATE TEST ROUTINE

```

TEST "LRR"  
INITIALIZE TEST.

```

FCOUNT LEADING ZEROS
WITH A SINGLE BIT IN
EACH OF 16 BIT POSITIONS.
A SINGLE BIT GOES TO
FC(AC0). THE TOTAL COUNT
FC(AC2) SHOULD BE 120.
ITERATE UNTIL ALL
116 BITS HAVE BEEN TESTED.

```

FC(AC3)=CORRECT  
FC(AC2)=LRR SUM

ITERATE TEST ROUTINE

10043 ECL11

```

01 02
03 04 02225 006263
05 06 02227 006017
07 08 02230 106400
09 10 02231 126400
11 12 02232 152400
13 14 02233 176440
15 16 02234 162510
17 18 02235 156510
19 20 02236 024402
21 22 02237 101001
23 24 02240 006022
25 26 02241 124415
27 28 02242 136414
29 30 02247 006264
31 32 02247 006264
33 34 02250 006263
35 36 02251 006017
37 38 02252 006266
39 40 02253 105400
41 42 02254 152220
43 44 02255 142510
45 46 02256 106414
47 48 02263 006264
49 50 02264 006263
51 52 02265 006017
53 54 02266 006266
55 56 02267 115000
57 58 02270 126000
59 60 02271 136510
61 62 02272 136510
63 64 02273 127113
65 66 02274 116014
67 68 02301 006264

      ;
      *** HEX SHIFT LEFT ***

      EX3T:
      SETUP 10
      JSR #ENTLN
      L0
      ZERDAC
      SUB 0,0
      SUB 1,1
      SUB 2,2
      SUBO 3,3
      LRB 3,0
      LRB 2,3
      LDA 1,-*2
      MOV 0,0,SKP
      SUB# 1,0,SNR
      SUB# 1,3,SNR
      ERROR
      LOOP
      JSR #ENTLO

      EX3U:
      SETUP 10
      JSR #ENTLN
      L0
      RAND
      JSR #ENTRA
      INC 0,1
      ADC# 2,2
      LRB 2,0
      SUB# 0,1,SZR
      ERROR
      LOOP
      JSR #ENTLO

      EX3V:
      SETUP 10
      JSR #ENTLN
      L0
      RAND
      JSR #ENTRA
      MOV 0,3
      ADC 1,1
      LRB 1,3
      LRB 1,3
      ADD# 1,1,SNR
      ADC# 0,3,SZR
      ERROR
      LOOP
      JSR #ENTLO

      ;TEST "LRB"
      ;INITIALIZE TEST,
      ;CLEAR ALL AC'S
      ;AND C(CARRY)
      ;C(AC0) SHOULD COUNT TO 20
      ;C(AC3) SHOULD COUNT TO 20
      ;C(AC1)=CORRECT
      ;C(AC0)=FIRST LRB RESULT
      ;C(AC3)=SECOND LRB RESULT
      ;ITERATE TEST ROUTINE
      ;TEST "LRB"
      ;INITIALIZE TEST,
      ;C(AC0)=RANDOM #
      ;C(AC2)=1 TO C(AC1)
      ;C(AC2)=77777
      ;C(AC0)=LRB RESULT
      ;C(AC1)=CORRECT
      ;ITERATE TEST ROUTINE
      ;TEST "LRB"
      ;INITIALIZE TEST,
      ;C(AC0)=RANDOM #
      ;REMOVE BIT 0, DONT COUNT C(AC3)
      ;REMOVE BIT 1, COUNT C(AC3), *1
      ;TEST BIT 1 FOR 0
      ;C(AC3) SHOULD BE 1 GREATER THAN
      ;C(AC0).
      ;ITERATE TEST ROUTINE

      ;TEST "HXL"
      ;INITIALIZE TEST,
      ;SETUP 5
      ;JSR #ENTLN
      ;HSLACD 0
      ;ANC 0,0
      ;ANC 1,1
      ;ANC 2,2
      ;ANC 3,3
      ;ADC 0,0
      ;HXL 0+1,0
      ;ADD# 0+1,0,SNR
      ;MOV# 0+3,0+3,SNR
      ;ERROR
      ;HSLACD 1
      ;ANC 0,0
      ;ANC 1,1
      ;ANC 2,2
      ;ANC 3,3
      ;ADC 1,1
      ;HXL 1+1,1
      ;ADD# 1+1,1,SNR
      ;MOV# 1+3,1+3,SNR
      ;ERROR
      ;HSLACD 2
      ;ANC 0,0
      ;ANC 1,1
      ;ANC 2,2
      ;ANC 3,3
      ;ADC 2,2
      ;HXL 2+1,2
      ;ADD# 2+1,2,SNR
      ;MOV# 2+3,2+3,SNR
      ;ERROR
      ;HSLACD 3
      ;ANC 0,0
      ;ANC 1,1
      ;ANC 2,2
      ;ANC 3,3
      ;ADC 3,3
      ;HXL 3+1,3
      ;ADD# 3+1,3,SNR
      ;MOV# 3+3,3+3,SNR
      ;ERROR
      ;LOOP
      ;JSR #ENTLO
      ;ITERATE TEST ROUTINE

```

10044 ECL11

```

01 02
03 04 02302 006263
05 06 02303 006005
07 08 02304 100610
09 10 02305 120610
11 12 02306 150610
13 14 02307 170610
15 16 02311 101410
17 18 02312 130615
19 20 02313 175014
21 22 02320 100610
23 24 02321 120610
25 26 02322 150610
27 28 02323 170610
29 30 02324 126000
31 32 02325 125410
33 34 02326 157015
35 36 02327 101014
37 38 02334 100610
39 40 02335 120610
41 42 02336 150610
43 44 02337 170610
45 46 02340 152000
47 48 02341 151410
49 50 02342 163015
51 52 02343 125014
53 54 02350 100610
55 56 02351 120610
57 58 02352 150610
59 60 02353 170610
61 62 02354 170000
63 64 02355 175410
65 66 02356 107015
67 68 02357 151014
69 70 02364 006264

      ;TEST "HXL"
      ;CODES ONLY TO AC 0
      ;PRESET AC0-3 TO ZERO
      ;RESULT TO AC0
      ;HSLACD 0
      ;ANC 0,0
      ;ANC 1,1
      ;ANC 2,2
      ;ANC 3,3
      ;ADC 0,0
      ;HXL 0+1,0
      ;ADD# 0+1,0,SNR
      ;MOV# 0+3,0+3,SNR
      ;ERROR
      ;HSLACD 1
      ;ANC 0,0
      ;ANC 1,1
      ;ANC 2,2
      ;ANC 3,3
      ;ADC 1,1
      ;HXL 1+1,1
      ;ADD# 1+1,1,SNR
      ;MOV# 1+3,1+3,SNR
      ;ERROR
      ;HSLACD 2
      ;ANC 0,0
      ;ANC 1,1
      ;ANC 2,2
      ;ANC 3,3
      ;ADC 2,2
      ;HXL 2+1,2
      ;ADD# 2+1,2,SNR
      ;MOV# 2+3,2+3,SNR
      ;ERROR
      ;HSLACD 3
      ;ANC 0,0
      ;ANC 1,1
      ;ANC 2,2
      ;ANC 3,3
      ;ADC 3,3
      ;HXL 3+1,3
      ;ADD# 3+1,3,SNR
      ;MOV# 3+3,3+3,SNR
      ;ERROR
      ;LOOP
      ;JSR #ENTLO
      ;ITERATE TEST ROUTINE

```



```

10048 ECL11
01
02
03
04 02564 000263      ;TEST "HXL"
05 02565 000040      ;INITIALIZE TEST.
06 02566 000040
07 02567 000017
08 02570 000777     LDA 2,0,-1
09 02571 113720     ANDZ 0,2
10 02572 153100     ADDL 2,2
11 02573 153100     ADDL 2,2
12 02574 115000     MOV 0,3
13 02575 141410     HXL 3,0
14 02576 112414     SUR# 0,2,SZR
15                                     ERROR
16                                     LOOP
17 025A3 000264      ;ITERATE TEST ROUTINE
18

```

```

10047 ECL11
01
02
03
04 02567 000263      ;TEST "HXL"
05 02568 000040      ;INITIALIZE TEST.
06 02569 000040
07 02572 000017
08 02575 000777     LDA 2,0,-1
09 02576 113720     ANDZ 0,2
10 02577 153100     ADDL 2,2
11 02578 153100     ADDL 2,2
12 02579 115000     MOV 0,3
13 02580 141410     HXL 3,0
14 02581 153100     SUR# 1,2,SZR
15 02582 132415     SUB# 1,3,SZR
16 02583 132415     SUB# 1,3,SZR
17 02584 136414     FROR
18                                     LOOP
19
20 02530 000264      ;ITERATE TEST ROUTINE
21
22
23 02531 000263      ;TEST "HXL"
24 02532 000100      ;INITIALIZE TEST.
25 02533 000100
26 02534 000266      ;SAVE ORIGINAL RANDOM #
27 02535 105000     MOV 0,1
28 02536 111000     HXL 3,1
29 02537 145410     HXL 2,0
30 02538 121410     HXL 1,0
31 02539 101410     SUR# 0,1,SZR
32 02540 106414     ERROR
33                                     LOOP
34
35 02546 000264      ;ITERATE TEST ROUTINE
36
37
38 02547 000263      ;TEST "HXL"
39 02548 000010      ;INITIALIZE TEST.
40 02549 000010
41 02551 000266      ;SAVE ORIGINAL RANDOM #
42 02552 105000     MOV 0,1
43 02553 111000     HXL 3,1
44 02554 145410     HXL 2,0
45 02555 121410     HXL 1,0
46 02556 101410     SUR# 0,1,SZR
47 02557 106414     ERROR
48                                     LOOP
49
50 02563 000264      ;ITERATE TEST ROUTINE
51

```

```

10046 ECL11
01
02
03
04 02564 000263      ;TEST "HXL"
05 02565 000040      ;INITIALIZE TEST.
06 02566 000040
07 02567 000017
08 02570 000777     LDA 2,0,-1
09 02571 113720     ANDZ 0,2
10 02572 153100     ADDL 2,2
11 02573 153100     ADDL 2,2
12 02574 115000     MOV 0,3
13 02575 141410     HXL 3,0
14 02576 112414     SUR# 0,2,SZR
15                                     ERROR
16                                     LOOP
17 025A3 000264      ;ITERATE TEST ROUTINE
18

```

```

10045 ECL11
01
02
03
04 02564 000263      ;TEST "HXL"
05 02565 000040      ;INITIALIZE TEST.
06 02566 000040
07 02567 000017
08 02570 000777     LDA 2,0,-1
09 02571 113720     ANDZ 0,2
10 02572 153100     ADDL 2,2
11 02573 153100     ADDL 2,2
12 02574 115000     MOV 0,3
13 02575 141410     HXL 3,0
14 02576 112414     SUR# 0,2,SZR
15                                     ERROR
16                                     LOOP
17 025A3 000264      ;ITERATE TEST ROUTINE
18

```



10049 ECL11

```

01
02
03 02607 106510
04 02607 124510
05
06 02604 006263
07 02604 006263
08 02605 006985
09 02606 106510
10 02607 124510
11 02610 150510
12 02611 174510
13 02612 106260
14 02613 101510
15 02614 133015
16 02615 175014
17 02615 175014
18
19 02622 106510
20 02623 124510
21 02624 150510
22 02625 174510
23 02626 125000
24 02627 125510
25 02630 157015
26 02631 101014
27 02631 101014
28
29 02636 106510
30 02637 124510
31 02640 150510
32 02641 174510
33 02642 152000
34 02643 151510
35 02644 163015
36 02645 125014
37 02645 125014
38
39 02652 106510
40 02653 124510
41 02654 150510
42 02654 174510
43 02655 174510
44 02656 176000
45 02657 175510
46 02660 107015
47 02661 151014
48
49 02666 006264

```

\*\*\* HEX SHIFT RIGHT \*\*\*

7

EX4G:

```

SETUP 5
JSK #ENTIN 5
MSRACD 0
XOR 0,0
XOR 1,1
XOR 2,2
XOR 3,3
XOR 0,0
XOR 0+1,0
ADD 0+183,0+283,SNR
MOV 0+383,0+383,SZR
ERROR
MSRACD 1
XOR 0,0
XOR 1,1
XOR 2,2
XOR 3,3
XOR 1,1
ADD 1+183,1+283,SNR
MOV 1+383,1+383,SZR
ERROR
MSRACD 2
XOR 0,0
XOR 1,1
XOR 2,2
XOR 3,3
XOR 2,2
XOR 2+1,2
ADD 2+183,2+283,SNR
MOV 2+383,2+383,SZR
ERROR
MSRACD 3
XOR 0,0
XOR 1,1
XOR 2,2
XOR 3,3
XOR 3,3
XOR 3+1,3
ADD 3+183,3+283,SNR
MOV 3+383,3+383,SZR
ERROR
LOOP

```

EX4H:

```

JSK #ENTIN
I# 4
AUC 1,1
M# 4,1
AUC 1,1
M# 4,1
ADC 1,1
M# 4,1
ADC 1,1
M# 4,1
M# 0,1,SNR
ADD 2,3,SZR
ERROR
LOOP
JSK #ENTLO

```

ITERATE TEST ROUTINE

```

JTEST "HXR"
INITIALIZE TEST.
FSET AC TO 177777
FTHEN SHIFT IN ALL ZEROS
FSHIFT TO ZERO RESULT
FSHIFT TO ZERO RESULT
FSHIFT TO ZERO RESULT
FSHIFT TO ZERO RESULT
FACB-3 SHOULD ALL
BE ZEROS. HSR FAILED
ITERATE TEST ROUTINE

```

49

10049 ECL11

01

```

02
03 02710 066263
04 02710 066263
05
06 02712 066266
07 02713 034492
08 02714 117701
09 02715 177402
10 02715 177402
11 02717 118000
12 02720 125510
13 02721 131510
14 02722 166415
15 02723 172414
16 02723 172414
17
18 02730 066264

```

EX41:

```

SETUP 40
JSK #ENTIN 40
RAND
JSK #ENTRA
LDA 3,+2
AND 0,3,SNP
177402
MOV 0,1
MOV 0,2
M# 2,1
M# 2,2
SUB 3,1,SNR
SUB 3,2,SZR
ERROR
LOOP
JSK #ENTLO

```

ITERATE TEST ROUTINE

```

JTEST "HXR"
INITIALIZE TEST.
FACB-3 SHOULD ALL
BE ZEROS. HSR FAILED
ITERATE TEST ROUTINE

```

49

```

10051 ECL11
01
02
03
04 02731 006263
05 02732 000048
06
07 02733 006266
08 02734 024402
09 02735 107401
10 02736 000377
11 02737 122400
12 02740 111300
13 02741 130300
14 02742 135510
15 02743 121510
16 02744 112415
17 02745 136414
18
19
20
21
22
23 02753 006263
24 02754 000042
25
26 02755 006266
27 02756 105220
28 02757 125220
29 02760 125220
30 02761 125220
31 02762 111000
32 02763 115000
33 02764 111510
34 02765 115510
35 02766 132415
36 02767 136414
37
38
39 02774 006264
40

```

```

EX4J:  SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
LDA 1,+2
AND 0,1,SKP
377
SUB 1,0
MOVS 0,2
MOVS 1,3
MFR 2,3
MFR 2,0
SUB# 0,2,SNR
SUB# 1,3,SNR
ERROR
JSR #ENTLO
LOOP

```

```

!TEST "HXR"
!INITIALIZE TEST.
FC(AC0)=RANDOM #
!SPLIT RANDOM INTO 2
!PARTS. THEN HAVE PARTS IN
!OPPOSITE HAVES OF ACS
!SO (HXR) CAN ALIGN THEM
!SMOULD SHIFT C(AC3) TO=C(AC1)
!SHOULD SHIFT C(AC0) TO=C(AC2)
FC(AC1-2)=CORRECT
FC(AC0-3)=HNR RESULT
!ITERATE TEST ROUTINE
!TEST "HXR"
!INITIALIZE TEST.
FC(AC0)=RANDOM #
!SIMULATE A HEX
!SHIFT IN C(AC1)
FC(AC2-3)=RANDCH
!SHIFT C(AC2) RIGHT 4 TIMES
!SHIFT C(AC3) RIGHT 4 TIMES
FC(AC1)=CORRECT
FC(AC0)=ORIGINAL
FC(AC2-3)=HNR RESULT
!ITERATE TEST ROUTINE

```

```

EX4L:  SETUP 100
JSR #ENTIN
100
RAND
JSR #ENTRA
LOA 3,+2
AND 0,3,SKP
177760
MOV 0,1
MOV 0,1
*UG 8.
MFR 1,0
MFR 1,1
MFR 1,0
MFR 1,1
MFR 1,0
MFR 1,1
MFR 1,0
MFR 1,1
MFR 1,0
MFR 1,1
MFR 1,0
MFR 1,1
MFR 1,0
MFR 1,1
MFR 1,0
MFR 1,1
MFR 1,0
MFR 1,1
MFR 1,0
MFR 1,1
MFR 1,0
MFR 1,1
MFR 1,0
MFR 1,1
MFR 1,0
MFR 1,1
MFR 1,0
MFR 1,1
MFR 1,0
MFR 1,1
SUB# 3,0,SNR
SUB# 3,1,SNR
ERROR
JSR #ENTLO
LOOP

```

```

10052 ECL11
01
02
03
04 02775 006263
05 02776 000102
06
07 02777 006266
08 03000 034402
09 03001 117401
10 03002 177760
11 03003 105000
12 03004 105000
13
14 03005 101510
15 03006 105510
16 03007 101410
17 03010 105410
18 03011 101510
19 03012 105510
20 03015 101410
21 03014 105410
22 03015 101510
23 03016 105510
24 03017 101410
25 03020 105410
26 03021 101510
27 03022 105510
28 03023 101410
29 03024 105410
30 03025 101510
31 03026 105510
32 03027 101410
33 03030 105410
34 03031 101510
35 03032 105510
36 03033 101410
37 03034 105410
38 03035 101510
39 03036 105510
40 03037 101410
41 03040 105410
42 03041 101510
43 03042 105510
44 03043 101410
45 03044 105410
46 03045 102415
47 03048 106414
48
49
50 03053 006264
51

```

```

!TEST "HXR"
!INITIALIZE TEST.
FC(AC0)=RANDOM #
!REMOVE 4 LOW ORDER BITS
!FROM C(RANDOM) AND PUT IN
FC(AC3).
FC(AC0), RIGHT 4 PLACES
FC(AC1), RIGHT 4 PLACES
FC(AC0), LEFT 4 PLACES
FC(AC1), LEFT 4 PLACES
FC(AC0), RIGHT 4 PLACES
FC(AC1), RIGHT 4 PLACES
FC(AC0), LEFT 4 PLACES
FC(AC1), LEFT 4 PLACES
FC(AC0), RIGHT 4 PLACES
FC(AC1), RIGHT 4 PLACES
FC(AC0), LEFT 4 PLACES
FC(AC1), LEFT 4 PLACES
FC(AC0), RIGHT 4 PLACES
FC(AC1), RIGHT 4 PLACES
FC(AC0), LEFT 4 PLACES
FC(AC1), LEFT 4 PLACES
FC(AC0), RIGHT 4 PLACES
FC(AC1), RIGHT 4 PLACES
FC(AC2)=CORRECT
FC(AC3)=ORIGINAL
FC(AC0-1)=HNR/SL RESULT
!ITERATE TEST ROUTINE

```

```

!0654 ECL11
01
02 03111 006203
03 03112 006100
04
05 03113 006266
06 03114 115021
07 03115 170000
08 03116 024777
09 03117 107402
10 03120 155510
11 03121 155410
12 03122 156415
13 03123 131002
14
15
16 03130 006264
17
18
19 03131 006263
20 03132 006100
21
22 03133 006266
23 03134 111000
24 03135 174510
25 00 4
26 03136 101140
27 03137 175100
28 03140 101100
29 03141 175100
30 03142 101140
31 03143 175100
32 03144 101140
33 03145 175100
34 03146 151510
35 03147 156415
36 03150 101002
37
38
39 03155 006264
40
41
42 03156 006263
43 03157 006100
44
45 03160 006266
46 03161 034402
47 03162 117401
48 03163 006217
49 03164 105042
50 03165 145410
51 03166 145510
52 03167 136415
53 03170 101003
54
55
56 03175 006264
57

```

```

!TEST "HXR"
!INITIALIZE TEST.

SUBU 0,0
HXR 2,0
MOV 0,0,SNR
MOV 0,0,SZC
ERROR
LOOP
JSR #ENTLO

!ITERATE TEST ROUTINE

!TEST "HXR"
!INITIALIZE TEST.

!SHIFTING ALL ZEROS
!SHOULD NOT CHANGE C(AC0)
FOR C(CARRY)

!ITERATE TEST ROUTINE

!TEST "HXR"
!INITIALIZE TEST.

!C(AC0)=RANDOM #
!ITERATE TEST ROUTINE

!TEST "HXR"
!INITIALIZE TEST.

!C(AC0)=RANDOM #
!4 BITS FROM C(AC0) TO C(AC3)
!CLEANS C(CARRY)
!CLEANS C(CARRY)
!CLEANS C(CARRY)
!SHIFT RIGHT 12 PLACES
!C(AC3)=CORRECT
!C(AC2)=HSR RESULT
!C(CARRY) SHOULD BE (0)
!ITERATE TEST ROUTINE

!TEST "HXR"
!INITIALIZE TEST.

!C(AC0)=RANDOM #
!C(CARRY)=(1)
!LEFT
!RIGHT
!C(AC3)=CORRECT
!C(AC0)=ORIGINAL
!C(AC1)=HSL/HSR RESULT
!ITERATE TEST ROUTINE

```

```

1W055 ECL11
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

*** DOUBLE HEX LEFT/RIGHT SHIFT ***

EX4R:
07 03176 006263          /TEST "DHXL"
08 03177 006264          INITIALIZE TEST.
09 03200 102000          /SET AC 0 TO 17777
10 03201 102000          /OTHER AC'S SET TO 0
11 03202 102000          /DHXL 4,0
12 03203 102000          /ADD# 0,1,SNR
13 03204 102000          /ADD# 2,3,SRZ
14 03205 102000          /ERR#
15 03206 102000          /LOOP
16 03207 102000          /ITERATE TEST ROUTINE

EX4S:
07 03214 006263          /TEST "DHXL"
08 03215 006264          INITIALIZE TEST.
09 03216 102000          /SET AC 1 TO 17777
10 03217 102000          /OTHER AC'S SET TO 0
11 03218 102000          /DHXL 4,1
12 03219 102000          /ADD# 0,1,SNR
13 03220 102000          /ADD# 2,3,SRZ
14 03221 102000          /ERR#
15 03222 102000          /LOOP
16 03223 102000          /ITERATE TEST ROUTINE

EX4T:
07 03231 006264          /TEST "DHXL"
08 03232 006265          INITIALIZE TEST.
09 03233 006266          /SET AC 2 TO 17777
10 03234 102000          /OTHER AC'S SET TO 0
11 03235 102000          /DHXL 4,2
12 03236 102000          /ADD# 0,1,SNR
13 03237 102000          /ADD# 2,3,SRZ
14 03238 102000          /ERR#
15 03239 102000          /LOOP
16 03240 102000          /ITERATE TEST ROUTINE

EX4U:
04 03250 006263          /TEST "DHXL"
05 03251 006264          INITIALIZE TEST.
06 03252 102000          /SET AC 3 TO 17777
07 03253 102000          /OTHER AC'S SET TO 0
08 03254 102000          /DHXL 4,3
09 03255 102000          /ADD# 0,1,SNR
10 03256 102000          /ADD# 2,3,SRZ
11 03257 102000          /ERR#
12 03258 102000          /LOOP
13 03259 102000          /ITERATE TEST ROUTINE

EX4V:
04 03265 006264          /TEST "DHXL"
05 03266 006265          INITIALIZE TEST.
06 03267 102000          /SET AC 3 TO 17777
07 03268 102000          /OTHER AC'S SET TO 0
08 03269 102000          /DHXL 4,3
09 03270 102000          /ADD# 0,1,SNR
10 03271 102000          /ADD# 2,3,SRZ
11 03272 102000          /ERR#
12 03273 102000          /LOOP
13 03274 102000          /ITERATE TEST ROUTINE

```

```

1W056 ECL11
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16

EX4U:
04 03250 006263          /TEST "DHXL"
05 03251 006264          INITIALIZE TEST.
06 03252 102000          /SET AC 3 TO 17777
07 03253 102000          /OTHER AC'S SET TO 0
08 03254 102000          /DHXL 4,3
09 03255 102000          /ADD# 0,1,SNR
10 03256 102000          /ADD# 2,3,SRZ
11 03257 102000          /ERR#
12 03258 102000          /LOOP
13 03259 102000          /ITERATE TEST ROUTINE

EX4V:
04 03265 006264          /TEST "DHXL"
05 03266 006265          INITIALIZE TEST.
06 03267 102000          /SET AC 3 TO 17777
07 03268 102000          /OTHER AC'S SET TO 0
08 03269 102000          /DHXL 4,3
09 03270 102000          /ADD# 0,1,SNR
10 03271 102000          /ADD# 2,3,SRZ
11 03272 102000          /ERR#
12 03273 102000          /LOOP
13 03274 102000          /ITERATE TEST ROUTINE

```

10057 ECL11

01

02

03

04 03266 000263

05 03267 000102

06

07 03270 000266

08 03271 111302

09 03272 105300

10 03273 161610

11 03274 125005

12 03275 112414

13

14

15 03302 000264

16

17

18

19 03303 000263

20 03304 000040

21

22 03305 000266

23 03306 105000

24 03307 111000

25 03310 115000

26 03311 161710

27 03312 161610

28 03313 125005

29 03314 112414

30

31

32 03321 000264

33

34

35

36 03322 000263

37 03323 000040

38

39 03324 000266

40 03325 105000

41 03326 111000

42 03327 115000

43 03330 165710

44 03331 165610

45 03332 151005

46 03333 130414

47

48 03340 000264

TEST "DHXL"  
INITIALIZE TEST.

IC(AC0)=RANDOM #

SHIFT C(AC1) INTO C(AC0)  
C(AC1) SHOULD CONTAIN ZEROS  
C(AC2)=CORRECT  
C(AC0)=DHXL RESULT

ITERATE TEST ROUTINE

TEST "DHXL/DHSL"  
INITIALIZE TEST.

IC(AC0)=RANDOM #  
IC(AC 0) SHOULD NOT CHANGE

DOUBLE SHIFT 16 PLACES  
RIGHT THEN LEFT

ITERATE TEST ROUTINE

TEST "DHXL/DHSL"  
INITIALIZE TEST.

IC(AC0)=RANDOM #  
IC(AC 1) SHOULD NOT CHANGE

DOUBLE SHIFT 16 PLACES  
RIGHT THEN LEFT

ITERATE TEST ROUTINE

10058 ECL11

01

02

03

04 03341 000263

05 03342 000040

06

07 03343 000266

08 03344 105000

09 03345 111000

10 03346 115000

11 03347 171710

12 03350 171610

13 03351 175005

14 03352 142414

15

16

17 03357 000264

18

19

20

21 03360 000263

22 03361 000040

23

24 03362 000266

25 03363 105000

26 03364 111000

27 03365 115000

28 03366 175710

29 03367 175610

30 03370 101005

31 03371 100414

32

33

34 03376 000264

35

36

37 03377 000263

38 03380 000102

39

40 03401 000266

41 03402 105000

42 03403 111320

43 03404 121610

44 03405 112415

45 03406 101002

46

47

48 03413 000264

EX50:

L=SHIFT 2

SETUP 40

JSR #ENTIN

40

RAND

JSR #ENTRA

MOV 0,1

MOV 0,2

MOV 0,3

DHXL 4,2

DHXL 4,2

MOV 2+183,2+183,SNR

SUB# 2,2+283,SZR

EMROR

LOOP

JSR #ENTLO

ITERATE TEST ROUTINE

L=SHIFT 3

SETUP 40

JSR #ENTIN

40

RAND

JSR #ENTRA

MOV 0,1

MOV 0,2

MOV 0,3

DHXL 4,3

DHXL 4,3

MOV 3+183,3+183,SNR

SUB# 3,3+283,SZR

EMROR

LOOP

JSR #ENTLO

ITERATE TEST ROUTINE

SETUP 100

JSR #ENTIN

100

RAND

JSR #ENTRA

MOV 0,1

MOV 0,2

DHXL 2,0

SUB# 0,2,SNR

MOV 0,0,SZR

EMROR

LOOP

JSR #ENTLO

ITERATE TEST ROUTINE

EX50:

TEST "DHXL/DHSL"  
INITIALIZE TEST.

IC(AC0)=RANDOM #  
IC(AC 2) SHOULD NOT CHANGE

DOUBLE SHIFT 16 PLACES  
RIGHT THEN LEFT

ITERATE TEST ROUTINE

TEST "DHXL/DHSL"  
INITIALIZE TEST.

IC(AC0)=RANDOM #  
IC(AC 3) SHOULD NOT CHANGE

DOUBLE SHIFT 16 PLACES  
RIGHT THEN LEFT

ITERATE TEST ROUTINE

TEST "DHXL"  
INITIALIZE TEST.

IC(AC0)=RANDOM #  
IC(AC2)=R,A  
IC(AC0) SHOULD=R,A  
IC(CARRY) SHOULD REMAIN ZERO

IC(AC0)=CORRECT  
IC(AC0)=DHXL RESULT

ITERATE TEST ROUTINE

```

10059 ECL11
01
02
03
04 03414 006203
05 03415 000042
06
07 03416 006206
08 03417 101000
09 03420 105000
10 03421 113340
11 03422 121610
12 03423 110415
13 03424 101003
14
15
16 03431 006204
17
18
19
20 03432 006203
21 03433 000042
22
23 03434 006206
24 03435 105000
25 03437 121340
26 03437 121340
27 03440 125610
28 03441 122415
29 03442 101003
30
31
32 03447 006204
33

EX56:
DL8 0
SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
MOV 0,0
MOV 0,0+183
MOVOS 0,0+383
DXL 2,0
SUB# 0,0+383,SNR
MOV 0,0,SNR
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE

INITIALIZE TEST,
ITERATE TEST ROUTINE

DMSL FAILED
DMSL FAILED

FC(AC0)=RANDOM #
TEST THE HIGH ORDER PART
OF A DOUBLE SHIFT LEFT
18 PLACES TO AC 0
FC(CARRY) SHOULD NOT CHANGE
DMSL FAILED
ITERATE TEST ROUTINE

DMSL FAILED
ITERATE TEST ROUTINE

EX57:
DL8 1
SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
MOV 0,1
MOV 1,1+183
MOVOS 1,1+383
DXL 2,1
SUB# 1,1+383,SNR
MOV 0,0,SNR
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE

INITIALIZE TEST,
ITERATE TEST ROUTINE

DMSL FAILED
DMSL FAILED

FC(AC0)=RANDOM #
TEST THE HIGH ORDER PART
OF A DOUBLE SHIFT LEFT
18 PLACES TO AC 1
FC(CARRY) SHOULD NOT CHANGE
DMSL FAILED
ITERATE TEST ROUTINE

DMSL FAILED
ITERATE TEST ROUTINE

EX51:
DL8 2
SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
MOV 0,2
MOV 2,2+183
MOVOS 2,2+383
DXL 2,2
SUB# 2,2+383,SNR
MOV 0,0,SNR
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE

INITIALIZE TEST,
ITERATE TEST ROUTINE

DMSL FAILED
DMSL FAILED

FC(AC0)=RANDOM #
TEST THE HIGH ORDER PART
OF A DOUBLE SHIFT LEFT
18 PLACES TO AC 2
FC(CARRY) SHOULD NOT CHANGE
DMSL FAILED
ITERATE TEST ROUTINE

DMSL FAILED
ITERATE TEST ROUTINE

EX5J:
DL8 3
SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
MOV 0,3
MOV 3,3+183
MOVOS 3,3+383
DXL 2,3
SUB# 3,3+383,SNR
MOV 0,0,SNR
ERROR
LOOP
JSR #ENTLG
ITERATE TEST ROUTINE

INITIALIZE TEST,
ITERATE TEST ROUTINE

DMSL FAILED
DMSL FAILED

FC(AC0)=RANDOM #
TEST THE HIGH ORDER PART
OF A DOUBLE SHIFT LEFT
18 PLACES TO AC 3
FC(CARRY) SHOULD NOT CHANGE
DMSL FAILED
ITERATE TEST ROUTINE

DMSL FAILED
ITERATE TEST ROUTINE

```

14061 ELL11

```

01 DRB 0
02 SETUP 40
03 JSR #ENTIN
04 40
05 03504 006263
06 03505 006042
07
08 03506 006266
09 03507 101000
10 03514 101000
11 03511 101000
12 03512 121710
13 03513 106415
14 03514 101002
15
16 JSR #ENTLU
17
18
19
20
21 03522 006263
22 03523 006042
23
24 03524 006266
25 03525 101000
26 03526 131000
27 03527 121320
28 03530 125710
29 03531 142415
30 03532 101002
31
32
33 03537 006264
34
35
36
37
38
39 03541 006042
40 03542 006266
41 03543 101000
42 03544 105000
43 03545 145320
44 03546 131710
45 03547 166415
46 03550 101002
47
48
49 03555 006264

```

```

EXSK:
DRB 0
SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
JC(AC0)=RANDOM #
I TEST THE LOW CRUER PART
FOR THE DOUBLE SHIFT
MOV 0,0
MOV 0,0+183
MOVZ 0,0+383
DHR 2,0
SUB 0,0+183,0+383,SNR
MOV 0,0,5ZC
ERRR
LOOP
JSR #ENTLU
ITERATE TEST ROUTINE

EXSL:
DRB 1
SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
JC(AC0)=RANDOM #
I TEST THE LOW ORDER PART
FOR THE DOUBLE SHIFT
MOV 0,1
MOVZ 1,1+383
DHR 2,1
SUB 0,0+183,1+383,SNR
MOV 0,0,5ZC
ERRR
LOOP
JSR #ENTLU
ITERATE TEST ROUTINE

```

14062 ELL11

```

01 DRB 3
02 SETUP 40
03 JSR #ENTIN
04 40
05 03555 006263
06 03557 006042
07
08 03558 006266
09 03561 115000
10 03562 161000
11 03563 171320
12 03564 135710
13 03565 142415
14 03566 101002
15
16 JSR #ENTLU
17
18
19
20 03574 006263
21 03575 006042
22
23 03576 006266
24 03577 101000
25 03580 006042
26 03581 115701
27 03582 177400
28 03583 115300
29 03584 105000
30 03585 121710
31 03586 112415
32 03587 135414
33
34
35 03614 006264
36
37
38
39 03615 006263
40 03616 006042
41
42 03617 006266
43 03620 105000
44 03621 006042
45 03622 137701
46 03623 177400
47 03624 121300
48 03625 131000
49 03626 125710
50 03627 136415
51 03630 142414
52
53
54 03635 006264

```

```

EXSN:
DRB 3
SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
JC(AC0)=RANDOM #
I TEST THE LOW ORDER PART
FOR THE DOUBLE SHIFT
MOV 0,3
MOVZ 3,3+383
DHR 2,3
SUB 0,0+183,3+383,SNR
MOV 0,0,5ZC
ERRR
LOOP
JSR #ENTLU
ITERATE TEST ROUTINE

EXSO:
DRB 0
SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
JC(AC0)=RANDOM #
I TEST THE LOW ORDER PART
FOR THE DOUBLE SHIFT
MOV 0,0
MOVZ 0,0+283,0+283,SNR
DHR 2,0
SUB 0,0+183,0+283,SNR
MOV 0,0,5ZC
ERRR
LOOP
JSR #ENTLU
ITERATE TEST ROUTINE

```

14063 ELL11

```

01 DRB 2
02 SETUP 40
03 JSR #ENTIN
04 40
05 03641 006263
06 03642 006042
07
08 03643 006266
09 03646 105000
10 03647 145320
11 03648 131710
12 03649 166415
13 03652 101002
14
15
16 JSR #ENTLU
17
18
19
20
21 03658 006263
22 03659 006042
23
24 03660 006266
25 03663 105000
26 03664 145320
27 03665 131710
28 03666 166415
29 03669 101002
30
31
32
33 03694 006264
34
35
36
37
38
39 03695 006263
40 03696 006042
41
42 03697 006266
43 03700 105000
44 03701 145320
45 03702 131710
46 03703 166415
47 03706 101002
48
49 03711 006264

```

```

EXSM:
DRB 2
SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
JC(AC0)=RANDOM #
I TEST THE LOW ORDER PART
FOR THE DOUBLE SHIFT
MOV 0,2+183
MOVZ 2,2+383
DHR 2,2
SUB 0,0+183,2+383,SNR
MOV 0,0,5ZC
ERRR
LOOP
JSR #ENTLU
ITERATE TEST ROUTINE

EXSP:
DRB 1
SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
JC(AC0)=RANDOM #
I TEST THE LOW ORDER PART
FOR THE DOUBLE SHIFT
MOV 0,1
MOVZ 1,1+383
DHR 2,1
SUB 0,0+183,1+383,SNR
MOV 0,0,5ZC
ERRR
LOOP
JSR #ENTLU
ITERATE TEST ROUTINE

```

```

00063 ECL11
01
02
03
04 03636 006263
05 03637 000020
06
07 03640 006266
08 03641 111000
09 03642 024402
10 03643 143701
11 03644 177400
12 03645 145300
13 03646 155000
14 03647 131710
15 03650 142415
16 03651 166414
17
18
19 03656 006264
20
21
22
23 03657 006263
24 03660 000020
25
26 03661 006266
27 03662 115000
28 03663 024402
29 03664 167701
30 03665 177400
31 03666 171300
32 03667 161000
33 03670 135710
34 03671 166415
35 03672 112414
36
37
38 03677 006264
39
40
41
42 03700 006263
43 03701 000020
44
45 03702 006266
46 03703 101000
47 03704 034402
48 03705 113701
49 03706 006377
50 03707 115300
51 03710 100000
52 03711 121610
53 03712 116415
54 03713 132414
55
56
57 03720 006264
00064 ECL11
01
02
03
04 03721 006263
05 03722 000020
06
07 03723 006266
08 03724 105000
09 03725 034402
10 03726 137701
11 03727 006377
12 03730 121300
13 03731 131000
14 03732 125610
15 03733 122415
16 03734 166414
17
18
19 03741 006264
20
21
22
23 03742 006263
24 03743 000020
25
26 03744 006266
27 03745 111000
28 03746 024402
29 03747 143701
30 03750 006377
31 03751 145300
32 03752 155000
33 03753 131610
34 03754 140415
35 03755 162414
36
37
38 03762 006264
39
40
41
42 03763 006263
43 03764 000020
44
45 03765 006266
46 03766 115000
47 03767 024402
48 03770 167701
49 03771 006377
50 03772 171300
51 03773 161000
52 03774 135610
53 03775 172415
54 03776 166414
55
56
57 04003 006264
INITIALIZE TEST.
JSHSL TEST ROUTINE
ITERATE TEST ROUTINE
DSLR 1
SETUP 20
JSR #ENTLN
RAND
JSR #ENTRA
MOV #0,1
LDA #2+283,+2
ANDS #1+283,SKP
J77
MOVS #1,+383
MOV #1,+183
DHAL 2,1
SUB# #1+383,SNR
SUB# #1+183,+1+183,0
ERROR
JSHSL FAILED
LOOP
JSR #ENTLU
ITERATE TEST ROUTINE
DSLR 2
SETUP 20
JSR #ENTLN
RAND
JSR #ENTRA
MOV #0,2
LDA #2+283,+2
ANDS #2+283,SKP
J77
MOVS #2,+383
MOV #2,+183
DHAL 2,2
SUB# #2+383,SNR
SUB# #2+183,+2+283,SKP
ERROR
JSHSL FAILED
LOOP
JSR #ENTLU
ITERATE TEST ROUTINE
DSLR 3
SETUP 20
JSR #ENTLN
RAND
JSR #ENTRA
MOV #0,3
LDA #3+283,+2
ANDS #3+283,SKP
J77
MOVS #3,+383
MOV #3,+183
DHAL 2,3
SUB# #3+383,SNR
SUB# #3+183,+3+283,SKP
ERROR
JSHSL FAILED
LOOP
JSR #ENTLU
ITERATE TEST ROUTINE
78 PLACE DOUBLE SHIFT
INITIALIZE TEST.
JSHSL TEST ROUTINE
ITERATE TEST ROUTINE
DSLR 3
SETUP 40
JSR #ENTLN
RAND
JSR #ENTRA
MOV #0,2
LDA #2+283,+2
ANDS #2+283,SKP
J77
MOVS #2,+383
MOV #2,+183
DHAL 2,2
SUB# #2+283,SNR
SUB# #2+183,+2+283,SKP
ERROR
JSHSL FAILED
LOOP
JSR #ENTLU
ITERATE TEST ROUTINE
78 PLACE DOUBLE SHIFT
INITIALIZE TEST.
JSHSL TEST ROUTINE
ITERATE TEST ROUTINE
DSLR 0
SETUP 20
JSR #ENTLN
RAND
JSR #ENTRA
MOV #0,0
LDA #0+283,+2
ANDS #0+283,SKP
J77
MOVS #0,+383
MOV #0,+183
DHAL 2,0
SUB# #0+383,SNR
SUB# #0+183,+0+283,SKP
ERROR
JSHSL FAILED
LOOP
JSR #ENTLU
ITERATE TEST ROUTINE

```



10065 ECL11

01 EX5M1

02 DS4 0

03 SETUP 10

04 JSR #ENTIN

05 10

06 ADZ 0,0,SKP

07 7777

08 LDA 0,183,-1

09 DMXL 1,0

10 SUB# 0,0,183,SNK

11 MOV 0,0,SZC

12 ERROR

13 LOOP

14 JSR #ENTLO

15 14 04020 006264

16 16

17 EX5X1

18 DS4 1

19 SETUP 10

20 JSR #ENTIN

21 10

22 ADZ 1,1,SKP

23 7777

24 LDA 1,183,-1

25 DMXL 1,1

26 SUB# 1,1,183,SNR

27 MOV 0,0,SZC

28 ERROR

29 LOOP

30 JSR #ENTLO

31 28 04035 006264

32 30

33 EX5Y1

34 DS4 2

35 SETUP 10

36 JSR #ENTIN

37 10

38 ADZ 2,2,SKP

39 7777

40 LDA 2,183,-1

41 DMXL 1,2

42 SUB# 2,2,183,SNR

43 MOV 0,0,SZC

44 ERROR

45 LOOP

46 JSR #ENTLO

47 42 04052 006264

48 46

10066 ECL11

01 EX5Z1

02 DS4 0

03 SETUP 10

04 JSR #ENTIN

05 10

06 ADZ 3,3,SKP

07 7777

08 LDA 3,183,-1

09 DMXL 1,3

10 SUB# 3,3,183,SNH

11 MOV 0,0,SZC

12 ERROR

13 LOOP

14 JSR #ENTLO

15 14 04067 006264

16 16

17 EX6A1

18 DS4 0

19 SETUP 10

20 JSR #ENTIN

21 10

22 ADZ 0,183,0,183,SKP

23 177760

24 LDA 0,-1

25 DMXL 1,0

26 SUB# 0,0,183,SNK

27 MOV 0,0,SNC

28 ERROR

29 LOOP

30 JSR #ENTLO

31 28 04104 006264

32 30

33 EX6B1

34 DS4 1

35 SETUP 10

36 JSR #ENTIN

37 10

38 ADZ 1,183,1,183,SKP

39 177760

40 LDA 1,-1

41 DMXL 1,1

42 SUB# 1,1,183,SNF

43 MOV 0,0,SNC

44 ERROR

45 LOOP

46 JSR #ENTLO

47 42 04121 006264

48 46

INITIALIZE TEST.

7777 SHIFTED LEFT WILL  
BEQUAL 177740, SAME AS  
HIGH ORDER PART

FC(CARRY) SHOULD NOT CHANGE  
FONSL FAILED

ITERATE TEST ROUTINE

INITIALIZE TEST.

777760 SHIFTED RIGHT=7777  
FALL ONES SHIFTED RIGHT  
FALSO EQUALS 7777

FC(CARRY) SHOULD NOT CHANGE  
FONSL FAILED

ITERATE TEST ROUTINE

INITIALIZE TEST.

777760 SHIFTED RIGHT=7777  
FALL ONES SHIFTED RIGHT  
FALSO EQUALS 7777

FC(CARRY) SHOULD NOT CHANGE  
FONSL FAILED

ITERATE TEST ROUTINE

```

10067 ECL11
01
02
03
04 04122 006263
05 04123 000010
06 04124 176041
07 04125 177760
08 04126 006777
09 04127 111710
10 04130 156415
11 04131 101003
12
13
14 04135 006264
15
16
17 04137 006263
18 04140 000010
19 04141 102041
20 04142 177760
21 04143 034777
22 04144 115710
23 04145 152415
24 04145 152415
25 04145 101003
26
27
28 04153 006264
29
30
31
32
33
34
35
36
37
38
39
40
41

10068 ECL11
01
02
03
04
05 04154 006263
06 04155 000020
07
08 04156 006266
09 04157 101000
10 04160 105000
11 04161 131000
12 04162 141710
13 04165 141610
14 04164 004402
15 04165 157401
16 04166 170000
17 04167 112415
18 04170 136414
19
20
21 04175 006264
22
23
24
25 04176 006263
26 04177 000020
27
28 04200 006266
29 04201 105000
30 04202 131000
31 04203 155000
32 04204 145710
33 04205 145610
34 04206 020002
35 04207 163401
36 04214 170000
37 04211 136415
38 04212 142414
39
40
41 04217 006264

DSR4 2
SETUP 10
JSR #ENTIN
ADCO 2+183,2+183,SKP
177760
LDA 2,-1
DHXR 1,2
SUB# 2,2+183,SNR
MOV 0,0,SNR
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE

EX6C:
04122 006263
04123 000010
04124 176041
04125 177760
04126 006777
04127 111710
04130 156415
04131 101003
04135 006264
04137 006263
04140 000010
04141 102041
04142 177760
04143 034777
04144 115710
04145 152415
04145 152415
04145 101003
04153 006264

DSR4 3
SETUP 10
JSR #ENTIN
ADCO 3+183,3+183,SKP
177760
LDA 3,-1
DHXR 1,3
SUB# 3,3+183,SNR
MOV 0,0,SNR
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE

EX6D:
04137 006263
04140 000010
04141 102041
04142 177760
04143 034777
04144 115710
04145 152415
04145 152415
04145 101003
04153 006264

DSR4 4
SETUP 20
JSR #ENTIN
ADCO 2+183,2+183,SKP
177760
LDA 2,-1
DHXR 1,2
SUB# 2,2+183,SNR
MOV 0,0,SNR
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE

EX6E:
04154 006263
04155 000020
04156 006266
04157 101000
04160 105000
04161 131000
04162 141710
04165 141610
04164 004402
04165 157401
04166 170000
04167 112415
04170 136414
04175 006264
04176 006263
04177 000020
04200 006266
04201 105000
04202 131000
04203 155000
04204 145710
04205 145610
04206 020002
04207 163401
04214 170000
04211 136415
04212 142414
04217 006264

DSR4 5
SETUP 20
JSR #ENTIN
ADCO 3+183,3+183,SKP
170000
LDA 0+383,+2
AND 0+283,0+383,SKP
SUB# 0,0+283,SNR
SUB# 0+183,0+383,SKP
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE

EX6F:
04176 006263
04177 000020
04200 006266
04201 105000
04202 131000
04203 155000
04204 145710
04205 145610
04206 020002
04207 163401
04214 170000
04211 136415
04212 142414
04217 006264

DSR4 6
SETUP 20
JSR #ENTIN
ADCO 2+183,2+183,SKP
170000
LDA 0+383,+2
AND 1+283,1+383,SKP
SUB# 1,1+283,SNR
SUB# 1+183,1+383,SKP
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE

EX6G:
04154 006263
04155 000020
04156 006266
04157 101000
04160 105000
04161 131000
04162 141710
04165 141610
04164 004402
04165 157401
04166 170000
04167 112415
04170 136414
04175 006264
04176 006263
04177 000020
04200 006266
04201 105000
04202 131000
04203 155000
04204 145710
04205 145610
04206 020002
04207 163401
04214 170000
04211 136415
04212 142414
04217 006264

```



```

10071 ECL11
01
02
03
04 04354 006263
05 04351 000000
06
07 04352 006266
08 04353 111000
09 04354 153000
10 04355 151000
11 04356 151610
12 04357 151710
13 04358 024402
14 04359 107401
15 04360 000017
16 04361 106415
17 04362 106414
18
19
20 04351 006264
21
22
23
24 04352 006263
25 04353 000000
26
27 04354 006266
28 04355 115000
29 04356 151000
30 04357 151000
31 04358 155610
32 04359 155710
33 04360 030402
34 04361 133401
35 04362 172415
36 04363 106414
37 04364 106414
38
39
40 04373 006264
41
EX6K:
DSL12 2
SETUP 20
JSR #ENTIN
20
RAND
JSR #ENTRA
MOV 0,2
MOV 2,2+183
MOV 2,183,2+283
DHL 3,2
DHL 3,2
LDA 2+383,+2
AND 2+283,2+383,SKP
17
SUB# 2,2+383,SNR
SUB# 2+183,2+283,SNR
ERRR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE
21
EX6L:
DSL12 3
SETUP 20
JSR #ENTIN
20
RAND
JSR #ENTRA
MOV 0,3
MOV 3,3+183
MOV 3,183,3+283
DHL 3,3
DHL 3,3
LDA 3+383,+2
AND 3+283,3+383,SKP
17
SUB# 3,3+383,SNR
SUB# 3+183,3+283,SNR
ERRR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE
21
EX6M:
DADACU 0
SETUP 5
JSR #ENTIN
5
ZEROC
SUB 0,0
SUB 1,1
SUB 2,2
SUB0 3,3
ADCL 0,0
DAD 0,0
ADD# 0+183,0+283,SNR
MOV# 0+383,0+383,SNR
ERRR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE
21
EX6O:
DADACU 1
SETUP 5
JSR #ENTIN
5
ZEROC
SUB 0,0
SUB 1,1
SUB 2,2
SUB0 3,3
ADCL 1,1
DAD 1,1
ADD# 1+183,1+283,SNR
MOV# 1+383,1+383,SNR
ERRR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE
21
EX6P:
DADACU 2
SETUP 5
JSR #ENTIN
5
ZEROC
SUB 0,0
SUB 1,1
SUB 2,2
SUB0 3,3
ADCL 2,2
DAD 2,2
ADD# 2+183,2+283,SNR
MOV# 2+383,2+383,SNR
ERRR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE
21

```

```

10072 ECL11
01
02
03
04
05
06
07
08 04374 006263
09 04375 000000
10
11 04376 102400
12 04377 126400
13 04378 152400
14 04379 176400
15 04380 182120
16 04381 106210
17 04382 133015
18 04383 175014
19
20
21 04412 006264
22
23
24
25 04413 006263
26 04414 000000
27
28 04415 102400
29 04416 126400
30 04417 152400
31 04418 176400
32 04419 126100
33 04420 126100
34 04421 157015
35 04422 101014
36
37 04431 006264
38
39
40
41 04432 006263
42 04433 000000
43
44 04434 102400
45 04435 126400
46 04436 152400
47 04437 176400
48 04438 152120
49 04439 152120
50 04440 150210
51 04441 163015
52 04442 125014
53
54
55 04450 006264

```

```

*** DECIMAL ADD ***
TEST THAT RESULT GOES TO 0, ONL
INITIALIZE TEST.
DADACU 0
SETUP 5
JSR #ENTIN
5
ZEROC
SUB 0,0
SUB 1,1
SUB 2,2
SUB0 3,3
ADCL 0,0
DAD 0,0
ADD# 0+183,0+283,SNR
MOV# 0+383,0+383,SNR
ERRR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE
TEST THAT RESULT GOES TO 1, ONL
INITIALIZE TEST.
DADACU 1
SETUP 5
JSR #ENTIN
5
ZEROC
SUB 0,0
SUB 1,1
SUB 2,2
SUB0 3,3
ADCL 1,1
DAD 1,1
ADD# 1+183,1+283,SNR
MOV# 1+383,1+383,SNR
ERRR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE
TEST THAT RESULT GOES TO 2, ONL
INITIALIZE TEST.
DADACU 2
SETUP 5
JSR #ENTIN
5
ZEROC
SUB 0,0
SUB 1,1
SUB 2,2
SUB0 3,3
ADCL 2,2
DAD 2,2
ADD# 2+183,2+283,SNR
MOV# 2+383,2+383,SNR
ERRR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE

```

10873 ECL11

EX60:

01 DADAC0 3
02 SETUP 5
03 JSR #ENTLN
04 04451 006263
05 04452 000005
06 ZEROC
07 SUB 0,0
08 04454 126400
09 04455 152400
10 04456 176440
11 04457 176120
12 04460 174210
13 04461 107015
14 04462 151014
15 ERROR
16 LOOP
17 04467 006264
18
19
20

TEST THAT RESULT GOES TO 3, UNL
INITIALIZE TEST.

EX61:

01 10874 ECL11
02
03
04 04530 006263
05 04531 000040
06 04532 024400
07 04533 152421
08 04534 177700
09 04535 135000
10 04536 000012
11 04537 144210
12 04537 101000
13 04540 144210
14 04541 101000
15 04542 144210
16 04543 101000
17 04544 144210
18 04545 101000
19
20 04546 144210
21 04547 101000
22
23 04548 144210
24 04549 101000
25
26 04550 144210
27 04551 101000
28
29 04552 144210
30 04553 101000
31
32 04554 144210
33 04555 101000
34
35 04556 144210
36 04557 101000
37
38 04560 144210
39 04561 101000
40
41 04562 101003
42 04563 136414
43 ERROR
44 LOOP
45 04570 006264

TEST "DAD"
INITIALIZE TEST.

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

10873 ECL11

EX60:

01 DADAC0 3
02 SETUP 5
03 JSR #ENTLN
04 04451 006263
05 04452 000005
06 ZEROC
07 SUB 0,0
08 04454 126400
09 04455 152400
10 04456 176440
11 04457 176120
12 04460 174210
13 04461 107015
14 04462 151014
15 ERROR
16 LOOP
17 04467 006264
18
19
20

TEST THAT RESULT GOES TO 3, UNL
INITIALIZE TEST.

EX61:

01 10874 ECL11
02
03
04 04530 006263
05 04531 000040
06 04532 024400
07 04533 152421
08 04534 177700
09 04535 135000
10 04536 000012
11 04537 144210
12 04537 101000
13 04540 144210
14 04541 101000
15 04542 144210
16 04543 101000
17 04544 144210
18 04545 101000
19
20 04546 144210
21 04547 101000
22
23 04548 144210
24 04549 101000
25
26 04550 144210
27 04551 101000
28
29 04552 144210
30 04553 101000
31
32 04554 144210
33 04555 101000
34
35 04556 144210
36 04557 101000
37
38 04560 144210
39 04561 101000
40
41 04562 101003
42 04563 136414
43 ERROR
44 LOOP
45 04570 006264

TEST "DAD"
INITIALIZE TEST.

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

10873 ECL11

EX60:

01 DADAC0 3
02 SETUP 5
03 JSR #ENTLN
04 04451 006263
05 04452 000005
06 ZEROC
07 SUB 0,0
08 04454 126400
09 04455 152400
10 04456 176440
11 04457 176120
12 04460 174210
13 04461 107015
14 04462 151014
15 ERROR
16 LOOP
17 04467 006264
18
19
20

TEST THAT RESULT GOES TO 3, UNL
INITIALIZE TEST.

EX61:

01 10874 ECL11
02
03
04 04530 006263
05 04531 000040
06 04532 024400
07 04533 152421
08 04534 177700
09 04535 135000
10 04536 000012
11 04537 144210
12 04537 101000
13 04540 144210
14 04541 101000
15 04542 144210
16 04543 101000
17 04544 144210
18 04545 101000
19
20 04546 144210
21 04547 101000
22
23 04548 144210
24 04549 101000
25
26 04550 144210
27 04551 101000
28
29 04552 144210
30 04553 101000
31
32 04554 144210
33 04555 101000
34
35 04556 144210
36 04557 101000
37
38 04560 144210
39 04561 101000
40
41 04562 101003
42 04563 136414
43 ERROR
44 LOOP
45 04570 006264

TEST "DAD"
INITIALIZE TEST.

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

10075 ECL11

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

```

EX6U:

```

SETUP 40
JSR #ENTIN
40
SUB 3,3
ADCU 2,2,SXP
177767
LDA 0,-1
DAD 3,2
DAD 3,2
SUB# 0,2,SNK
MOV 0,0,SZC
ERROR
LOOP
JSR #ENTLU
ITERATE TEST ROUTINE

```

EX6V:

```

SETUP 20
JSR #ENTIN
20
ADCU 1,1,SKP
5,
LDA 2,-1
DAD 1,0
MOV 0,0,SZC
SUB# 0,2,SZR
ERROR
LOOP
JSR #ENTLU
ITERATE TEST ROUTINE

```

10076 ECL11

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

```

EX6W:

```

SETUP 20
JSR #ENTIN
20
SUB 1,1
ADCU 0,0
DAD 0,1
LDA 2,*0
SUB# 2,1,SNR
MOV 1,1,SNR
ERROR
LOOP
JSR #ENTLU
ITERATE TEST ROUTINE

```

EX6X:

```

SETUP 20
JSR #ENTIN
20
RAND
JSR #ENTRA
CALL HEXN
HEXN
JSR #ICAL
MOV 0,2
SUB0 1,1
DD 4
DAD 1,2
DHR 1,2
ENDC
DAD 1,2
DHR 1,2
ENDC
DAD 1,2
DHR 1,2
ENDC
DAD 1,2
DHR 1,2
ENDC
MOV 0,0,SNR
SUB# 3,0,SZR
ERROR
LOOP
JSR #ENTLU
ITERATE TEST ROUTINE

```







```

10081 ECL11
01
02
03
04 05131 00263
05 05132 00264
06
07 05133 00266
08 05134 00267
09 05135 05251
10 05136 11400
11 05137 11100
12
13 05140 13421
14 05141 11571
15 05142 10510
16 05143 10100
17 05144 13421
18 05145 11571
19 05146 10510
20 05147 10100
21 05150 13421
22 05151 11571
23 05152 10510
24 05153 10100
25 05154 13421
26 05155 11571
27 05156 10510
28 05157 10100
29 05160 03475
30 05161 10241
31
32
33 05166 00264
34

*** DECIMAL SUBTRACTION ***
TEST "DAD"
INITIALIZE TEST.
SETUP 40
JSR PENTIN
40
RAND
JSR PENTRA
MOVZ 0,1,SKP
052525
EX7F.*
CUM 0,3
MOV 0,2
DO 4
DAD 1,3
DHRX 1,3
HXR 1,1
MOV 0,0
DAD 1,3
DHRX 1,3
HXR 1,1
MOV 0,0
DAD 1,3
DHRX 1,3
HXR 1,1
MOV 0,0
DAD 1,3
DHRX 1,3
HXR 1,1
MOV 0,0
DAD 1,3
DHRX 1,3
HXR 1,1
MOV 0,0
LDA 3,EX7F.
SUB# 3,0,JSZ
ERRR
LOOP
JSR PENTLO
ITERATE TEST ROUTINE

10082 ECL11
01
02
03
04
05
06
07 05167 00263
08 05170 00005
09
10
11 05171 10240
12 05172 10240
13 05173 10240
14 05174 17640
15 05175 10252
16 05176 10030
17 05177 130015
18 05200 175014
19
20
21 05205 10240
22 05206 10240
23 05207 10240
24 05210 17640
25 05211 10252
26 05212 10030
27 05213 15015
28 05214 101014
29
30
31
32
33 05221 10240
34 05222 10240
35 05223 10240
36 05224 17640
37 05225 10252
38 05226 10030
39 05227 10015
40 05230 10014
41
42
43
44 05245 10240
45 05246 10240
46 05247 10240
47 05248 17640
48 05249 176520
49 05242 174310
50 05243 107015
51 05244 151014
52
53
54 05251 00264
55
56
SETUP 5
JSR PENTIN
5
DSBACD 0
ZERUAC
SUB 0,0
SUB 1,1
SUB 2,2
SUB 3,3
SUBZL 0,0
DSB 0,0
ADD# 0+100,0+200,SNK
MOV# 0+300,0+300,JSZ
ERRR
DSBACD 1
ZERUAC
SUB 0,0
SUB 1,1
SUB 2,2
SUB 3,3
SUBZL 1,1
DSB 1,1
ADD# 1+100,1+200,SNK
MOV# 1+300,1+300,JSZ
ERRR
DSBACD 2
ZERUAC
SUB 0,0
SUB 1,1
SUB 2,2
SUB 3,3
SUBZL 2,2
DSB 2,2
ADD# 2+100,2+200,SNK
MOV# 2+300,2+300,JSZ
ERRR
DSBACD 3
ZERUAC
SUB 0,0
SUB 1,1
SUB 2,2
SUB 3,3
SUBZL 3,3
DSB 3,3
ADD# 3+100,3+200,SNK
MOV# 3+300,3+300,JSZ
ERRR
LOOP
JSR PENTLO
ITERATE TEST ROUTINE

```







10089 ECL11

```

01
02
03 04 05535 006263
05 05536 000040
06
07 05537 006266
08
09 05540 006220
10 05541 001377
11 05542 115020
12 05543 111060
13 05544 000004
14 05545 140310
15 05546 101710
16 05547 111510
17
18 05547 140310
19 05550 101710
20 05551 111510
21
22 05552 140310
23 05553 101710
24 05554 111510
25
26 05555 140310
27 05556 101710
28 05557 111510
29
30 05560 107014
31
32 05565 006264
33
34
35

```

```

EX70:
SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
CALL HEXN
JSR #ICAL
HEXN
MOVZ 0,3
DO 4
DSB 2,0
DHR 1,0
HR 1,2
.ENDC
OSB 2,0
DHR 1,0
HR 1,2
.ENDC
DSB 2,0
DHR 1,0
HR 1,2
.ENDC
OSB 2,0
DHR 1,0
HR 1,2
.ENDC
JSR #ENTLO
LOOP
JSR #ENTLO

```

```

/TEST "DSB"
/INITIALIZE TEST.
/C(A0)=RANDOM #
/DIGIT RANGE=0-9
/CALL SUBROUTINE HEXN
/C(A0)=RANDOM
/C(A0)=RANDOM, C(CARRY)=1
/DIFF TO C(A0), BITS 12-15
/SHIFT RESULT TO C(A1)
/RANDOM-RANDOM SHOULD=0
/DIFF TO C(A0), BITS 12-15
/SHIFT RESULT TO C(A1)
/RANDOM-RANDOM SHOULD=0
/DIFF TO C(A0), BITS 12-15
/SHIFT RESULT TO C(A1)
/RANDOM-RANDOM SHOULD=0
/DIFF TO C(A0), BITS 12-15
/SHIFT RESULT TO C(A1)
/RANDOM-RANDOM SHOULD=0
/C(A0)=DSB RESULT
/C(A0) SHOULD=0 VIA SHIFT
/ITERATE TEST ROUTINE

```

10090 ECL11

```

01
02
03 05566 006263
04 05567 000020
05
06 05570 006266
07
08 05571 006220
09 05572 001377
10 05573 105040
11 05574 174510
12 05575 020024
13 05576 105310
14 05577 105710
15
16 05577 105310
17 05600 105710
18
19 05601 105310
20 05602 105710
21
22 05603 105310
23 05604 105710
24
25 05605 112414
26
27 05612 006264
28
29
30
31 05613 006263
32 05614 000010
33
34 05015 006266
35
36 05016 006220
37 05017 001377
38 05020 115020
39 05021 124510
40 05022 134310
41 05023 126520
42 05024 134210
43 05025 116414
44
45 05032 006264
46
47

```

```

EX70:
SETUP 20
JSR #ENTIN
20
RAND
JSR #ENTRA
CALL HEXN
JSR #ICAL
HEXN
MOVZ 0,1
XOR 3,3
DO 4
DSB 3,1
DHR 1,1
.ENDC
DSB 3,1
DHR 1,1
.ENDC
DSB 3,1
DHR 1,1
.ENDC
DSB 3,1
DHR 1,1
.ENDC
SUBS 0,2,SZR
ERROR
LOOP
JSR #ENTLO

```

```

/ITERATE TEST ROUTINE
/TEST "DSB"
/INITIALIZE TEST.
/C(A0)=RANDOM #
/DIGITS 0-9
/CALL SUBROUTINE HEXN
/C(A0)=1
/C(A0)+1
/C(A0)=ORIGINAL AND CORRECT
/C(A0)=DSB/DAD RESULT
/ITERATE TEST ROUTINE

```

10091 ECL11

```

01
02 05633 014204 EX7H:
03 05634 000437
04 05635 010203
05 05636 101001
06 05637 063077
07 05640 020205
08 05641 040204
09
10 05642 000477
11 05643 101112
12 05644 000403
13 05645 122470
14 05646 000034
15 05647 143770
16 05648 004000
17 05651 101004
18 05652 000406
19
20 05653 000221
21 05654 001226
22 05655 120020
23 05656 024203
24 05657 000224
25
26 05660 030100=PSCK1:
27 05661 021400
28 05662 101005
29 05663 000410
30 05664 015403
31 05665 000406
32 05666 002677
33 05667 021403
34 05670 035404
35 05671 041776
36 05672 001400
37
38 05073 006220
39
40 05674 001014
41
42
43
44

```

```

10092 ECL11
01
02
03
04
05 05675 000000 AUTO: 0
06 05676 000000 DEV: 0
07 05677 000000 CATSW: 0
08 05700 000000 PCNT: 0
09 05701 000000 RTN: 0
10 05702 000000 SMREG: 0
11
12 05703 005703 PRGENO: PRGENO
13
14 05704 047503 .TXT /COPYRIGHT(C)06G,1974,75,76
15 054520
16 044522
17 044107
18 024124
19 024503
20 043504
21 020103
22 034401
23 032007
24 033454
25 020005
26 033007
27 05721 040101 ALL NIGHTS RESERVED/
28 020114
29 044522
30 044107
31 051524
32 051040
33 051505
34 051105
35 045526
36 000104
37
38 05733 141705 UIRT: .TXTE IECLIPSE11J;
39 144714
40 051520
41 130705
42 031601
43 000000
44 05741 000000
45 05742 000200
46 05743 175772
47 05744 000000
48 05745 000000
49 05746 000000
50 05747 000000
51 05750 000000
52
53
54 .EMU DTOSB
55 00010-000006
56 000012
57 000017
58 000240
59 17770
60 000212

```

```

; *****EGGS & DIRT DATA BLOCKS*****
;
; DTOS AUTO MODE SWITCH
; PRIMARY DEVICE CODE TO BE TESTED
; CAT SWITCH, SET IF CAT LOADED
; PASS COUNT, # OF TIMES TO RUN
; RETURN POINT TO RESTORE DTOS
; DEFAULT SWITCH REGISTER

```

\*\*\* END OF TEST ROUTINES \*\*\*

0893 ECL11  
 01 10440P  
 02 02200Z  
 03 000077  
 04 000000  
 05 000011  
 06 033031  
 07 000144  
 08 000200  
 09 000340  
 10 001777  
 11 000263  
 12 000257  
 13 000004

0894 ECL11  
 ACP 000020  
 AC1 000020  
 AC2 000027  
 AC3 000030  
 18/16 21/08 21/30 22/21 22/34 23/05 23/28  
 23/58 22/22 22/35 23/15 23/30 23/46  
 18/17 22/22 22/35 23/16 23/32 23/47  
 18/18 22/33 23/36 23/42 34/32 34/42  
 18/19 23/34 23/59 34/22 34/32 36/27  
 35/10 35/21 35/32 36/11 36/23 36/31  
 36/47 37/27 37/38 38/16 38/31 38/52  
 40/20 40/45 41/19 41/37 42/19 42/38 43/19  
 43/32 43/47 44/19 44/38 44/41 44/52 45/18  
 45/34 46/19 46/37 46/53 47/19 47/34 47/48  
 48/16 49/19 49/29 49/39 49/49 50/18 50/36  
 51/19 51/38 52/49 53/09 53/29 54/15 54/38  
 54/55 55/18 55/33 55/48 56/14 57/31 57/51  
 57/48 58/16 58/33 58/47 59/15 59/31 60/15  
 60/31 61/16 61/32 61/48 62/15 62/34 62/53  
 63/16 63/37 63/56 64/18 64/37 64/56 65/13  
 65/27 65/41 66/13 66/27 66/41 67/13 67/27  
 68/20 68/40 69/19 69/39 70/19 70/39 71/19  
 71/39 72/20 72/37 72/54 73/16 73/36 73/51  
 74/44 75/15 75/29 76/13 76/42 77/49 78/48  
 79/59 80/39 81/32 82/20 82/31 82/42 82/53  
 83/11 83/22 83/33 83/44 84/19 85/27 85/39  
 86/27 87/29 88/38 89/32 90/27 90/45

ACU 000104 MC 8/33  
 ADDAL 000083 MC 8/20  
 ANCAC 000132 MC 9/12  
 ASSEN 000072 MC 8/27  
 AUTO 000575 20/14  
 BANBY 000247 18/41 18/42 18/50  
 BCA1 001412 18/37 24/33 25/37  
 BCS1 001341 33/13 77/15 78/14  
 BCS2 001356 33/18 33/33  
 BEGIN 001442 32/04 86/14 87/14  
 BGNAD 000202 32/12 32/24  
 CAL 000706 16/47 25/40 34/07  
 CAL0 000237 17/45 17/47  
 CAL1 000240 18/08 24/22  
 CAL2 000241 18/26 24/22 24/29  
 CALL 000010 MC 18/27 24/23 24/30  
 7/18 18/44 25/38 24/27 24/31  
 78/12 86/07 86/12 76/21 77/08 77/13 78/07  
 00/07 90/35 91/37 87/08 87/12 88/07 89/08  
 17/31 18/11 19/32 25/08 92/07  
 26/42 26/57 28/60 28/08  
 28/30 28/47 29/81 28/08  
 28/33 28/39 28/46 28/57  
 28/27 28/57 28/53  
 18/21 28/04 28/50 28/57  
 18/22 28/30 28/50 28/57  
 28/65 28/62 28/03  
 11/03 23/07 23/25 72/46 73/02  
 18/20 72/23 72/46 73/02  
 16/03 92/06 92/38 60/02 60/18  
 000576 17/05 59/02 59/18  
 000553 MC 12/30 60/02 60/18







0499 ECL11

|              |       |       |       |       |       |       |       |
|--------------|-------|-------|-------|-------|-------|-------|-------|
| IMSS 000221  | 18/09 | 20/04 | 20/08 | 20/19 | 20/21 | 22/24 | 23/   |
| INIT 000505  | 23/22 | 23/43 | 21/20 |       |       |       |       |
| INIT 000507  | 18/50 | 18/56 | 21/06 |       |       |       |       |
| IPDC 000224  | 21/24 | 21/30 |       |       |       |       |       |
| IPUCT 000223 | 18/12 | 20/07 | 23/21 | 23/29 | 23/33 | 23/35 | 23/   |
| ISIZE 000222 | 23/42 | 19/06 |       |       |       |       |       |
| ITR 000206   | 18/10 | 21/27 | 22/10 | 22/31 |       |       |       |
| ITRCT 000207 | 17/52 | 21/12 | 22/05 | 22/11 |       |       |       |
| ITRCL 000211 | 17/55 | 22/28 | 22/28 | 22/38 | 23/08 |       |       |
| ITRER 000210 | 17/54 | 21/19 | 22/07 | 22/40 | 23/10 | 23/14 | 24/   |
| ITRET 000201 | 17/46 | 21/07 | 21/31 | 22/48 |       |       |       |
| JMPR 000232  | 7/30  | 37/38 |       |       |       |       |       |
| KCRLF 001315 | 20/09 | 20/22 |       |       |       |       |       |
| LISIN 000215 | 17/59 | 21/16 |       |       |       |       |       |
| LOBAC 000232 | 10/03 |       |       |       |       |       |       |
| LOBAL 000262 | 18/15 |       |       |       |       |       |       |
| LOOP 000000  | 7/05  | 34/52 | 35/10 | 35/21 | 35/32 | 36/11 | 36/3  |
|              |       | 37/47 | 37/44 | 38/16 | 38/31 | 39/23 | 39/5  |
|              |       | 40/20 | 40/45 | 41/19 | 41/37 | 42/19 | 42/38 |
|              |       | 43/32 | 43/47 | 44/53 | 45/18 | 46/19 | 46/3  |
|              |       | 46/53 | 47/19 | 47/34 | 47/48 | 48/16 | 48/49 |
|              |       | 50/36 | 51/19 | 51/38 | 52/49 | 53/09 | 50/1  |
|              |       | 54/38 | 54/55 | 55/18 | 55/33 | 55/48 | 57/1  |
|              |       | 57/31 | 57/48 | 58/16 | 58/33 | 58/47 | 59/31 |
|              |       | 60/31 | 61/16 | 61/32 | 61/48 | 62/15 | 62/31 |
|              |       | 62/53 | 63/18 | 63/37 | 63/56 | 64/18 | 64/35 |
|              |       | 65/13 | 65/27 | 66/41 | 66/27 | 66/41 | 67/13 |
|              |       | 67/27 | 68/20 | 68/40 | 69/19 | 69/39 | 70/19 |
|              |       | 71/19 | 71/39 | 72/20 | 72/37 | 72/54 | 73/36 |
|              |       | 73/51 | 74/44 | 75/15 | 75/29 | 76/13 | 77/45 |
|              |       | 78/48 | 79/59 | 80/39 | 81/32 | 82/53 | 83/11 |
|              |       | 83/33 | 83/44 | 84/19 | 85/27 | 85/39 | 86/27 |
|              |       | 88/38 | 89/32 | 90/27 | 90/45 |       | 87/29 |
|              |       | 18/51 | 18/57 | 22/04 |       |       |       |
| LOP 000611   | 22/13 | 22/37 |       |       |       |       |       |
| LOP1 000521  | 22/20 | 22/40 |       |       |       |       |       |
| LOP2 000501  | 22/06 | 22/40 |       |       |       |       |       |
| LOP3 000503  | 17/57 | 22/04 |       |       |       |       |       |
| LOPRE 000213 | 10/27 | 34/12 | 22/09 | 22/49 |       |       |       |
| LOBAC 000323 | 12/16 | 57/17 | 57/34 | 58/02 | 34/42 |       |       |
| LNSHI 000521 | 9/30  |       |       |       | 58/19 |       |       |
| LNSAC 000212 | 18/04 | 19/37 | 25/14 | 25/29 |       |       |       |
| LXALD 000217 | 17/23 | 19/09 | 20/06 | 24/46 |       |       |       |
| MCFID 000011 | 20/05 | 30/47 | 26/59 |       |       |       |       |
| MES12 001301 | 18/24 | 20/49 |       |       |       |       |       |
| MESKE 000233 | 18/09 | 26/50 |       |       |       |       |       |
| MESS 001126  | 26/52 | 27/01 |       |       |       |       |       |
| MESS1 001131 | 18/03 | 19/35 | 24/36 | 25/15 |       |       |       |
| MINLO 000210 | 17/47 | 19/21 | 19/29 |       |       |       |       |
| NOCAT 000524 | 17/47 | 19/05 |       |       |       |       |       |
| NSSTR 000504 | 17/30 |       |       |       |       |       |       |
| OFF 000174   | 30/04 | 51/21 | 21/22 | 23/19 | 91/04 | 91/25 |       |
| OR 000171    | 17/48 | 26/12 | 91/02 |       |       |       |       |
| PASME 001220 | 17/48 |       |       |       |       |       |       |
| PASS 000203  | 17/49 |       |       |       |       |       |       |
| PASS1 000204 | 17/49 |       |       |       |       |       |       |

0100 ECL11

|              |       |       |       |       |       |       |       |
|--------------|-------|-------|-------|-------|-------|-------|-------|
| PASSV 000205 | 17/50 | 91/07 |       |       |       |       |       |
| PCNT 000500  | 92/08 |       |       |       |       |       |       |
| PDEC 001053  | 18/12 | 26/04 |       |       |       |       |       |
| PDEC1 001100 | 26/26 | 26/44 |       |       |       |       |       |
| PDEC2 001112 | 26/32 | 26/36 |       |       |       |       |       |
| PDEC3 001117 | 26/07 | 26/17 | 26/29 | 26/41 |       |       |       |
| PDERE 000234 | 18/23 | 26/05 | 26/15 | 26/45 |       |       |       |
| PERCE 001233 | 22/25 | 30/09 |       |       |       |       |       |
| PCTI 001065  | 18/11 | 20/14 |       |       |       |       |       |
| PRGEN 000503 | 18/03 | 19/16 | 92/12 |       |       |       |       |
| PSCK1 000506 | 91/18 | 91/26 |       |       |       |       |       |
| RAN 000236   | 18/25 | 24/04 |       |       |       |       |       |
| RAND 000012  | 7/14  | 25/12 | 24/16 |       |       |       |       |
|              | 38/05 | 39/04 | 40/06 | 40/26 |       |       |       |
|              | 41/25 | 43/38 | 45/24 | 46/25 |       |       |       |
|              | 46/43 | 47/06 | 47/40 | 50/24 |       |       |       |
|              | 52/06 | 53/15 | 54/04 | 54/44 |       |       |       |
|              | 57/38 | 58/06 | 58/23 | 59/06 |       |       |       |
|              | 60/22 | 61/07 | 61/23 | 61/39 |       |       |       |
|              | 63/06 | 63/25 | 63/44 | 64/06 |       |       |       |
|              | 68/27 | 69/06 | 69/26 | 70/06 |       |       |       |
|              | 76/19 | 77/06 | 78/05 | 79/06 |       |       |       |
|              | 85/06 | 86/05 | 87/06 | 88/05 |       |       |       |
|              | 25/21 | 25/08 | 25/22 | 25/22 |       |       |       |
| REL 001014   | 25/21 | 25/32 |       |       |       |       |       |
| REL1 001043  | 25/17 | 25/38 |       |       |       |       |       |
| REL2 001051  | 17/58 | 18/39 | 28/32 | 21/14 | 23/38 | 24/24 | 25/23 |
| RELOC 000214 | 25/32 |       |       |       |       |       |       |
|              | 28/16 | 28/20 |       |       |       |       |       |
| REST 001214  | 92/09 |       |       |       |       |       |       |
| RTRN 000501  | 20/20 |       |       |       |       |       |       |
| SETSM 001317 | 7/09  | 31/01 |       |       |       |       |       |
| SETUP 000005 | 36/34 | 37/03 | 35/02 | 35/13 | 35/24 | 36/03 | 36/14 |
|              | 40/03 | 40/23 | 37/38 | 38/02 | 38/21 | 39/01 | 39/26 |
|              | 43/22 | 43/35 | 41/03 | 41/22 | 42/03 | 42/22 | 43/03 |
|              | 46/40 | 47/03 | 44/06 | 45/02 | 45/21 | 46/03 | 46/22 |
|              | 50/21 | 51/03 | 47/22 | 47/37 | 48/03 | 49/06 | 49/02 |
|              | 54/16 | 54/41 | 51/22 | 52/03 | 53/01 | 53/12 | 54/01 |
|              | 57/18 | 57/35 | 55/07 | 55/22 | 55/37 | 56/03 | 57/03 |
|              | 60/03 | 60/19 | 58/03 | 58/20 | 58/36 | 59/03 | 59/19 |
|              | 62/38 | 63/03 | 61/04 | 61/26 | 61/36 | 62/03 | 62/19 |
|              | 65/03 | 65/17 | 63/22 | 63/41 | 64/03 | 64/22 | 64/41 |
|              | 67/17 | 68/04 | 65/03 | 65/17 | 66/17 | 66/31 | 67/03 |
|              | 73/39 | 74/03 | 68/24 | 69/03 | 69/23 | 70/03 | 70/23 |
|              | 78/02 | 79/03 | 71/03 | 71/23 | 72/04 | 72/41 | 73/03 |
|              | 83/25 | 83/36 | 73/39 | 74/03 | 75/02 | 75/18 | 76/03 |
|              | 86/02 | 87/03 | 78/02 | 79/03 | 80/02 | 81/03 | 82/06 |
|              | 18/10 | 24/36 | 84/01 | 85/03 | 85/30 | 86/02 | 87/03 |
|              | 20/17 | 20/28 | 90/02 | 90/02 | 90/02 | 91/02 | 91/03 |
|              | 19/27 | 19/35 | 24/47 | 24/47 | 24/47 | 24/47 | 24/47 |
| START 000562 | 8/13  | 20/04 | 22/45 | 28/10 | 91/13 | 92/10 |       |
| STLOC 000531 | 17/34 |       |       |       |       |       |       |
| SUBAL 000053 | 20/26 |       |       |       |       |       |       |
| SWRES 000534 | 9/21  |       |       |       |       |       |       |
| SWREG 000702 | 9/03  |       |       |       |       |       |       |
| XCHAC 000172 | 8/00  |       |       |       |       |       |       |
| XORAC 000132 | 8/00  |       |       |       |       |       |       |
| ZERQA 000042 | 82/21 |       |       |       |       |       |       |

0101 ECL11

.BCV 000245  
.BC1 000244  
.BC2 000243  
.BC3 000242  
.EGS 000010  
.RAND 000750

18/32 32/07 32/18 32/24 32/27 33/18 33/28  
18/31 32/06 32/16 32/33 33/15 33/19 33/34  
18/30 32/05 32/34 33/14 33/35  
18/29 32/04 32/35 33/13 33/36  
17/22 23/54 91/26  
18/53 18/59 24/04



**DataGeneral**

---

---

**DIAGNOSTIC  
LISTING**

---

---

LISTING

096-000241-02

PROGRAM

EXERCISER FOR ECLIPSE  
PART 3

TAPE

095-000226-02

ABSTRACT

'ECLIPSE20' IS AN EXERCISER PROGRAM USED TO TEST THE RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF THE ECLIPSE COMPUTER. 'ECLIPSE20' EXERCISES THE EXTENDED INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES OF ITS RELIABLE OPERATION.

COPYRIGHT (C) DATA GENERAL CORPORATION, 1974, 1976  
ALL RIGHTS RESERVED. PRINTED IN U.S.A.



```

01 0001 ECL20 MACRO REV 03.00      14144115 08/08/76
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

*****
/ NAME: ECLIPSE20.SR              PART NUMBER: 094-000625
/
/ DESCRIPTION: ECLIPSE EXERCISER, PART 3
/ REVISION HISTORY:
/
/ REV.      DATE
/ 00        08/02/74
/ 01        12/20/74
/ 02        08/08/76
/
/ COPYRIGHT (C) DATA GENERAL CORPORATION, 1974, 1976
/ ALL RIGHTS RESERVED.
/ *****

```

```

10002 ECL20
02
03
04
05
06
07

```

```

.YITL ECL20
/ECLIPSE20
/ECLIPSE20 - CONTINUATION OF ECLIPSE11
/PART 3 OF EXERCISER FOR ECLIPSE
/

```

```

18003 ECL20
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

EXERCISER FOR ECLIPSE1 PART 3
PROGRAM NAME
-----
ECLIPSE20
GENERAL DESCRIPTION
-----
ECLIPSE20 IS AN EXERCISER PROGRAM USED TO TEST THE
RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF
THE ECLIPSE COMPUTER. ECLIPSE20 EXERCISES THE EXTENDED
INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES
OF ITS RELIABLE OPERATION.

12.2 THE INSTRUCTIONS EXERCISED ARE AS FOLLOWS:
MULTI, DIV, MULS, DIVS, 9LM, BAH, SZB, SZBO, STO, STZ, LDB AND
STB

12.3 LOCATIONS 200 TO 205 IN PAGE 0 ARE FIXED FOR ECLIPSE20
PROGRAM.
LOCATION 203 KEEPS TRACK OF NUMBER OF PASSES RUN
THROUGH ECLIPSE20 PROGRAM.
LOCATION 201 KEEPS TRACK OF THE TEST RUNNING AT
PRESENT AND IS USEFUL FOR DEBUG WHEN LOOPING
OCCURS IN THE PROGRAM.
LOCATION 202 CONTAINS THE STARTING ADDRESS OF
ECLIPSE20 PROGRAM.
LOCATION 200 IS USED BY DTDS.
LOCATION 204 KEEPS TRACK OF INTERNAL PASS COUNT
WHICH IS FIXED BY LOCATION 205.

12.4 FIRST PASS THROUGH ECLIPSE20 TEST WILL RUN SUPERFAST.
NEXT PASSES WILL RUN SLOWER AS EACH TEST IS RUN SEVERAL
TIMES TO RUN ALL RANDOM NUMBER COMBINATIONS.

13. MACHINE REQUIREMENTS
-----
ECLIPSE PROCESSOR
4K READ-WRITE MEMORY
CONSOLE EQUIPMENT

18004 ECL20
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

SWITCH SETTINGS
-----
THIS PROGRAM USES DATA SWITCHES AS FOLLOWS
SW"0" - USE CONTENTS OF "SWREG" IF 0
USE DATA SWITCHES IF 1
SW"1" - LOOP ON FAILING TEST IF 0
PROCEED TO NEXT TEST IF 1
SW"2" - OUTPUT TO TTY IF 0
INHIBIT PRINTING TO TTY IF 1
SW"3" - DO NOT PRINT X ERRORS IF 0
PRINT FAILURE RATE IF 1
SW"4" - PRINT PASS COUNT IF 0
SW"5" - INHIBIT PRINTING TO LINE PRINTER IF 0
OUTPUT TO LINE PRINTER IF 1

EDTOS PRESUMES THAT ALL SWITCHES ARE 0. RETURN IS
MADE TO THE OPER SYS UPON AN ERROR AFTER FIRST PRINTING
THE FAILURE REPORT. IN STAND ALONE MODE, THE OPERATOR
IS ASKED TO PRESET THESE SWITCHES (THE MACHINE HALTS
FOR HIM TO DO SO) AND PRESS CONTINUE. THIS PRESET VALUE
CAN BE CHANGED AT ANY TIME BY SETTING DATA SWITCH "0"
TO A ONE. THE PROGRAM WILL THEN FOLLOW ACTUAL SETTINGS.

14.2 STAND ALONE STARTING ADDRESS = 200
IF ICATI OR IKITENT HAS LOADED FROM DTDS AND RESRST
WAS NEEDED, THEN USE AS FOLLOWS:
STARTING ADDR = 170 (FOR START WITH NO ICATI)
STARTING ADDR = 171 (FOR START WITH ICATI)

14.3 MONITOR LOCATION 203 TO CHECK THE CURRENT PASS COUNT

14.4 MONITOR LOCATION X6000 TO MAKE SURE THAT ICATI OR
IKITENT IS RUNNING. IN CASES WHERE PROGRAM IS
STARTED WITH ICATI OR IKITENT LOCATION X6000 WILL SHOW
A PATTERN CHANGING FROM ZEROS TO ALL ONES
TO AN INC/SWAP PATTERN.

(X= THE NUMBER OF THE HIGHEST MEMORY MODULE IN THE
SYSTEM AND MAY BE A VALUE 0 - 7)

```



10005 ECL20

```

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
OPERATING PROCEDURE/OPERATOR INPUT
-----
LOAD THE PROGRAM VIA THE BINARY LOADER OR INSERT A
RELOADED MEMORY MODULE.
SET SWITCHES TO 200.
PRESS START.
PROGRAM WILL HALT AFTER PRINTING THE MESSAGE
'SET DATA SWITCHES AND PRESS CONTINUE'.
SET REQUIRED SWITCH SETTINGS AND PRESS CONTINUE.
THE PROGRAM WILL RUN UNTIL MANUALLY STOPPED. IN CASE
OF MALFUNCTIONING, THE PROGRAM WILL PRINT ERROR
MESSAGE AND TAKE APPROPRIATE ACTION AS PER THE SW
SETTINGS.
PROGRAM OUTPUT/ERROR DESCRIPTION
-----
FOR ANY ERRORS DETECTED, THE PROGRAM WILL PRINT ERROR
REPORT OR X FAILURES DEPENDING UPON THE SW SETTINGS.
ERROR REPORT CONSISTS OF ALL ACCUMULATORS, CARRY,
RELOCATED PROGRAM COUNTER OF THE TEST THAT IS FAILING
AND PC IN THE LISTING AT THE TIME OF FAILURE.
THE PROGRAM WILL LOOP IN THE TEST THAT IS FAILING IF
SW"1" IS 0.
THE PRINTING OF ERROR REPORT CAN BE ABORTED BY SETTING
SW"2" TO 1.
IF LOOPING OCCURS IN THE PROGRAM, STOP THE COMPUTER
AND CHECK LOCATION 201 TO FIND OUT THE TEST THAT WAS
RUNNING BEFORE THE LOOPING OCCURRED.

```

10006 ECL20

```

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
PROGRAM DESCRIPTION/THEORY OF OPERATION
-----
EACH TEST IS COMPLETE IN ITSELF, SO THE PROGRAM CAN
BE STARTED FROM ANY TEST WITHOUT CAUSING ANY
INITIALIZATION ERRORS.
WHEN 'ECLIPSE201' IS STARTED AT LOCATION 200 OR BY
DIOS, IT WILL SIZE UP THE MEMORY AND WILL PRINT
THE TOP OF THE MEMORY.
AFTER SETTING UP THE SWITCHES AND PRESSING CONTINUE,
THE EXECISER WILL RUN THE FIRST PASS VERY FAST. IN
THE FIRST PASS EACH TEST IS RUN ONLY ONCE. ALL OTHER
PASSES WILL TAKE MORE TIME AS EACH TEST IS RUN
ACCORDING TO THE TEST COUNT SPECIFIED IN EACH TEST.
REFER TO THE LISTING TO FIND OUT THE INFORMATION
ABOUT EACH TEST.
RESTRICTIONS/MISC
-----
CERTAIN INSTRUCTIONS LIKE RLM, XCT, BAM, ETC.,
DO ALLOW INTERRUPTS TO OCCUR DURING THEIR
EXECUTION. THIS FEATURE OF THOSE INSTRUCTIONS IS
NOT CHECKED IN THIS TEST.

```

10007 ECL20

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

I
*** MACRO DEFINITIONS ***

.MACRO LOOP
JSR @ENTLO
ITERATE TEST ROUTINE,....
X

.MACRO SETUP
JSR @ENTIN
INITIALIZE TEST,....
X

.MACRO RAND
JSR @ENTRA
IC(AC0)=RANDOM #
X

.MACRO BRAND
JSR @ENTRB
X
.MACRO ERROR
JMP *+2
**
** JMP *+3
**
** STA 3,AC3
**
** JSR @ENTER
X

.MACRO STACK
JSR @ISTK
INITIALIZE STACK
IF FAULT ADDRESS IS A1
X

.MACRO PSHSP
SETUP 20
STACK PSHS:A1
MOV0 B,B
PSH A1,A1
LDA 0,SP
FOR A INCREMENT OF 0N1
IC(AC0)=STACK POINTER=C(40)
MOV 0,PSZC
SUB# 0,1,SZR
PSHS:A1: ERROR
LOOP
X

.MACRO TRAPER
JSR @ITRP
IF STACK FAULT, GO TO A1
IF TRAP ORIGIN IN CONTENTS OF A2
A1
A2
A3
IF SUBROUTINE ADDRESS IS A4
IF STACK POINTERS SET TO C(8REG)
X

```

10008 ECL20

```

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

.MACRO TRTST
LDA 0,BHEG
LDA 1,A1+1
ADD 0,1
XOP A1,A1,A2
SUB# 2,3,SMR
SUB# 1,2,SZR
ERROR
X

.MACRO ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUB 3,3
X

.MACRO BANER
** STA 3,ACS
** JSR @ENT0E
X

.MACRO MDERR
** STA 3,ACS
** JSR @ENT0E
X

.MACRO RANST
JSR @I,RAN
RANDOM # TO AC8=2
X

```

```

ITEST FOR ACS/ACD ADDRESS
STORED IN C(AC2=3),C(AC1)
CORRECT STACK LOCATION
ICAP # A2

```

ZERO ALL AC'S

```

10000 ECL20
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

```

```

      OVLIN
      LDA 0,0,+2
      MOVZ 0,0,SKP
      AZ
      STA 0,0,FF
      LDA 0,0,A1
      STA 0,0,SP
      STA 0,0,SL
      STA 0,0,FP
      X
      OVTI
      SETUP 000
      OVLIN BBEG,OV10A1
      LDA 1,0,A1
      ADD 0,1
      STA 1,0,SL
      SAVE
      0
      ERROR
      OVL0A11 LOOP
      X
      OVLIN
      LDA 0,0,FPBK
      STA 0,0,SP
      LDA 0,0,BBEG
      STA 0,0,SL
      LDA 0,0,A1
      STA 0,0,FP
      X
      OVTIP
      LDA 0,0,BBEG
      STA 0,0,SP
      STA 0,0,TO
      LDA 1,0,+5
      STA 1,0,TO
      LDA 1,0,+4
      STA 1,0,FP
      JMP 0,+3
      A1
      X

```

```

      ;THE ADDRESS TO GO TO
      ;ON OVERFLOW IS A2.
      ;SET C(CAC0),STACK, AND
      ;FRAME POINTERS TO
      ;THE CONTENTS OF A1.
      ;TEST OF STACK OVERFLOW.
      X
      ;STACK LIMIT IS A1
      ;GREATER THEN STACK POINTER.
      ;OVERFLOW SHOULD OCCURE.
      X
      ;SETUP STACK IN PAGE 0
      ;TO UNDER FLOW ON ANY
      ;POP,POPJ,RTN, OR POPB.
      ;INSTRUCTION, THE FAULT
      ;LOCATION IS A1.
      X
      ;TEST OVERFLOW OF THE STACK
      ;ON THE CAP INSTRUCTION.
      ;IF NO OV GO TO A2.
      ;IF OV GO TO A1.
      X

```

```

10010 ECL20
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

```

```

      VECTOR
      JSR 0,VEC
      VP
      A2
      A1
      0
      X
      OBTILL
      INC 2,3
      LDA 1,BEND
      LDA 0,A1
      STA 0,0,2
      BLM
      X
      AGAIN
      ISZ TEM
      LDA 1,BEND
      LDA 0,0,110
      SUB 0,1
      LDA 0,TEM
      ADCW 0,1,SZH
      JAP A1
      X
      FIND
      LDA 2,0,20
      SUBZL 3,3
      INCZL 3,3,S4C
      JAP A1
      LDA 2,0,2
      STA 2,0,0
      MOVZL# 2,2,SZC
      JMP 0,+5
      LDA 3,0,2
      LDA 0,20
      LDA 1,0,20
      STA 1,20
      LDA 1,0,20
      MOVZL 1,1
      MOVZL 1,1
      SUB# 1,2, SZR
      SUB# 1,3, SNR
      JMP A1
      LDA 1,20
      SUB# 0,1, SZR
      JMP 0,+10
      X

```

```

      ;DISPATCH TABLE AT LOCATION
      ;VCYAB,0,2 ENTRY 0 SET
      ;TO ADDRESS A1. OTHERS
      ;TO ADDRESS A2.
      X
      ;FILL THE BUFFER WITH
      ;A1.
      X
      ;ADVANCE TO NEXT BUFFER
      ;LOCATION AND TEST FOR
      ;END OF BUFFER-110. IF NOT
      ;END OF BUFFER GO TO A1.
      X
      ;FIND THE FINAL ADDRESS
      ;AND VALUE IN THE INDIRECT
      ;CHAIN. C(CAC2) FIRST
      ;ADDRESS AT WHICH TO
      ;START LOOKING.
      ;IF INDIRECT CHAIN IS
      ;P>15, EXIT TO A1.
      X
      ;AUTO INDEX REGISTERS
      ;ARE USED AS VARIABLES.
      X
      ;MAKE SURE THAT FINAL
      ;DATA WILL NOT POINT
      ;TO ANY SPOT IN THE CHAIN.
      ;IF IT DOES, EXIT TO A1.
      X
      ;C(CAC2)=FINAL ADDRESS
      ;C(CAC3)=DATA AT THAT
      ;ADDRESS.
      X

```

```

10011 ECL20
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16

```

\*\*\*\*\*LITERAL DEFINITIONS\*\*\*\*\*

```

SP#40
PF#41
SL#42
SF#43
TD#44
PF#45
SC#2
PF#3
ISPR#4
MK#5
ISL#6
ISF#7

```

STACK POINTER  
FRAME POINTER  
STACK LIMIT  
STACK FAULT  
TRAP ORIGIN  
FLOATING POINT FAULT

```

10012 ECL20
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58

```

\*\*\*\*\* DIAGNOSTIC PROGRAM PREAMBLE \*\*\*\*\*

```

      .LOC 0
      .DIRT 0
      .LOC 40
      .LOC 0
      .LOC 0
      .**1
      HALT
      .LOC 45
      EGGS
      .ZREL
      .BLK 8.
      19 00000-00010
      21 00010-005437 .EGGS EGGS
      23 00011-000000 MENTOP: 0
      24 00012-000000 ICAT: 0
      25
      26
      27
      28 000170
      29 00170 102441 OFF:
      30 00171 102000 ON:
      31 00172 142470
      32
      33 00174 002175
      34 00175 000533
      35
      36
      37
      38 000176
      39 00176 000002
      40
      41
      42
      43
      44
      45
      46
      47
      48
      49
      50
      51
      52
      53
      54
      55
      56
      57
      58

```

PERMANENT POINTER TO EGGS  
TOP OF MEMORY FROM SIZE  
START LOCATION OF CAT  
SPACE RESERVED FOR CAT/NO CAT RESTANT ENTRIES  
LOC 170  
SUBO 0,0,SKP  
ADC P,0  
ESTA 0,CATSW  
JMP 0,+1  
SAMESS  
SPACE RESERVED FOR SPECIAL RESTART ENTRIES  
LOC 176  
BLK 2  
LOCATIONS 200 = 213 RESERVED FOR ECLIPSE TESTS  
LOC 200  
JMP 0,BGNADR  
DTOSR: 0  
ITRET: 0  
BGNADR: NSTRY  
PASS: 0  
PASS1: 1  
PASSV: 1  
ITR: 0  
ITR1: 0  
ITR2: 0  
ITR3: 0  
ITR4: 0  
ITR5: 0  
ITR6: 0  
ITR7: 0

STACK CONTROL LOCATIONS  
POINTER TO DIRT BLOCK  
POINTER TO EGGS BLOCK  
LOCATIONS RESERVED  
FOR DEBUG BREAKPOINTS  
POINTER TO EGGS BLOCK  
EGGS  
ZREL  
BLK 8.  
LOCATIONS RESERVED  
FOR DEBUG BREAKPOINTS  
PERMANENT POINTER TO EGGS  
TOP OF MEMORY FROM SIZE  
START LOCATION OF CAT  
SPACE RESERVED FOR CAT/NO CAT RESTANT ENTRIES  
LOC 170  
SUBO 0,0,SKP  
ADC P,0  
ESTA 0,CATSW  
JMP 0,+1  
SAMESS  
SPACE RESERVED FOR SPECIAL RESTART ENTRIES  
LOC 176  
BLK 2  
LOCATIONS 200 = 213 RESERVED FOR ECLIPSE TESTS  
LOC 200  
JMP 0,BGNADR  
DTOSR: 0  
ITRET: 0  
BGNADR: NSTRY  
PASS: 0  
PASS1: 1  
PASSV: 1  
ITERATION VALUE FOR THIS TEST  
ITERATION COUNTER  
ERROR SWITCH  
ERROR COUNTER  
ERROR RETURN  
LAST PLACE LOOP EXECUTED

10013 ECL20

```

01
02
03
04 00214 000045 MINLOC: PRGENU+100
05 00215 007545 MAXLOC: PRGENU+1000
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59

```

\*\*\*\*\*LOCAL ZREL\*\*\*\*\*  
/ BOTTOM OF TEST BUFFER  
/ TOP OF TEST BUFFER  
/ POINTERS TO SUBROUTINES  
/ PRINT ROUTINES: UCTAL  
/                    DECIMAL  
/                    MESSAGE  
/ STACK INITIALIZER  
/ TRAP INITIALIZER  
/ RANDOM # GENERATOR  
/ MEN. SIZING ROUTINE  
/ INIT ROUTINE  
/ DATA BUFFER START  
/ DATA BUFFER END  
/ TEST LOOP INITIALIZER  
/ TEST LOOP TERMINATOR  
/ SHORT ERROR REPORT  
/ LONG ERROR REPORT  
/ RANDOM # GENERATOR  
/ RANDOM # GENERATOR  
/ TEMPORARY STORAGE FOR  
/ MACHINE STATE IN LOOP,  
/ ERROR AND INIT SUBROUTINES  
/ TEMPORARYS FOR  
/ PRINT ROUTINES  
/ CONSTANT 0

10014 ECL20

```

01
02 00272 177777 M1
03 00273 000000 TEM1
04 00274 000000 TEM2
05 00275 000000 TEM3
06
07 00276 000000 RAN
08 00277 000000 SHAN
09
10
11 00300 000000 ORIG0
12 00301 000000 ORIG1
13 00302 000000 ORIG2
14 00303 000000 ORIG3
15 00304 000000 OK0
16 00305 000000 OK1
17 00306 000000 OK2
18 00307 000000 OK3
19 00310 000000 PHULR
20 00311 000000 MORE1
21 00312 000000 PDIVR
22
23
24 00313 000000 BANSIZ
25 00314 000000 BANT1
26 00315 000000 BANT0
27 00316 000000 PROM
28 00317 000000 TOD
29 00320 000000 BAC0
30 00321 000000 BAC1
31 00322 000000 BAC2
32 00323 000000 BAC3
33
34 00324 000445 UPI
35 00325 005400 JSR3
36 00326 005500 JSR2
37 00327 000010 PRK

```

RANDOM #  
/ SAVED RANDOM #  
/ VARIABLES IN MUL/DIV TEST  
/ ORIGINAL #  
/ CORRECT NUMBER  
/ 8AM TEST VARIABLES  
VCTAB 0.3  
JSR 0.2  
JSR 0.1  
PRK 0.

14015 ELL20  
01 000500  
02 000501  
03 000502  
04 000503  
05 000504  
06 000505  
07 000506  
08 000507  
09 000508  
10 000509  
11 000510  
12 000511  
13 000512  
14 000513  
15 000514  
16 000515  
17 000516  
18 000517  
19 000518  
20 000519  
21 000520  
22 000521  
23 000522  
24 000523  
25 000524  
26 000525  
27 000526  
28 000527  
29 000528  
30 000529  
31 000530  
32 000531  
33 000532  
34

.LDC 500  
J \*\*\*\*\*SIZE SYSTEM & RESERVE MEMORY\*\*\*\*\*  
JNSTRT: JSR  
NIG  
COM  
STA  
ELDA  
MOV  
JMP  
ELOA  
LOA  
ADU  
AUCZ#  
JMP  
SUB  
LOA  
ADU  
STA  
JMP  
NOCAT  
1,2  
1,400  
2,1  
1,CAT  
2,MAXLOC  
SMRES  
LOA  
SUB  
STA  
ESTA  
STA  
2,MAXLOC

1,0  
1,0  
0,2,8NC  
NUCAT  
0,9,SNR  
NOCAT  
0,PRGEND  
1,81777  
1,0  
0,2,8NC  
NUCAT  
1,2  
1,400  
2,1  
1,CAT  
2,MAXLOC  
SMRES  
1,200  
1,2  
1,1  
1,CATSM  
2,MAXLOC

1,0,520 OFF THE  
1,0,520  
1,1  
1,CATSM  
2,MAXLOC

1,0,520 OFF THE  
1,1  
1,CATSM  
2,MAXLOC

1,0,520 OFF THE  
1,1  
1,CATSM  
2,MAXLOC

1,0,520 OFF THE  
1,1  
1,CATSM  
2,MAXLOC

1,0,520 OFF THE  
1,1  
1,CATSM  
2,MAXLOC

14016 ECL20  
01 000533  
02 000534  
03 000535  
04 000536  
05 000537  
06 000540  
07 000541  
08 000542  
09 000543  
10 000544  
11 000545  
12 000547  
13 000548  
14 000549  
15 000550  
16 000551  
17 000552  
18 000553  
19 000554  
20 000555  
21 000556  
22 000557  
23 000558  
24 000559  
25 000560  
26 000561

SMRES1 JSR  
RESIZ  
LOA  
MOVE  
JSR  
JSR  
KCR LF  
SUB  
STA  
ELDA  
MOV  
JMP  
JSR  
SETSM  
JSR  
KCR LF  
HALT  
JMP  
READ9  
ESTA  
JMP

0,SMRES  
1,HEMTOP  
0,0  
0,IPDCT  
0,SMRES  
1,1  
1,PASS  
1,AUTO  
1,1,92R  
START  
0,SMRES  
0,SMRES  
+1  
0  
0,SMREG  
START  
0,SMRES

PRINT SIZE OF MEMORY  
/HEMTOP  
/IPDCT  
/SMRES  
/RESET PASS COUNT  
/RUNNING IN AUTO MODE?  
/YES START PROGRAM.  
/NO, PRINT SET SWITCHES MESS.  
/READ NEW STATE OF SWITCHES

\*\*\*\*\*OUTPUT SIRT MESSAGE & READ SWITCHES\*\*\*\*\*  
/PRINT SIZE OF MEMORY  
/RESET PASS COUNT  
/RUNNING IN AUTO MODE?  
/YES START PROGRAM.  
/NO, PRINT SET SWITCHES MESS.  
/READ NEW STATE OF SWITCHES

```

10017 ECL20
01 00563 102400 BEG11 SUB 0,0
02 00564 040210 STA 0,ITRER
03
04
05 JSR #ENTRA
06 MOVS 0,1
07 LDA 2,MAXLOC
08 SUBZ 2,0,SZC
09 JMP #1
10 00571 008777 ADD 2,0
11 00572 143000 SUBZ 2,1,SZC
12 00573 146422 JMP #1
13 00574 008777 ADD 2,1
14 00575 147000
15 00576 030214 BEG21 LDA 2,MINLOC
16 00577 142433 SUBZ# 2,0,SNC
17 00578 000703 JMP BEG1
18 00579 146433 SUBZ# 2,1,SNC
19 00580 000701 JMP BEG1
20 00581 106433 SUBZ# 0,1,SNC
21 00582 106433 XCH 0,1
22 00583 104710
23 00584 104710
24 00585 131000 BEG31 MOV 1,2
25 00586 112400 LDA 3,#377
26 00587 034005" SUBZ# 3,2,SNC
27 00588 172433 JMP BEG1
28 00589 000702 STA 0,BBEG
29 00590 040227 STA 1,BEND
30 00591 002401 JMP #+1
31 00592 002014 BEGIN
32
33
34
35
36
37
38
39
40
41
42
10018 ECL20
01
02
03
04 00616 023401 TMP1 LDA 0,#1,3
05 00617 040040 STA 0,SP
06 00620 040041 STA 0,FP
07 00621 020227 LDA 0,BEND
08 00622 024064" LDA 1,#140
09 00623 122400 SUB 1,0
10 00624 040042 STA 0,SL
11 00625 021406 LDA 0,#73
12 00626 040043 STA 0,FP
13 00627 040045 STA 0,FF
14 00630 033401 LDA 2,#1,3
15 00631 024063" LDA 1,#10
16 00632 133000 ADD 1,2
17 00633 050044 STA 2,TO
18 00634 024062" LDA 1,#-32.
19 00635 041000 STA 0,0,2
20 00636 151400 TNC 2,2
21 00637 123404 INC 1,1,SZR
22 00640 006775 JMP #3
23 00641 023402 LDA 1,2,3
24 00642 133000 ADD 1,2
25 00643 025403 LDA 1,3,3
26 00644 045340 STA 1,-32.12
27 00645 001404 JMP 4,3
28
29
30
31
32 00646 020226 STK1
33 00647 100110 LDA 0,BBEG
34 00650 040040 SBI 1,0
35 00651 040041 STA 0,SP
36 00652 040004 STA 0,FP
37 00653 020227 STA 0,BEND
38 00654 024061" LDA 1,#100
39 00655 122400 SUB 1,0
40 00656 040042 STA 0,SL
41 00657 040006 LDA 0,ISL
42 00660 021408 STA 0,#73
43 00661 040045 STA 0,FP
44 00662 040043 STA 0,SP
45 00663 040007 STA 0,ISF
46 00664 001401 JMP 1,3
? TRAPER FAULT, URIGIN, TRAP#, SUBROUTINE ADDRESS
?SET STACK POINTER
?AND FRAME POINTER
?END OF BUFFER=100
?IS THE STACK LIMIT
?STACK FAULT ADDRESS,
?INIT FAULT AND
?FLOATING FAULT
?TRAP ORIGIN ADDRESS
?SET ORIGIN
?PUT FAULT RETURN
?IN THE DISPATCH TABLE
?TRAP NUMBER.....
?SUBROUTINE ADDRESS
?SET A SINGLE SUBROUTINE
?ADDRESS AND EXIT
?STACK INITIALIZATION ROUTINE
?INITIALIZE A STACK,....
?MAKE STACK POINTER
?POINT TO THE DATA
?BUFFER -1.
?BUFFER END =100 IS
?THE STACK LIMIT
?SETUP STACK FAULT
?AND FLOATING FAULT

```

```

10019 ECL20
01
02
03
04 00665 054420 VEC:
05 00666 033400
06 00667 021401
07 00670 041000
08 00671 155400
09 00672 024001
10 00673 133710
11 00674 034411
12 00675 021402
13 00676 025403
14 00677 133000
15 00700 041300
16 00701 062677
17 00702 102000
18 00703 062077
19 00704 001404
20 00705 000000 VRET: 0

; VECTOR TABLE INITIALIZER
;
; SETUP A VECTOR TABLE.....
;CALL*1*ORIGIN POINTER
;CALL*2*ERRROR ADDRESS
;CALL*3*CORRECT ENTRY
;CALL*4*ENTRY NUMBER
;TABLE IS FILLED WITH ERR
;RETURNS**
;GOOD ENTRY
;POSITION (DEVICE CODE),
;THE GOOD ENTRY.
;MASK OTHER DEVICES.
;
LDA 3,VRET
LDA 2,0,3
LDA 0,1,3
STA 0,0,2
INC 2,3
LDA 1,100
BLM
LDA 3,VRET
LDA 0,2,3
LDA 1,3,3
ADD 1,2
STA 0,-100,2
TOBST
ACC 0,0
MSXO 0
JMP 4,3

```

```

10020 ECL20
01
02
03
04
05
06 00706 175400 INIT: INC 3,3
07 00707 054201 STA 3,ITRET
08 00710 040245 STA 0,AC0
09
10 00711 021777 LDA 0,-1,3
11 00712 040206 STA 0,IT0
12 00713 040207 STA 0,ITRCT
13
14 00714 176400 SUB 3,3
15 00715 054210 STA 3,ITRER
16 00716 054211 STA 3,ITREC
17
18 00717 034203 LDA 3,PASS
19 00720 175004 MOV 3,3,SZR
20 00721 000404 JMP INIT1
21
22 00722 176500 SUBZL 3,3
23 00723 054206 STA 3,ITR
24 00724 054207 STA 3,ITRCT
25
26 00725 020245 INIT1: LDA 0,AC0
27 00726 002201 JMP 0,ITRET

```

```

; *****TEST UTILITY SUBROUTINES*****
; SUBROUTINE TO INITIALIZE A TEST LOOP
;TEST LOOP INITIALIZER
;SAVE RETURN LOCATION
;SAVE CONTENTS OF AC0
;GET # OF ITERATIONS
;SET ITER. VALUE
;SET ITER. COUNT
;CLEAR ERROR SWITCH
;CLEAR ERROR COUNT
;TEST FOR FIRST PASS
;THIS IS 1ST PASS
;SET ITERATIONS FOR
;1 LOOP ONLY.
;RESTORE ACIS AND
;EXIT TO TEST

```



10021 ECL20

```

01
02
03
04 00727 054213 LOP: STA 3,LOPNET
05 00730 014207 DSZ ITRCT
06 00731 000440 JMP LOP3
07 00732 034210 LVA 3,ITRER
08 00733 175005 MOV 3,3,SNR
09 00734 002213 JMP #LOPNET
10 00735 034206 LVA 3,ITR
11 00736 054207 STA 3,ITRCT
12
13 00737 074477 LOP1: REAS 3
14 00740 175112 MOVL# 3,3,SZC
15 00741 000403 JMP *3
16 00742 136470 ELDA 3,3,SWREG
17 00744 005501 ADOL 3,3
18 00745 177103 ADOL 3,3,SNR
19 00746 000421 JMP LOP2
20 00747 000245 STA 0,ACR
21 00750 044248 STA 1,AC1
22 00751 050247 STA 2,AC2
23 00752 000220 JSR #INES
24 00753 001725 PERCENT 0,0
25 00754 102400 SUB 1,ITREC
26 00755 024211 LVA 1,ITREC
27 00756 040211 STA 0,ITREC
28 00757 030060 LVA 2,ITR0,
29 00760 140710 MUL 2,ITR
30 00761 036206 LVA 2,ITR
31 00762 153710 DIV 3,IPDEC
32 00763 000217 JSR #IPDEC
33 00764 020245 LVA 1,AC1
34 00765 024248 LVA 2,AC2
35 00766 030247 LVA 3,3
36 00767 176400 LOP2: SUB 3,ITREC
37 00770 054211 STA 3,ITREC
38
39 00771 034210 LOP3: LVA 3,ITRER
40 00772 175004 MOV 3,3,SR
41 00773 074477 REAS 3
42 00774 175112 MOVL# 3,3,SZC
43 00775 000403 JMP *3
44 00776 136470 ELOA 3,3,SWREG
45 00777 005445
46 01000 177113 ACDL#
47 01001 002201 JMP #ITRET
48 01002 002213 JMP #LOPNET
49

```

? SUBROUTINE TO TERMINATE A TEST LOOP

```

? LONG AND SHORT FORM ERROR ROUTINES
10022 ECL20
01
02
03
04 01003 054252 ERR1: STA 3,EPC
05 01004 040245 SUBCL 0,0
06 01005 102500 JAC0 & CARRY.
07 01006 040251 STA 0,CY
08
09 01007 021400 LVA 0,0,3
10 01010 175400 INC 3,3
11 01011 101004 MOV 0,0,SR
12 01012 000775 JMP 0,0
13 01013 000405 JMP ERR
14
15 01014 054252 ERR1: STA 3,EPC
16 01015 040245 SUBCL 0,0
17 01016 102500 JAC0 & CARRY
18 01017 040251 STA 0,CY
19
20 01020 054212 ERR: STA 3,ERRET
21 01021 010211 ITRER
22
23 01022 020210 LVA 0,ITRER
24 01023 110033 ADCZ# 0,3,SNR
25 01024 000464 JMP ERR4
26
27 01025 054210 STA 3,ITRER
28 01026 044246 STA 1,AC1
29 01027 050247 STA 2,AC2
30 01030 000220 JSR #INES
31 01031 001337 EMSG
32 01032 024203 LVA 1,PASS
33 01033 125420 INCZ 1,1
34 01034 000217 JSR #IPDEC
35 01035 000220 JSR #INES
36 01036 001732 MOVZ HEADEF
37 01037 101020 LDA 1,CY
38 01040 024251 LDA 1,CY
39 01041 000217 JSR #IPDEC
40 01042 101040 MOVO 0,0
41 01043 024245 LDA 1,AC0
42 01044 000216 JSR #IPUCT
43 01045 024246 LDA 1,AC1
44 01046 000216 JSR #IPUCT
45 01047 024247 LDA 1,AC2
46 01050 000216 JSR #IPUCT
47 01051 024250 LDA 1,AC3
48 01052 000216 JSR #IPUCT
49 01053 024252 LDA 1,EPC
50 01054 000216 JSR #IPUCT
51

```

? LONG AND SHORT FORM ERROR ROUTINES

```

? SUBROUTINE TO TERMINATE A TEST LOOP
10021 ECL20
01
02
03
04 00727 054213 LOP: STA 3,LOPNET
05 00730 014207 DSZ ITRCT
06 00731 000440 JMP LOP3
07 00732 034210 LVA 3,ITRER
08 00733 175005 MOV 3,3,SNR
09 00734 002213 JMP #LOPNET
10 00735 034206 LVA 3,ITR
11 00736 054207 STA 3,ITRCT
12
13 00737 074477 LOP1: REAS 3
14 00740 175112 MOVL# 3,3,SZC
15 00741 000403 JMP *3
16 00742 136470 ELDA 3,3,SWREG
17 00744 005501 ADOL 3,3
18 00745 177103 ADOL 3,3,SNR
19 00746 000421 JMP LOP2
20 00747 000245 STA 0,ACR
21 00750 044248 STA 1,AC1
22 00751 050247 STA 2,AC2
23 00752 000220 JSR #INES
24 00753 001725 PERCENT 0,0
25 00754 102400 SUB 1,ITREC
26 00755 024211 LVA 1,ITREC
27 00756 040211 STA 0,ITREC
28 00757 030060 LVA 2,ITR0,
29 00760 140710 MUL 2,ITR
30 00761 036206 LVA 2,ITR
31 00762 153710 DIV 3,IPDEC
32 00763 000217 JSR #IPDEC
33 00764 020245 LVA 1,AC1
34 00765 024248 LVA 2,AC2
35 00766 030247 LVA 3,3
36 00767 176400 LOP2: SUB 3,ITREC
37 00770 054211 STA 3,ITREC
38
39 00771 034210 LOP3: LVA 3,ITRER
40 00772 175004 MOV 3,3,SR
41 00773 074477 REAS 3
42 00774 175112 MOVL# 3,3,SZC
43 00775 000403 JMP *3
44 00776 136470 ELOA 3,3,SWREG
45 00777 005445
46 01000 177113 ACDL#
47 01001 002201 JMP #ITRET
48 01002 002213 JMP #LOPNET
49

```

? SUBROUTINE TO TERMINATE A TEST LOOP

```

? LONG AND SHORT FORM ERROR ROUTINES
10022 ECL20
01
02
03
04 01003 054252 ERR1: STA 3,EPC
05 01004 040245 SUBCL 0,0
06 01005 102500 JAC0 & CARRY.
07 01006 040251 STA 0,CY
08
09 01007 021400 LVA 0,0,3
10 01010 175400 INC 3,3
11 01011 101004 MOV 0,0,SR
12 01012 000775 JMP 0,0
13 01013 000405 JMP ERR
14
15 01014 054252 ERR1: STA 3,EPC
16 01015 040245 SUBCL 0,0
17 01016 102500 JAC0 & CARRY
18 01017 040251 STA 0,CY
19
20 01020 054212 ERR: STA 3,ERRET
21 01021 010211 ITRER
22
23 01022 020210 LVA 0,ITRER
24 01023 110033 ADCZ# 0,3,SNR
25 01024 000464 JMP ERR4
26
27 01025 054210 STA 3,ITRER
28 01026 044246 STA 1,AC1
29 01027 050247 STA 2,AC2
30 01030 000220 JSR #INES
31 01031 001337 EMSG
32 01032 024203 LVA 1,PASS
33 01033 125420 INCZ 1,1
34 01034 000217 JSR #IPDEC
35 01035 000220 JSR #INES
36 01036 001732 MOVZ HEADEF
37 01037 101020 LDA 1,CY
38 01040 024251 LDA 1,CY
39 01041 000217 JSR #IPDEC
40 01042 101040 MOVO 0,0
41 01043 024245 LDA 1,AC0
42 01044 000216 JSR #IPUCT
43 01045 024246 LDA 1,AC1
44 01046 000216 JSR #IPUCT
45 01047 024247 LDA 1,AC2
46 01050 000216 JSR #IPUCT
47 01051 024250 LDA 1,AC3
48 01052 000216 JSR #IPUCT
49 01053 024252 LDA 1,EPC
50 01054 000216 JSR #IPUCT
51

```

10023 ECL20

! ERROR ROUTINES (CONTINUED)

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
ERR1: LDA 2,EPC
      ISZ EPC
      LDA 0,EMRET
      ADDL 2,0,SNR
      JMP ERR3
ERR2: LDA 3,0,2
      MOV# 3,3,SNR
      JMP ERR2
      STA 3,0,2
      JSR 0,IMESH
      JMP ERR1
ERR3: LDA 1,0,3
      0,IPROCT
      JMP ERR1
ERR4: JSR 0,IMESH
      KCRFL
      LDA 1,AC1
      LDA 2,AC2
      ELDA 3,AUTO
      MOV 3,3,SNR
      JMP ERR4
      JORST
      LDA 3,ALGGS
      LDA 3,4,3
      JMP 0,3
ERR4: LDA 0,AC0
      LDA 3,AC3
      JMP 0,ERR1

```

10024 ECL20

! RANDOM NUMBER GENERATOR SUBROUTINES

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
      LDA 0,RAN
      JMP 0,RAND*4
      LDA 0,RAN
      LDA 1,ITRER
      MOV 1,1,5ZR
      MOV 0,1
      HXL 2,1
      ADD 0,1
      MOVZL 1,1
      MOVZL 1,1
      ADD 1,0
      LDA 1,0,35031
      ADD 1,0
      STA 0,RAN
      JMP 0,1,3
      STA 3,ORIG3
      RANO
      JSR 0,ENTRA
      INCS 0,1
      NEG 0,2
      STA 0,ORIG0
      ADD 1,2
      STA 1,ORIG1
      STA 2,ORIG2
      JMP 0,ORIG3
      LDA 2,MINLOC
      INC 2,2
      MOV# 2,2,SIZE
      JMP 0,6
      LDA 0,0,2
      STA 2,0,2
      LDA 1,0,2
      STA 0,0,2
      SUB# 1,2,5ZR
      JMP 0,3
      STA 2,MENTOP
      JMP 0,SIZE+1
      ! GENERATE A NEW RANDOM
      ! NUMBER IN C(AC0) AND
      ! C(RAN), IF C(ITRER)=0
      ! OTHERWISE LOAD C(AC0)
      ! WITH OLD #.
      ! SIZING SUBROUTINE! RETURNS SIZE OF LOGICAL MEM IN AC2.
      ! SIZE LOGICAL MEMORY
      ! MEM IS 32K WORDS.
      ! SAVE MEMORY TOP ADDRESS

```

0025 ECL20  
01 01247 000767  
02

0026 ECL20  
01 01247 000767  
02

JMP MESS1

```
*****PRINT ROUTINES*****
04 01160 175100 PDEC1  MOVL 3,3
05 01161 054255 STA 3,POERET
06 01162 175200 MOVR 3,3
07 01163 064441 JSR PDEC3
08 01164 023420 10000.
09 01165 001750 1000.
10 01166 000144 100.
11 01167 000012 10.
12 01170 000001 1.
13 01171 000000 0
14 01172 175100 POC1:  MOVL 3,3
15 01173 054255 STA 3,POERET
16 01174 175200 MOVR 3,3
17 01175 064427 JSR PDEC3
18 01176 100000 100000
19 01177 010000 10000
20 01200 001000 1000
21 01201 000100 100
22 01202 000010 10
23 01203 000001 1
24 01204 000000 0
25 01205 020056=PDEC1: LDA 0,=11
26 01206 031377 LDA 2,=1,2
27 01207 151015 MOV# 2,2,SNR
28 01210 000415 JMP PDEC3+1
29 01211 102460 SUBC 0,0
30 01212 162462 SUBO# 2,1,3ZC
31 01213 000404 JMP PDEC2
32 01214 140420 SUBZ 2,1
33 01215 101400 INC 0,0
34 01216 000774 JMP =4
35 01217 151234 PDEC2: MOVZR 2,2,SZR
36 01220 192462 SUBC 2,2,SZC
37 01221 030035= LDA 2,=0,0
38 01222 143004 ADD 2,0,SZR
39 01223 171401 INC 3,2,SKP
40 01224 171401 INC 3,2,SKP
41 01225 004423 JSR CHAR
42 01226 004423 JSR CHAR
43 01227 100756 MOVR 2,3,SZR
44 01228 000756 JMP PDEC1
45 01230 034255 LDA 3,POERET
46 01231 175200 MOVR 3,3
47 01232 001400 JMP 0,3
48
49 01233 175400 MESS:  INC 3,3
50 01234 054256 STA 3,MESSRET
51 01235 031777 LDA 2,=1,3
52 01236 020065=MESS1: LDA 0,=0,77
53 01237 025000 LDA 1,0,2
54 01240 151420 INCZ 2,2
55 01241 103400 AND 1,0
56 01242 106700 SUBS 0,1
57 01243 004405 JSR CHAR
58 01244 121005 MOV 1,0,SNR
59 01245 022256 JMP #MESSRET
60 01246 004402 JSR CHAR
```

```

10227 ECL20
01
02
03 01250 175100 CHAR:
04 01251 054253 STA 3,3
05 01252 050450 STA 2,CHR$
06
07 01253 074477 READS 3
08 01254 175112 MOVLM 3,3,SZC
09 01255 050453 JMP *3
10 01256 136470 ELDA 3,SMREG
11 005165
12 01260 030254= LDA 2,*22000
13 01261 173400 AND 3,2
14 01262 153120 ADDZL 2,2
15 01263 153265 ADDCR 2,2,SNR
16 01264 000435 JMP REST
17
18 01265 034055= LDA 3,*377
19 01266 11725 ANDZS 0,3,SNR
20 01267 000432 JMP REST
21
22 01270 103004 ADD 0,0,3ZR
23 01271 000777 JMP *-1
24 01272 177850 ADDCR 3,3
25 01273 020853= LDA 0,*211*400
26 01274 152445 SUBO 3,0,SNR
27 01275 000430 JMP CHAR4
28
29 01276 161340 MOVOS 3,0
30 01277 010254 CHAR1: ISZ CHORZ
31
32 01300 151135 MOVZL# 2,2,SNR
33 01301 000405 JMP CHAR2
34 01302 061117 DGAS 0,LPT
35 01303 063517 SKP8Z LPT
36 01304 000777 JMP *-1
37 01305 060217 NI0C LPT
38
39 01306 151133 CHAR2: MOVZL# 2,2,SNR
40 01307 000405 JMP CHAR3
41 01310 061111 DGAS 0,TTD
42 01311 063511 SKP8Z TTD
43 01312 000777 JMP *-1
44 01313 060211 NI0C TTD
45
46 01314 175403 CHAR3: INC 3,3,SNR
47 01315 000762 JMP CHAR1
48 01316 050602= LDA 2,*212
49 01317 142405 SUB 2,0,SNR
50 01320 040254 STA 0,CHORZ
51
52 01321 030411 REST: LDA 2,CHR$V
53 01322 034253 LDA 3,CHAR#
54 01323 175200 MOVR 3,3
55 01324 001400 JMP 0,3
56
57 01325 034254 CHAR4: LDA 3,CHORZ
58 01326 020051= LDA 0,*#-8
59 01327 114410 XOR 0,3
60 01330 020050= LDA 0,*#240

```

```

0028 ECL20
01 01331 000745 JMP
02
03 01332 000000 CHR$V: 0

```

CHAR1

JMP

TEMP SAVE FOR AC2

LPT/TTD INTERFACE ROUTINE: CHARACTER PASSED IN AC0

CHAR#

01

```

;SAVE RETURN ADDR.
;SAVE AC2
;READ SWITCHES INTO AC3
;TEST SWITCH 0
; SW0 SET
; SW0 CLEAR, DEFAULT
;MASK SW2 & SW5 INTO
;AC2 FROM AC3
;LEFT JUSTIFY SW2
;COMPLEMENT SW2
;NO OUTPUT, RETURN
;MASK CHARACTER INTO L,BYTE
;OF AC3, CLEAR CARRY.
;IF NULL CHAR, RETURN
;DETERMINE REQUIRED
;STATE OF PARITY BIT &
;INSERT IT
;TEST FOR TAB
;TRUE: SETUP TAB SIMULATION
;RESTORE CHARACTER TO R,BYTE AC0
;SET CARRY, BUMP LINE COUNT.
;SEND TO LPT?
;NOPE, MUST BE TTD
;O.K, FETCH CHARACTER
;WAIT FOR DONE
;CLEAR DEVICE
;SEND TO TTD?
;O.K, SEND CHARACTER
;WAIT FOR DONE
;CLEAR DEVICE
;IF TABBING, AND NOT
;FINISHED, LOOP.
;TEST FOR CR/LF
;TRUE: ZERO LINE COUNT
;OTHERWISE RETURN
;SET UP TO TAB
;C(AC3) IS TWOS COMPLEMENT
;OF # OF SPACES NEEDED

```

| LINE | ADDRESS | OPERATION | DATA   | STATUS | REMARKS |
|------|---------|-----------|--|--------|---------|
| 01   | 0038    | ECL20     |  |        |         |
| 02   | 01333   | 005215    | PARNES: ,TXTE I<15><12>PASS I                          |        |         |
| 03   | 040520  |           |  |        |         |
| 04   | 051523  |           |  |        |         |
| 05   | 002020  |           |  |        |         |
| 06   | 01337   | 005215    | ERRSGI ,TXTE I<15><12>ERROR IN PASS I                  |        |         |
| 07   | 151505  |           |  |        |         |
| 08   | 147722  |           |  |        |         |
| 09   | 120322  |           |  |        |         |
| 10   | 047311  |           |  |        |         |
| 11   | 050240  |           |  |        |         |
| 12   | 051501  |           |  |        |         |
| 13   | 035123  |           |  |        |         |
| 14   | 000240  |           |  |        |         |
| 15   | 01350   | 005215    | RIOH1: ,TXTE I<15><12>                                 |        |         |
| 16   | 01351   | 152102    | BYD ADDR GOOD BAD WORD <15><12> I                      |        |         |
| 17   | 004717  |           |  |        |         |
| 18   | 042101  |           |  |        |         |
| 19   | 151104  |           |  |        |         |
| 20   | 043411  |           |  |        |         |
| 21   | 147717  |           |  |        |         |
| 22   | 004504  |           |  |        |         |
| 23   | 040502  |           |  |        |         |
| 24   | 024504  |           |  |        |         |
| 25   | 147727  |           |  |        |         |
| 26   | 042322  |           |  |        |         |
| 27   | 106640  |           |  |        |         |
| 28   | 000012  |           |  |        |         |
| 29   | 01366   | 005215    | BT2H1: ,TXTE I<15><12>                                 |        |         |
| 30   | 01367   | 152102    | BT2 ADDR GOOD BAD WORD <15><12> I                      |        |         |
| 31   | 004032  |           |  |        |         |
| 32   | 042101  |           |  |        |         |
| 33   | 151104  |           |  |        |         |
| 34   | 043411  |           |  |        |         |
| 35   | 147717  |           |  |        |         |
| 36   | 004504  |           |  |        |         |
| 37   | 040502  |           |  |        |         |
| 38   | 004504  |           |  |        |         |
| 39   | 147727  |           |  |        |         |
| 40   | 042322  |           |  |        |         |
| 41   | 106640  |           |  |        |         |
| 42   | 000012  |           |  |        |         |
| 43   | 01404   | 005215    | BANH1: ,TXTE I<15><12>AC15 AFTER BAN INSTRUCTION WRONG |        |         |
| 44   | 141501  |           |  |        |         |
| 45   | 051447  |           |  |        |         |
| 46   | 040640  |           |  |        |         |
| 47   | 152300  |           |  |        |         |
| 48   | 151305  |           |  |        |         |
| 49   | 041240  |           |  |        |         |
| 50   | 040501  |           |  |        |         |
| 51   | 144640  |           |  |        |         |
| 52   | 051516  |           |  |        |         |
| 53   | 151324  |           |  |        |         |
| 54   | 141025  |           |  |        |         |
| 55   | 144724  |           |  |        |         |
| 56   | 047317  |           |  |        |         |
| 57   | 153640  |           |  |        |         |
| 58   | 147722  |           |  |        |         |
| 59   | 040516  |           |  |        |         |
| 60   | 01420   | 005215    | <15><12>ORIG C(AC0-3) I                                |        |         |

| LINE | ADDRESS | OPERATION | DATA  | STATUS | REMARKS |
|------|---------|-----------|---|--------|---------|
| 01   | 151317  |           |   |        |         |
| 02   | 043711  |           |   |        |         |
| 03   | 141040  |           |   |        |         |
| 04   | 040450  |           |   |        |         |
| 05   | 030303  |           |   |        |         |
| 06   | 031455  |           |   |        |         |
| 07   | 120251  |           |   |        |         |
| 08   | 000000  |           |   |        |         |
| 09   | 01436   | 005215    | BANH2: ,TXTE I<15><12>GOOD C(AC0-3) I       |        |         |
| 10   | 147507  |           |   |        |         |
| 11   | 042317  |           |   |        |         |
| 12   | 141040  |           |   |        |         |
| 13   | 040450  |           |   |        |         |
| 14   | 030303  |           |   |        |         |
| 15   | 031455  |           |   |        |         |
| 16   | 120251  |           |   |        |         |
| 17   | 000000  |           |   |        |         |
| 18   | 01447   | 005215    | BANH3: ,TXTE I<15><12>BAD C(AC0-3) I        |        |         |
| 19   | 040502  |           |   |        |         |
| 20   | 120104  |           |   |        |         |
| 21   | 024303  |           |   |        |         |
| 22   | 141501  |           |   |        |         |
| 23   | 025400  |           |   |        |         |
| 24   | 124403  |           |   |        |         |
| 25   | 000240  |           |   |        |         |
| 26   | 01457   | 005215    | BANH4: ,TXTE I<15><12>RBANM SET C(CARRY) I  |        |         |
| 27   | 041042  |           |   |        |         |
| 28   | 040501  |           |   |        |         |
| 29   | 120042  |           |   |        |         |
| 30   | 142523  |           |   |        |         |
| 31   | 120324  |           |   |        |         |
| 32   | 024303  |           |   |        |         |
| 33   | 040703  |           |   |        |         |
| 34   | 151322  |           |   |        |         |
| 35   | 124531  |           |   |        |         |
| 36   | 000000  |           |   |        |         |
| 37   | 01472   | 005215    | BANH5: ,TXTE I<15><12>                      |        |         |
| 38   | 01473   | 151306    | FROM TO MURD ORIG C(AC0) GOOD BAD<15><12> I |        |         |
| 39   | 040717  |           |   |        |         |
| 40   | 152011  |           |   |        |         |
| 41   | 004717  |           |   |        |         |
| 42   | 147727  |           |   |        |         |
| 43   | 042322  |           |   |        |         |
| 44   | 147411  |           |   |        |         |
| 45   | 144722  |           |   |        |         |
| 46   | 004507  |           |   |        |         |
| 47   | 024303  |           |   |        |         |
| 48   | 141501  |           |   |        |         |
| 49   | 124400  |           |   |        |         |
| 50   | 043411  |           |   |        |         |
| 51   | 147717  |           |   |        |         |
| 52   | 004504  |           |   |        |         |
| 53   | 040502  |           |   |        |         |
| 54   | 105504  |           |   |        |         |
| 55   | 020012  |           |   |        |         |
| 56   | 01515   | 005215    | EDIVH: ,TXTE I<15><12>                      |        |         |
| 57   | 01516   | 144504    | DIVX AC0 AC1 AC2<15><12>                    |        |         |
| 58   | 154126  |           |   |        |         |
| 59   | 040411  |           |   |        |         |
| 60   | 030303  |           |   |        |         |

| Line | Text  | Code | Text | Code |
|------|---|------|------|------|
| 01   | 0031 ECL20  |      |      |      |
| 02   | 044411  |      |      |      |
| 03   | 130703  |      |      |      |
| 04   | 040411  |      |      |      |
| 05   | 131303  |      |      |      |
| 06   | 005215  |      |      |      |
| 07   | 01527 151317 ORIG I   |      |      |      |
| 08   | 043711  |      |      |      |
| 09   | 000011  |      |      |      |
| 10   | 09 01532 005215 BAHM7: .TXTE I<15><12>SHOULD BE ZERO, C(AC0=3)= I |      |      |      |
| 11   | 044123  |      |      |      |
| 12   | 052717  |      |      |      |
| 13   | 042314  |      |      |      |
| 14   | 041240  |      |      |      |
| 15   | 126305  |      |      |      |
| 16   | 142502  |      |      |      |
| 17   | 147782  |      |      |      |
| 18   | 120254  |      |      |      |
| 19   | 024303  |      |      |      |
| 20   | 141501  |      |      |      |
| 21   | 026466  |      |      |      |
| 22   | 124463  |      |      |      |
| 23   | 120275  |      |      |      |
| 24   | 000000  |      |      |      |
| 25   | 24 01551 005215 BAHM8: .TXTE I<15><12> GOOD BAD                   |      |      |      |
| 26   | 01052 042101 ADDR GOOD BAD  |      |      |      |
| 27   | 131104  |      |      |      |
| 28   | 043411  |      |      |      |
| 29   | 147717  |      |      |      |
| 30   | 004504  |      |      |      |
| 31   | 040502  |      |      |      |
| 32   | 004524  |      |      |      |
| 33   | 040502  |      |      |      |
| 34   | 004515  |      |      |      |
| 35   | 040706  |      |      |      |
| 36   | 145311  |      |      |      |
| 37   | 042305  |      |      |      |
| 38   | 106640  |      |      |      |
| 39   | 000012  |      |      |      |
| 40   | 01570 005215 MULM1: .TXTE I<15><12> AC0 AC1 AC2<15><12>           |      |      |      |
| 41   | 01571 052515 MUL  |      |      |      |
| 42   | 004714  |      |      |      |
| 43   | 141501  |      |      |      |
| 44   | 004460  |      |      |      |
| 45   | 040601  |      |      |      |
| 46   | 141501  |      |      |      |
| 47   | 106662  |      |      |      |
| 48   | 01601 147412 ORIG I   |      |      |      |
| 49   | 144722  |      |      |      |
| 50   | 004507  |      |      |      |
| 51   | 000000  |      |      |      |
| 52   | 01609 005215 MULM2: .TXTE I<15><12>BAD I                          |      |      |      |
| 53   | 043502  |      |      |      |
| 54   | 045504  |      |      |      |
| 55   | 000000  |      |      |      |
| 56   | 01611 005215 MULM3: .TXTE I<15><12>GOOD I                         |      |      |      |
| 57   | 147507  |      |      |      |
| 58   | 042317  |      |      |      |
| 59   | 000011  |      |      |      |
| 60   | 01615 005215 DIVH1: .TXTE I<15><12>DIV AC0 AC1 AC2<15><12>        |      |      |      |
| 01   | 0032 ECL20  |      |      |      |
| 02   | 144504  |      |      |      |
| 03   | 004526  |      |      |      |
| 04   | 141501  |      |      |      |
| 05   | 004460  |      |      |      |
| 06   | 141501  |      |      |      |
| 07   | 004661  |      |      |      |
| 08   | 141501  |      |      |      |
| 09   | 006662  |      |      |      |
| 10   | 09 01626 147412 ORIG I  |      |      |      |
| 11   | 144722  |      |      |      |
| 12   | 000000  |      |      |      |
| 13   | 12 01632 005215 MOH4: .TXTE I<15><12>DIV/MUL AC0 AC1 AC2          |      |      |      |
| 14   | 144504  |      |      |      |
| 15   | 127526  |      |      |      |
| 16   | 052515  |      |      |      |
| 17   | 004714  |      |      |      |
| 18   | 141501  |      |      |      |
| 19   | 004460  |      |      |      |
| 20   | 141501  |      |      |      |
| 21   | 004661  |      |      |      |
| 22   | 141501  |      |      |      |
| 23   | 23 01644 106662 <15><12>ORIG I                                    |      |      |      |
| 24   | 147412  |      |      |      |
| 25   | 144722  |      |      |      |
| 26   | 004507  |      |      |      |
| 27   | 000000  |      |      |      |
| 28   | 28 01651 005215 SOVH1: .TXTE I<15><12>SDIV AC0 AC1 AC2            |      |      |      |
| 29   | 042123  |      |      |      |
| 30   | 053311  |      |      |      |
| 31   | 040411  |      |      |      |
| 32   | 030303  |      |      |      |
| 33   | 040411  |      |      |      |
| 34   | 130703  |      |      |      |
| 35   | 040411  |      |      |      |
| 36   | 131303  |      |      |      |
| 37   | 01662 005215 <15><12>ORIG I                                       |      |      |      |
| 38   | 151317  |      |      |      |
| 39   | 043711  |      |      |      |
| 40   | 000011  |      |      |      |
| 41   | 01666 005215 SHH1: .TXTE I<15><12>SHUL AC0 AC1 AC2                |      |      |      |
| 42   | 046523  |      |      |      |
| 43   | 146125  |      |      |      |
| 44   | 040411  |      |      |      |
| 45   | 030303  |      |      |      |
| 46   | 040411  |      |      |      |
| 47   | 130703  |      |      |      |
| 48   | 040411  |      |      |      |
| 49   | 131303  |      |      |      |
| 50   | 01677 005215 <15><12>ORIG I                                       |      |      |      |
| 51   | 151317  |      |      |      |
| 52   | 043711  |      |      |      |
| 53   | 000011  |      |      |      |
| 54   | 01703 005215 BLMH5: .TXTE I<15><12> WORD GOOD BAD BLM FAILED I    |      |      |      |
| 55   | 01704 151306 FROM TO  |      |      |      |
| 56   | 046317  |      |      |      |
| 57   | 152011  |      |      |      |
| 58   | 004717  |      |      |      |
| 59   | 147727  |      |      |      |
| 60   | 042322  |      |      |      |

0034 ECL20  
 01 051523  
 02 141640  
 03 047317  
 04 144724  
 05 052516  
 06 000305

0033 ECL20  
 01 049411  
 02 147717  
 03 064564  
 04 040502  
 05 084504  
 06 146102  
 07 120115  
 08 040706  
 09 146311  
 10 042305  
 11 000240  
 12 01726 005215 PERCENT .TXTE I<15><12>X FAIL=I  
 13 120645  
 14 040700  
 15 146311  
 16 000275  
 17 01732 005215 HEADERS .TXTE I<15><12>I>B><12>  
 18 005215  
 19 01734 151303 CRY AC0 AC1 AC2 AC3 LISTING <15><12>I  
 20 004531  
 21 141501  
 22 004460  
 23 141501  
 24 004601  
 25 141501  
 26 004662  
 27 141501  
 28 004463  
 29 144714  
 30 152123  
 31 047311  
 32 004507  
 33 005215  
 34 000000  
 35 01754 005215 MERIZI .TXTE I<15><12>LAST LOGICAL ADDRESS=I  
 36 040714  
 37 152123  
 38 146240  
 39 043717  
 40 141711  
 41 146101  
 42 000640  
 43 042104  
 44 142722  
 45 051523  
 46 000275  
 47 01770 005215 KGRUFI .TXTE I<15><12>I  
 48 000000  
 49 01772 142523 SETSWI .TXTE ISET DATA SWITCHS AND PRESS CONTINUEI  
 50 120324  
 51 040504  
 52 040724  
 53 031640  
 54 144727  
 55 141724  
 56 051510  
 57 040640  
 58 042116  
 59 050240  
 60 142722

|             |                 |
|-------------|-----------------|
| 10035 ECL20 | 10036 ECL20     |
| 01 BEGIN:   | DIV01           |
| 02          | 02 02046 006230 |
| 03          | 03 02047 020000 |
| 04          | 04 02048 006223 |
| 05          | 05 02049 006223 |
| 06          | 06 02050 006223 |
| 07          | 07 02051 004525 |
| 08          | 08 02052 020300 |
| 09          | 09 02053 024301 |
| 10          | 10 02054 030302 |
| 11          | 11 02055 153710 |
| 12          | 12 02056 004525 |
| 13          | 13 02057 006420 |
| 14          | 14 02062 101615 |
| 15          | 15 02063 000300 |
| 16          | 16 02064 000301 |
| 17          | 17 02065 000302 |
| 18          | 18 02066 101605 |
| 19          | 19 02067 000245 |
| 20          | 20 02070 000246 |
| 21          | 21 02071 000247 |
| 22          | 22 02072 101611 |
| 23          | 23 02073 000304 |
| 24          | 24 02074 000305 |
| 25          | 25 02075 000306 |
| 26          | 26 02076 000000 |
| 27          | 27 02077 006231 |
| 28          | 28 02078 000000 |
| 29          | 29 02079 000000 |
| 30          | 30 02080 000000 |
| 31          | 31 02081 000000 |
| 32          | 32 02082 000000 |
| 33          | 33 02083 000000 |
| 34          | 34 02084 006231 |

|       |       |
|-------|-------|
| MUL01 | MUL11 |
| 01    | 01    |
| 02    | 02    |
| 03    | 03    |
| 04    | 04    |
| 05    | 05    |
| 06    | 06    |
| 07    | 07    |
| 08    | 08    |
| 09    | 09    |
| 10    | 10    |
| 11    | 11    |
| 12    | 12    |
| 13    | 13    |
| 14    | 14    |
| 15    | 15    |
| 16    | 16    |
| 17    | 17    |
| 18    | 18    |
| 19    | 19    |
| 20    | 20    |
| 21    | 21    |
| 22    | 22    |
| 23    | 23    |
| 24    | 24    |
| 25    | 25    |
| 26    | 26    |
| 27    | 27    |
| 28    | 28    |
| 29    | 29    |
| 30    | 30    |
| 31    | 31    |
| 32    | 32    |
| 33    | 33    |
| 34    | 34    |

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



```

10037 ECL20
01
02
03
04 02100 006230 MD01 SETUP 2000R
05 02101 020000 MD01 JSR #ENTRIN
06
07 02102 006234 RAND
08 02103 105120 MOVL 0,1
09 02104 127100 ADDL 1,1
10 02105 107300 AODS 0,1
11 02106 110705 NEGS 0,2,SNC
12 02107 000773 JMP MD1
13 02110 142532 SUB# 2,0,SZC
14 02111 110710 XCH 0,2
15 02112 040300 STA 0,OKI00
16 02113 044301 STA 1,OKI01
17 02114 050302 STA 2,OKI02
18 02115 153710 DIV
19 02116 143710 MDR1 MUL
20 02117 034300 LOA 3,OKI00
21 02120 116414 SUB# 0,3,SZK
22 02121 000411 JMP MD4
23 02122 034301 LDA 3,OKI01
24 02123 136414 JMP MD4
25 02124 000406 SUB# 1,3,SZK
26 02125 034302 LOA 3,OKI02
27 02126 156414 SUB# 2,3,SZK
28 02127 000403 JMP MD4
29
30 02130 006231 LOOP
31 02131 002415 JSR #ENTLO
32 02132 002415 JMP #MD55
33
34 02134 101532 #MD4
35 02135 000300 OKI00
36 02136 000301 OKI01
37 02137 000302 OKI02
38 02140 101605 #MD4
39 02141 000245 AC1
40 02142 000246 AC2
41 02143 000247 0
42 02145 000763 JMP MD3
43
44 02146 002335 MD51
45
46
47
48
49
50
51
52
53
10038 ECL20
01
02
03 02147 054310 PHUL1
04 02150 034047-
05 02151 125303
06 02152 101201
07 02153 143200
08 02154 175404 INC 3,3,SZK
09 02155 000774 JMP #4
10 02156 125300 MOVR 1,1
11 02157 040304 STA 0,OK0
12 02158 044305 STA 1,OK1
13 02161 050306 STA 2,OK2
14 02162 002310 JMP #PHULR
15
16 02163 054311 MDCK1
17 02164 034304
18 02165 162414 SUB# 3,0,SZK
19 02166 000406 JMP MDCK1
20 02167 034305 LOA 3,OK1
21 02170 166414 SUB# 3,1,SZK
22 02171 000403 JMP MDCK1
23 02172 034306 LOA 3,OK2
24 02173 172414 SUB# 3,2,SZK
25 02174 010311 MDCK1
26 02175 002311 JMP #MDRET
27
28 02176 054312 PDIV1
29 02177 142432 SUB# 2,0,SZC
30 02200 000411 JMP PUIV2
31 02201 034047- LOA 3,0,SZC
32 02202 125120 MOVL 1,1
33 02203 101100 PDIV1
34 02204 142412 SUB# 2,0
35 02205 142400 MOVL 1,1
36 02206 125100 INC 3,3,SZK
37 02207 175404 JMP PDIV1
38 02210 000773 STA 0,OK0
39 02211 040304 STA 1,OK1
40 02212 044305 STA 2,OK2
41 02213 050306 JMP PDIVR
42 02214 002312
43
44 02215 054311 MDCK1
45 02216 034271
46 02217 101003
47 02220 175400 INC 3,3
48 02221 175213 MOVR 3,3,SNC
49 02222 000752 JMP MDCK1
50 02223 101003 MOV 0,0,SNC
51 02224 000740 JMP MDCK+1
52 02225 002311 JMP #MDRET
53
SOFTWARE UNSIGNED MULTIPLY
FC(AC1)=C(AC2)*C(AC0)
FRESULT TO C(AC0)=1
STA 3,PHULR
LDA 3,0,SZC
MOVR 0,0,SZC
ADDR 2,0
INC 3,3,SZK
JMP #4
MOVR 1,1
STA 0,OK0
STA 1,OK1
STA 2,OK2
JMP #PHULR
STA 3,MDRET
LDA 3,OK0
SUB# 3,0,SZK
JMP MDCK1
LOA 3,OK1
SUB# 3,1,SZK
JMP MDCK1
LOA 3,OK2
SUB# 3,2,SZK
ISZ MDRET
JMP #MDRET
STA 3,PDIVR
SUB# 2,0,SZC
JMP PUIV2
LOA 3,0,SZC
MOVL 1,1
MOVL 1,1
INC 3,3,SZK
JMP PDIV1
STA 0,OK0
STA 1,OK1
STA 2,OK2
JMP PDIVR
STA 3,MDRET
LUA 3,PSCRY
MOV 0,0,SNC
INC 3,3
MOVR 3,3,SNC
JMP MDCK1
MOV 0,0,SNC
JMP MDCK+1
JMP #MDRET
FC(CARRY) IS WRONG
END O.V. CHECK RESULT.
O.V. DONT CHECK.

```

```

10039 ECL20
01
02
03 02226 040304 EDIV: STA 0,OK0
04 02227 102400 SUBC 0,0
05 02230 12312 MOVL# 1,1,SZC
06 02231 102000 ADD 0,0
07 02233 101001 MOV 0,0,SKP
08 02233 040304 PSDIV: STA 0,OK0
09 02234 054307 STA 3,OK3
10 02235 050306 STA 2,OK2
11 02236 044305 STA 1,OK1
12 02237 151102 MOVL 2,3,SZC
13 02240 150400 NEG 2,2
14 02241 175500 SUBCL 3,3
15 02242 175100 MOVZL 3,3
16 02244 101113 MOVL# 0,0,SNC
17 02244 000405 JMP PSDIV
18 02245 175400 INC 3,3
19 02245 124400 NEG 1,1,SZC
20 02247 100041 COMD 0,0,SKP
21 02250 100440 NEG 0,0
22 02251 142430 SUBZ# 2,0,SZC
23 02252 028415 JMP PSDIV
24 02253 054046=PSDV2: STA 3,0
25 02254 034047= LDA 3,--20
26 02255 125120 MOVZL 1,1
27 02256 101100 MOVL 0,0
28 02257 142412 SUB# 2,0,SZC
29 02260 142400 SUB 2,0
30 02261 125100 MOVL 1,1
31 02262 175404 INC 3,3,SZC
32 02263 000773 JMP PSDIV+3
33 02264 030306 PSDIV3: LDA 2,OK2
34 02265 125113 MOVL# 1,1,SNC
35 02266 000404 JMP ++
36 02267 175520 PSDIV4: SUBZL 3,3
37 02270 054271 STA 3,PSCRY
38 02271 002307 JMP #OK3
39 02272 034046= LDA 3,0
40 02273 175203 MOVR 3,3,SNC
41 02274 174001 COM 3,3,SKP
42 02275 100400 NEG 0,0
43 02276 175203 MOVR 3,3,SNC
44 02277 124400 NEG 1,1
45 02300 175500 SUBOL 3,3
46 02301 054271 STA 3,PSCRY
47 02302 040304 STA 0,OK0
48 02303 044305 STA 1,OK1
49 02304 050306 STA 2,OK2
50 02305 002307 JMP #OK3

10040 ECL20
01
02
03 02306 054307 PSHUL: STA 3,OK3
04 02307 040257 STA 0,MTEM
05 02310 102400 SUBC 0,0
06 02311 176400 SUB 3,3
07 02312 125112 MOVL# 1,1,SZC
08 02313 157000 ADD 2,3
09 02314 151112 MOVL# 2,2,SZC
10 02315 137000 ADD 1,3
11 02315 054046= STA 3,0
12 02317 004030 JSR PSHUL
13 02320 034046= LDA 3,0
14 02321 162400 SUB 3,0
15 02322 034257 LDA 3,MTEM
16 02323 175113 MOVL# 3,3,SNC
17 02324 152401 FIN BIT 15
18 02325 150000 NEG 0,0,SKP
19 02326 157000 ADC 2,2
20 02327 151400 INC 2,2
21 02350 143000 ADD 2,0
22 02351 030306 STA 0,OK2
23 02352 040304 STA 0,OK0
24 02353 044305 STA 1,OK1
25 02354 002307 JMP #OK3

10040 ECL20
01
02
03 02356 002307 PSDIV0: SETUP 1000
04 02357 002307 JSR #ENTIN
05 02358 100000
06 02359 002307 RANST
07 02360 002307 JSR #1,RAN
08 02361 002307 JSR PSDIV
09 02362 024300 LDA 0,ORIG0
10 02363 024301 LDA 1,ORIG1
11 02364 030302 LDA 2,ORIG2
12 02365 157110 DIVS
13 02366 004650 JSR SHOCK
14 02367 000420 JMP SDIV2
15 02368 000420 MOERR
16 02369 101651 #SDIV1
17 02370 000300 ORIG0
18 02371 000301 ORIG1
19 02372 000302 ORIG2
20 02373 101605 #MULH2
21 02374 000245 ACC
22 02375 000246 AC1
23 02376 000247 AC2
24 02377 101611 #MULM3
25 02378 000304 OK0
26 02379 000305 OK1
27 02380 000306 OK2
28 02381 000000 #
29 02382 000000 SDIV2: LOOP
30 02383 000231 JSR #ENTLO
31 02384 000401 JMP SHUL0

```

SOFTWARE SIGNED MULTIPLY

UNSIGNED MULTIPLY

TEST #SDIV#  
INITIALIZE TEST....

RANDOM # TO ACS=2  
SIMULATE SIGNED DIVIDE.

#SDIV ACC 01 AC2  
FORIGH

#RESULT#

#CORRECT#

ITERATE TEST ROUTINE....

```

10041 ECL20
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
01 02422 000230
02 02423 000001
03 02424 024227
04 02425 032226
05 02426 032226
06 02427 044240
07 02428 044240
08 02429 050273
09
10 02431 000234
11 02432 111000
12 02433 101120
13 02434 101220
14 02435 024240
15 02436 122422
16 02437 000777
17 02440 123000
18 02441 151112
19 02442 103240
20 02443 024226
21 02444 123000
22 02445 042273
23 02446 010273
24 02447 024273
25 02450 024227
26 02451 105114
27 02452 000777
28
29 02453 000231
INITIALIZE TEST....
FILL EACH WORD OF THE
BUFFER WITH A PSEUDO
RANDOM ADDRESS. THESE
POINTERS ARE ALL WITHIN
THE RANGE OF THE BUFFER.
IC(ACB)#RANDOM #
MAKE RANDOM MODULE
BUFFER SIZE.
RANDOMLY SET THE
INDIRECT BIT.
ITERATE TO NEXT WORD
ITERATE TEST ROUTINE....
SETUP 1
JSR #ENTIN
1
LDA 1,REND
LDA 2,REBEG
SUB 2,1
STA 3,TEM
STA 2,TEM1
RAND
JSR #ENTRA
MOV 0,2
MOVZL 0,0
MOVZR 0,0
LDA 1,TEM
SUBZ 1,0,1SZC
JMP #1
ADD 1,0
MOVW 2,2,84C
ADDR 0,0
LDA 1,REBEG
ADD 1,0
STA 0,ITEM1
ISZ TEM1
LDA 0,REND
SUBW 0,1,1SZK
JMP #BBAU
LOOP
JSR #ENTLO
10042 ECL20
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
01 02422 000230
02 02423 000001
03 02424 024227
04 02425 032226
05 02426 032226
06 02427 044240
07 02428 044240
08 02429 050273
09
10 02431 000234
11 02432 111000
12 02433 101120
13 02434 101220
14 02435 024240
15 02436 122422
16 02437 000777
17 02440 123000
18 02441 151112
19 02442 103240
20 02443 024226
21 02444 123000
22 02445 042273
23 02446 010273
24 02447 024273
25 02450 024227
26 02451 105114
27 02452 000777
28
29 02453 000231
TEST "SMUL"
INITIALIZE TEST....
RANDOM # TO AC0=2
SIMULATE SIGNED MULTIPLY
SIGNED MUL
CHECK DATA
JOK
ERROR
"SHUM AC0 AC1 AC2
FORIG"
"RESULT"
"CORRECT"
ITERATE TEST ROUTINE....
SETUP 1000#
JSR #ENTIN
1000#
RAND
JSR #1,KAN
JSR #SMUL
LDA 1,ORIG0
LDA 1,ORIG1
LDA 2,ORIG0
MULS
JSR #SMUCK
JMP #MUL2
NOERR
#SMH1
ORIG0
ORIG1
ORIG2
#MULM2
AC0
AC1
AC2
#MULM3
OK0
OK1
OK2
%
SMUL21
LOOP
JSR #ENTLO
10041 ECL20
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
01 02374 000230
02 02375 010000
03 02376 000223
04 02377 004713
05 02378 020300
06 02379 024301
07 02376 030302
08 02377 147714
09 02400 000645
10 02401 000628
11 02404 101066
12 02405 000300
13 02406 000301
14 02407 000302
15 02410 101065
16 02411 000245
17 02412 000246
18 02413 000247
19 02414 101011
20 02415 000304
21 02416 000305
22 02417 000306
23 02420 000000
24
25 02421 000231
SMUL21
LOOP
JSR #ENTLO

```

```

10043 ECL20
01 02454 020226
02 02455 040240
03
04 02456 006230
05 02457 006005
06 02458 030240
07
08 02461 030044
09 02462 050020
10 02463 170520
11 02464 170520
12 02465 006451
13 02466 031000
14 02467 052020
15 02470 151132
16 02471 000773
17 02472 030000
18 02473 030020
19 02474 024044
20 02475 044020
21 02476 026020
22 02477 125120
23 02500 125220
24 02501 132414
25 02502 136415
26 02503 000433
27 02504 024020
28 02505 106414
29 02506 000770
30 02507 050274
31 02510 054275
32 02511 030240
33 02512 153240
34 02513 155000
35 02514 126420
36 02515 113710
37 02516 101002
38 02517 000406
39
40 02524 000411
41 02525 020274
42 02526 024240
43 02527 156415
44 02530 112414
45
46
47 02536 000231
48
49 02536 010240
50 02537 024227
51 02540 020043
52 02541 106400
53 02542 024240
54 02543 170614
55 02544 000712

```

```

10044 ECL20
01
02 02545 020226
03 02546 040240
04
05 02547 006230
06 02550 006005
07 02551 030240
08
09 02552 030044
10 02553 050020
11 02554 170520
12 02555 170522
13 02556 006451
14 02557 031000
15 02560 052020
16 02561 151132
17 02562 000773
18 02563 030000
19 02564 024044
20 02565 044020
21 02566 026020
22 02567 125120
23 02570 125120
24 02571 125220
25 02572 132414
26 02573 136415
27 02574 000433
28 02575 024020
29 02576 106414
30 02577 000770
31 02600 050274
32 02601 054275
33 02602 030240
34 02603 153240
35 02604 155000
36 02605 126420
37 02606 113710
38 02607 101002
39 02610 000406
40
41 02615 000411
42 02616 020274
43 02617 024240
44 02620 156415
45 02621 112414
46
47
48 02626 000231
49
50 02627 010240
51 02630 024227
52 02631 020043
53 02632 106400
54 02633 020240
55 02634 106614
56 02635 000712

```

```

10045 ECL20
01 02636 020226
02 02637 040240
03
04 02638 006230
05 02639 006005
06 02640 030240
07
08 02641 030044
09 02642 050020
10 02643 170520
11 02644 170522
12 02645 006451
13 02646 031000
14 02647 052020
15 02650 151132
16 02651 000773
17 02652 030000
18 02653 030020
19 02654 024044
20 02655 044020
21 02656 026020
22 02657 125120
23 02660 125220
24 02661 132414
25 02662 136415
26 02663 000433
27 02664 024020
28 02665 106414
29 02666 000770
30 02667 050274
31 02670 054275
32 02671 030240
33 02672 153240
34 02673 155000
35 02674 126420
36 02675 113710
37 02676 101002
38 02677 000406
39
40 02684 000411
41 02685 020274
42 02686 024240
43 02687 156415
44 02690 112414
45
46
47 02695 000231
48
49 02696 010240
50 02697 024227
51 02700 020043
52 02701 106400
53 02702 024240
54 02703 170614
55 02704 000712

```

```

10046 ECL20
01 02705 020226
02 02706 040240
03
04 02707 006230
05 02710 006005
06 02711 030240
07
08 02712 030044
09 02713 050020
10 02714 170520
11 02715 170522
12 02716 006451
13 02717 031000
14 02720 052020
15 02721 151132
16 02722 000773
17 02723 030000
18 02724 030020
19 02725 024044
20 02726 044020
21 02727 026020
22 02730 125120
23 02733 125220
24 02734 132414
25 02735 136415
26 02736 000433
27 02737 024020
28 02738 106414
29 02739 000770
30 02740 050274
31 02743 054275
32 02744 030240
33 02745 153240
34 02746 155000
35 02747 126420
36 02748 113710
37 02749 101002
38 02750 000406
39
40 02757 000411
41 02758 020274
42 02759 024240
43 02760 156415
44 02763 112414
45
46
47 02768 000231
48
49 02769 010240
50 02770 024227
51 02773 020043
52 02774 106400
53 02775 024240
54 02776 170614
55 02777 000712

```

```

;INDIRECT BLM TEST.
;SIMILAR TO PREVIOUS TEST.
;INITIALIZE TEST....
LUA 0,BREG
STA 0,TEM
SETUP 5
JSR @ENTIN
5
LUA 2,TEM
FIND BBB4
LUA 2,20
SUBZL 3,3
INCL 3,3,S/C
JMP BBB4
LUA 2,0,2
STA 2,0,20
MOVZL# 2,2,SZC
JMP #5
;AUTO INDEX REGISTERS
;ARE USED AS VARIABLES.
LUA 3,0,2
LUA 0,20
LUA 1,20
STA 1,20
LUA 1,0,20
MOVZL 1,1
MOVZL 1,2,SZK
SUB# 1,3,S/NK
JMP BBB4
LUA 1,20
SUB# 0,1,SZK
JMP #10
STA 3,TEM0
LUA 2,TEM
LUA 2,TEM
ADDOR 2,2
MOV 2,3
SUBO 1,1
BLM
MOV 0,0,SZC
JMP BBB2
ERROR
JMP BBB3
LUA 0,TEM2
LUA 1,TEM
SUB# 2,3,S/NK
SUB# 0,2,SZK
ERROR
LOOP
JSR @ENTLO
AGAIN RBR
ISZ TEM
LUA 1,BEND
LUA 0,110
SUB 0,1
LUA 0,TEM
AUC# 0,1,SZK
JMP BBB

```

```

;FIND THE FINAL ADDRESS
;AND VALUE IN THE INDIRECT
;CHAIN. C(AC2)= FIRST
;ADDRESS AT WHICH TO
;START LOOKING.
;IF INDIRECT CHAIN IS
;#15, EXIT TO BBB4.
;MAKE SURE THAT FINAL
;DATA WILL NOT POINT
;TO ANY SPOT IN THE CHAIN.
;IF IT DOES, EXIT TO BBB4.
;C(AC2)=FINAL ADDRESS
;C(AC3)=DATA AT THAT
;ADDRESS.
;BLM INDIRECT CHANGED
;C(CARRY).
;C(AC0)= CORRECT
;C(AC1)= ORIGINAL ADDRESS
;C(AC2)= BLM RESULT
;C(AC3)= SLM RESULT
;ITERATE TEST ROUTINE....
;ADVANCE TO NEXT BUFFER
;LOCATION AND TEST FOR
;END OF BUFFER-110. IF NOT
;END OF BUFFER GO TO BBB.

```

```

;BLM WITH INDIRECT
;POINTERS IN AC2/3 TEST.
;INITIALIZE TEST....
;FIND THE FINAL ADDRESS
;AND VALUE IN THE INDIRECT
;CHAIN. C(AC2)= FIRST
;ADDRESS AT WHICH TO
;START LOOKING.
;IF INDIRECT CHAIN IS
;#15, EXIT TO BBB4.
;AUTO INDEX REGISTERS
;ARE USED AS VARIABLES.
;MAKE SURE THAT FINAL
;DATA WILL NOT POINT
;TO ANY SPOT IN THE CHAIN.
;IF IT DOES, EXIT TO BBB4.
;C(AC2)=FINAL ADDRESS
;C(AC3)=DATA AT THAT
;ADDRESS.
;BLM INDIRECT CHANGED
;C(CARRY).
;C(AC0)= CORRECT
;C(AC1)= ORIGINAL
;C(AC2)= BLM RESULT
;C(AC3)= BLM RESULT
;ITERATE TEST ROUTINE....
;ADVANCE TO NEXT BUFFER
;LOCATION AND TEST FOR
;END OF BUFFER-110. IF NOT
;END OF BUFFER GO TO BBA.

```

```

;BLM WITH INDIRECT
;POINTERS IN AC2/3 TEST.
;INITIALIZE TEST....
;FIND THE FINAL ADDRESS
;AND VALUE IN THE INDIRECT
;CHAIN. C(AC2)= FIRST
;ADDRESS AT WHICH TO
;START LOOKING.
;IF INDIRECT CHAIN IS
;#15, EXIT TO BBB4.
;AUTO INDEX REGISTERS
;ARE USED AS VARIABLES.
;MAKE SURE THAT FINAL
;DATA WILL NOT POINT
;TO ANY SPOT IN THE CHAIN.
;IF IT DOES, EXIT TO BBB4.
;C(AC2)=FINAL ADDRESS
;C(AC3)=DATA AT THAT
;ADDRESS.
;BLM INDIRECT CHANGED
;C(CARRY).
;C(AC0)= CORRECT
;C(AC1)= ORIGINAL
;C(AC2)= BLM RESULT
;C(AC3)= BLM RESULT
;ITERATE TEST ROUTINE....
;ADVANCE TO NEXT BUFFER
;LOCATION AND TEST FOR
;END OF BUFFER-110. IF NOT
;END OF BUFFER GO TO BBA.

```

```

;BLM WITH INDIRECT
;POINTERS IN AC2/3 TEST.
;INITIALIZE TEST....
;FIND THE FINAL ADDRESS
;AND VALUE IN THE INDIRECT
;CHAIN. C(AC2)= FIRST
;ADDRESS AT WHICH TO
;START LOOKING.
;IF INDIRECT CHAIN IS
;#15, EXIT TO BBB4.
;AUTO INDEX REGISTERS
;ARE USED AS VARIABLES.
;MAKE SURE THAT FINAL
;DATA WILL NOT POINT
;TO ANY SPOT IN THE CHAIN.
;IF IT DOES, EXIT TO BBB4.
;C(AC2)=FINAL ADDRESS
;C(AC3)=DATA AT THAT
;ADDRESS.
;BLM INDIRECT CHANGED
;C(CARRY).
;C(AC0)= CORRECT
;C(AC1)= ORIGINAL
;C(AC2)= BLM RESULT
;C(AC3)= BLM RESULT
;ITERATE TEST ROUTINE....
;ADVANCE TO NEXT BUFFER
;LOCATION AND TEST FOR
;END OF BUFFER-110. IF NOT
;END OF BUFFER GO TO BBA.

```

```

10045 ECL2P
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59

;FORM A ADDRESS TEST
INITIALIZE TEST,....
;WITH THE "BAM" INSTRUCTION
;C(AC1)=SIZE OF BUFFER
;DESTINATION IS SOURCE+1
;FIRST BUFFER LUC GETS ITS
;ADDRESS. "BAM" WILL INC
;ALSO FILLED WITH ITS ADDRESS
;CHECK WORD FROM MEMORY
;C(AC2)=ADDRESS AND
;CORRECT
;"ADDR GOOD BAD"
;NEXT ADDRESS
;TEST FOR LAST ADDR
;ITERATE
;ITERATE TEST ROUTINE,....
;FIRST "BAM" WITH ADDRESS
INITIALIZE TEST,....
;PATTERN, RANDOM FROM/TO
;ADDRESS AND ADDITION OF V
;IF ERROR SET, LOCK
;COUNT NUMBERS

;C(AC1)=BUFFER SIZE
;MODULE BUFFER SIZE
;MODULE BUFFER SIZE
;C(AC1)=WORDS TO MOVE
;C(10)←C(FROM)

```

```

10046 ECL2P
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59

NEG 1,0
STA 3,0,3
INC 3,3
INCR 0,0,3
JMP *-3

LDA 3,1,FROM
XCH 2,3
RAM
STA 0,1,BAC0
STA 1,8,AC1
STA 2,1,AC2
STA 3,1,AC3
LDA 0,1,FROM
LDA 1,1,TO
LDA 2,1,BANSI4
ADD 2,0
STA 0,1,BX2
STA 1,8,OK3

LOA 2,1,AC2
SUBR 0,2,1,SNR
SUBR 1,1,1,SNR
JMP BLT6
LDA 0,1,BAC0
LVA 1,1,AC1
ADDR 0,1,1,SNR
JMP BLT6
MOV 0,1,0,1,SNR
JMP BLT7
BANKR
#BAMH4

BANKR
#BAMH1
K6
BANSI4
FROM
TUO
#BAMH2
K6
K6
RUK2
RUK3
#BAMH3
RAC0
RAC1
RAC2
RAC3
9

;STORE PATTERN
;C(ADDRESS)=ADDRESS
;C(AC1)=NUMBER OF WORDS
;IN THE PATTERN

;INIT AC'S FOR THE "BAM"
;***TRANSFER***
;SAVE RESULTS OF AC0-3

;CHECK THE AC'S LEFT BY
;THE "BAM" INSTRUCTION

;AC'S AFTER "BAM" INST WRDNG
;FORIG C(AC0-3)"
;ZERO
;NUMBER OF WORDS
;SOURCE
;DESTINATION
;GOOD CAC0-3"
;SUM
;COUNT
;SOURCE
;DESTINATION
;BAD C(AC0-3)"

```

10047 ECL20

```

01
02
03 03020 000231
04 03010 024316
05 03011 021000
06 03012 106414
07 03013 000407
08 03014 125400
09 03015 151400
10 03016 014315
11 03017 000772
12
13 03020 000231
14 03021 000426
15
16 03022 040260
17 03023 044261
18 03024 050262
19 03025 034313
20 03026 030315
21 03027 150400
22 03030 054263
23
24
25 03033 101472
26 03034 000261
27 03035 000262
28 03036 000263
29 03037 000261
30 03040 000264
31 03041 000261
32 03042 000260
33 03043 000000
34 03044 024261
35 03045 030262
36 03046 000746

```

```

LDA 2,T00
LDA 1,FR0M
LDA 0,0,2
SUB# 0,1,SZR
JMP BLT9
INC 1,1
INC 2,2
DSZ BANI0
JME BLT7*2
LOOP
JSR #ENTL0
JMP BAN0
STA 0,0UK0
STA 1,0UK1
STA 2,0UK2
LDA 3,0UK3
LDA 2,0UK4
SUB 2,1,3
STA 3,0UK3
#BAM5
#BAM5
0UK1
0UK2
0UK3
0UK4
0UK5
0UK6
0UK7
0UK8
LDA 1,0UK1
LDA 2,0UK2
JMP BLT8

```

```

10048 ECL20
01
02
03 03047 000230
04 03050 000200
05 03051 020210
06 03052 020210
07 03053 020210
08 03054 125004
09 03055 040277
10 03056 040277
11 03057 040277
12 03058 115300
13 03059 030227
14 03061 024226
15 03062 132468
16 03063 150400
17 03064 000777
18 03065 157000
19 03066 142422
20 03067 000777
21 03070 113000
22 03071 172432
23 03072 154710
24 03073 024226
25 03074 135000
26 03075 137000
27 03076 024227
28 03077 166400
29 03100 044313
30 03101 044315
31 03102 050317
32 03103 054316
33 03104 054314
34
35
36 03105 000235
37 03106 042314
38 03107 010314
39 03110 014315
40 03111 000774
41
42 03112 030042
43 03113 034041
44 03114 024313
45 03115 101320
46 03116 040265
47

```

```

SETUP 200
JSR #ENTN
200
LDA 1,ITRER
LDA 0,FRAN
MOV 1,1,SZR
LDA 0,SRAN
STA 0,FRAN
MOV# 0,1,3
LDA 2,0END
JUFFER END
JUFFER BEGIN
JC(AC2)*SIZE OF JUFFER.
JC(AC3)*RANDOM MODULO BUFF SIZE.
JC(AC2)*RANDOM MODULO BUFF SIZE.
JMAKE DESTINATION<SOURCE
JC(AC2)*C(AC3)
JNUMBER OF WORDS TO TRANSFER.
JDESTINATION BLOCK
JSOURCE BLOCK
JFILL THE SOURCE BLOCK WITH
RANDOM NUMBERS.
JINITIALIZE THE C(AC0-3) FOR
THE BAN INSTRUCTION
JTHE NUMBER TO BE ADDED.

```

10040 ECL20

01  
02  
03 03117 113710 BAMB:  
04 03120 000320 STA 0,BAC0  
05 03121 044321 STA 1,BAC1  
06 03122 050322 STA 2,BAC2  
07 03123 054323 STA 3,BAC3  
08 03124 102420 SUBC 0,0  
09 03125 040261 STA 0,BUK1  
10 03126 020316 LDA 0,FR0M  
11 03127 024317 LDA 1,T00  
12 03130 030313 LDA 2,BAMS1Z  
13 03131 140004 ADD 2,1  
14 03132 147000 STA 0,BUK2  
15 03133 040262 STA 1,BUK3  
16 03134 044263 LDA 2,BAC2  
17 03135 030322 SUB# 0,2,SNR  
18 03136 112415 SUB# 1,3,SZR  
19 03137 135414 JMP BAMB  
20 03140 000411 LDA 0,BAMSUM  
21 03141 020265 LDA 1,BAC0  
22 03142 024320 LDA 2,BAC1  
23 03143 030321 MOV 2,2,SNR  
24 03144 151005 JMP BAMB  
25 03145 000403 MOV 0,0,SNR  
26 03146 000403 JMP BAMB  
27 03147 101003  
28 03150 000423

01  
02  
03 03153 101404 BAMB:  
04 03154 000265 STA 0,BAC0  
05 03155 000313 STA 1,BAC1  
06 03156 000316 STA 2,BAC2  
07 03157 000317 STA 3,BAC3  
08 03161 000265 SUBC 0,0  
09 03162 000264 STA 0,BUK1  
10 03163 000262 STA 1,BUK2  
11 03164 000263 STA 2,BAM1Z  
12 03175 020317 BAMB:  
13 03176 020317 BAMB:  
14 03177 020315 BAMB:  
15 03200 000315 BAMB:  
16 03201 000235 BAMB:  
17 03202 020314 BAMB:  
18 03203 030265 BAMB:  
19 03204 113000 BAMB:  
20 03205 132414 BAMB:  
21 03206 000406 BAMB:  
22 03207 010314 BAMB:  
23 03210 014315 BAMB:  
24 03211 000770 BAMB:  
25 03212 000231 BAMB:  
26 03213 000427 BAMB:  
27 03214 050262 BAMB:  
28 03215 040260 BAMB:  
29 03216 044261 BAMB:  
30 03217 034313 BAMB:  
31 03220 020315 BAMB:  
32 03221 110400 BAMB:  
33 03222 044263 BAMB:  
34 03223 020316 BAMB:  
35 03224 150000 BAMB:  
36 03225 040320 BAMB:  
37 03230 101472 BAMB:  
38 03231 000320 BAMB:  
39 03232 000314 BAMB:  
40 03233 000265 BAMB:  
41 03234 000260 BAMB:  
42 03235 000265 BAMB:  
43 03236 000262 BAMB:  
44 03237 000261 BAMB:  
45 03240 000000 BAMB:  
46 03241 000746 BAMB:  
47  
48  
49  
50

10000 ECL20

01  
02  
03 03165 101447  
04 03166 000320  
05 03167 000321  
06 03170 000322  
07 03171 000323  
08 03172 000000  
09 03173 020277 BAMB:  
10 03174 040276 BAMB:  
11 03175 020317 BAMB:  
12 03176 040314 BAMB:  
13 03177 020315 BAMB:  
14 03178 000315 BAMB:  
15 03200 000315 BAMB:  
16 03201 000235 BAMB:  
17 03202 020314 BAMB:  
18 03203 030265 BAMB:  
19 03204 113000 BAMB:  
20 03205 132414 BAMB:  
21 03206 000406 BAMB:  
22 03207 010314 BAMB:  
23 03210 014315 BAMB:  
24 03211 000770 BAMB:  
25 03212 000231 BAMB:  
26 03213 000427 BAMB:  
27 03214 050262 BAMB:  
28 03215 040260 BAMB:  
29 03216 044261 BAMB:  
30 03217 034313 BAMB:  
31 03220 020315 BAMB:  
32 03221 110400 BAMB:  
33 03222 044263 BAMB:  
34 03223 020316 BAMB:  
35 03224 150000 BAMB:  
36 03225 040320 BAMB:  
37 03230 101472 BAMB:  
38 03231 000320 BAMB:  
39 03232 000314 BAMB:  
40 03233 000265 BAMB:  
41 03234 000260 BAMB:  
42 03235 000265 BAMB:  
43 03236 000262 BAMB:  
44 03237 000261 BAMB:  
45 03240 000000 BAMB:  
46 03241 000746 BAMB:  
47  
48  
49  
50

DO THE "BAM" TRANSFER...  
SAVE C(ACC-3) FOR TEST.  
CHECK THE AC RESULT  
OF THE BAM INSTRUCTION  
AFTER BAM THE SOURCE  
AND DESTINATION ACS  
SHOULD BE INCREMENTED  
BY THE BLOCK SIZE.  
CHECK SOURCE/DEST INCREMENT  
ERROR  
BAM SHOULD EXIT  
IF THE C(ACC) UNCHANGED AND,  
FCOUNT SHOULD BE 0  
ERROR  
FACIS OK AFTER BAM,  
"BAM" SET C(CARRY).  
FACIS AFTER "BAM" INSTRUCTION WRONG"  
"ORIG C(ACC-3)"  
FSUM  
FSIZE  
FDESTINATION ADDRESS  
FGOOD C(ACC-3)"  
FC(ACC) AMOUNT TO ADD  
FC(ACC) SOURCE  
FC(ACC) DESTINATION

01  
02  
03 03165 101447  
04 03166 000320  
05 03167 000321  
06 03170 000322  
07 03171 000323  
08 03172 000000  
09 03173 020277 BAMB:  
10 03174 040276 BAMB:  
11 03175 020317 BAMB:  
12 03176 040314 BAMB:  
13 03177 020315 BAMB:  
14 03178 000315 BAMB:  
15 03200 000315 BAMB:  
16 03201 000235 BAMB:  
17 03202 020314 BAMB:  
18 03203 030265 BAMB:  
19 03204 113000 BAMB:  
20 03205 132414 BAMB:  
21 03206 000406 BAMB:  
22 03207 010314 BAMB:  
23 03210 014315 BAMB:  
24 03211 000770 BAMB:  
25 03212 000231 BAMB:  
26 03213 000427 BAMB:  
27 03214 050262 BAMB:  
28 03215 040260 BAMB:  
29 03216 044261 BAMB:  
30 03217 034313 BAMB:  
31 03220 020315 BAMB:  
32 03221 110400 BAMB:  
33 03222 044263 BAMB:  
34 03223 020316 BAMB:  
35 03224 150000 BAMB:  
36 03225 040320 BAMB:  
37 03230 101472 BAMB:  
38 03231 000320 BAMB:  
39 03232 000314 BAMB:  
40 03233 000265 BAMB:  
41 03234 000260 BAMB:  
42 03235 000265 BAMB:  
43 03236 000262 BAMB:  
44 03237 000261 BAMB:  
45 03240 000000 BAMB:  
46 03241 000746 BAMB:  
47  
48  
49  
50

"BAD C(ACC-3)"  
RESULTING C(ACC-3) FROM BAM.  
PRESET RANDOM NUMBER FOR  
CHECK ROUTINE.  
STARTING POINT  
NUMBER OF WORDS TO CHECK  
GENERATE ORIGINAL RANDOM  
GET WORD FROM MEMORY  
FORIG\*(ACC) SHOULD BE CORRECT.  
DATA ERROR  
ITERATE FOR ALL WORDS.  
ITERATE TEST ROUTINE...  
NEXT TEST  
FORM...  
GOOD/ORIG/BAD  
C(BAM1)=DESTINATION (TO)  
WORD  
FROM TO WORD ORIG C(ACC)  
FROM RADII  
FROM ADDRESS  
WORD  
FORIG  
GOOD  
BAD  
JMP BAMB1

01  
02  
03 03165 101447  
04 03166 000320  
05 03167 000321  
06 03170 000322  
07 03171 000323  
08 03172 000000  
09 03173 020277 BAMB:  
10 03174 040276 BAMB:  
11 03175 020317 BAMB:  
12 03176 040314 BAMB:  
13 03177 020315 BAMB:  
14 03178 000315 BAMB:  
15 03200 000315 BAMB:  
16 03201 000235 BAMB:  
17 03202 020314 BAMB:  
18 03203 030265 BAMB:  
19 03204 113000 BAMB:  
20 03205 132414 BAMB:  
21 03206 000406 BAMB:  
22 03207 010314 BAMB:  
23 03210 014315 BAMB:  
24 03211 000770 BAMB:  
25 03212 000231 BAMB:  
26 03213 000427 BAMB:  
27 03214 050262 BAMB:  
28 03215 040260 BAMB:  
29 03216 044261 BAMB:  
30 03217 034313 BAMB:  
31 03220 020315 BAMB:  
32 03221 110400 BAMB:  
33 03222 044263 BAMB:  
34 03223 020316 BAMB:  
35 03224 150000 BAMB:  
36 03225 040320 BAMB:  
37 03230 101472 BAMB:  
38 03231 000320 BAMB:  
39 03232 000314 BAMB:  
40 03233 000265 BAMB:  
41 03234 000260 BAMB:  
42 03235 000265 BAMB:  
43 03236 000262 BAMB:  
44 03237 000261 BAMB:  
45 03240 000000 BAMB:  
46 03241 000746 BAMB:  
47  
48  
49  
50

01  
02  
03 03165 101447  
04 03166 000320  
05 03167 000321  
06 03170 000322  
07 03171 000323  
08 03172 000000  
09 03173 020277 BAMB:  
10 03174 040276 BAMB:  
11 03175 020317 BAMB:  
12 03176 040314 BAMB:  
13 03177 020315 BAMB:  
14 03178 000315 BAMB:  
15 03200 000315 BAMB:  
16 03201 000235 BAMB:  
17 03202 020314 BAMB:  
18 03203 030265 BAMB:  
19 03204 113000 BAMB:  
20 03205 132414 BAMB:  
21 03206 000406 BAMB:  
22 03207 010314 BAMB:  
23 03210 014315 BAMB:  
24 03211 000770 BAMB:  
25 03212 000231 BAMB:  
26 03213 000427 BAMB:  
27 03214 050262 BAMB:  
28 03215 040260 BAMB:  
29 03216 044261 BAMB:  
30 03217 034313 BAMB:  
31 03220 020315 BAMB:  
32 03221 110400 BAMB:  
33 03222 044263 BAMB:  
34 03223 020316 BAMB:  
35 03224 150000 BAMB:  
36 03225 040320 BAMB:  
37 03230 101472 BAMB:  
38 03231 000320 BAMB:  
39 03232 000314 BAMB:  
40 03233 000265 BAMB:  
41 03234 000260 BAMB:  
42 03235 000265 BAMB:  
43 03236 000262 BAMB:  
44 03237 000261 BAMB:  
45 03240 000000 BAMB:  
46 03241 000746 BAMB:  
47  
48  
49  
50

```

18051 ECL20
M1 00000 000000
02 03243 000004
03 03245 000004
04 03244 024227
05 03245 036226
06 03246 146460
07 03247 155400
08 03250 020403
09 03251 141002
10 03252 102400
11 03253 113710
12 03254 004402
13 03255 004415
14 03255 004206
15 03257 054040
16 03260 056227
17
18 03261 102400
19 03262 126400
20 03263 152400
21 03264 176400
22 03265 102400
23 03266 126400
24 03267 152400
25 03270 176400
26 03271 002226
27
28 03272 107015
29 03273 157014
30 03274 000416
31 03275 034756
32 03276 024927
33 03277 030926
34 03300 146400
35 03301 124400
36 03302 021000
37 03303 162414
38 03304 000417
39 03305 151400
40 03306 125404
41 03307 000773
42
43 03310 000631
44 03311 000422
45
46 03314 101532
47 03315 000245
48 03316 000246
49 03317 000247
50 03320 000250
51 03321 000000
52 03322 000766
53
54 03325 101551
55 03326 000247
56 03327 000250
57 03330 000247
58 03331 000000
59 03332 000753

18052 ECL20
01
02
03 03333 006230
04 03334 000002
05 03335 024227
06 03336 036226
07 03337 140000
08 03340 155400
09 03341 175400
10 03342 040313
11 03343 020037
12 03344 041000
13 03345 020036
14 03346 041001
15 03347 102440
16 03350 113710
17 03351 004492
18 03352 004496
19 03353 042770
20 03354 024313
21 03355 125200
22 03356 044313
23 03357 002226
24
25
26
27 03360 054267
28 03361 014313
29 03362 102401
30 03363 000405
31 03364 024313
32 03365 036226
33 03366 155000
34 03367 002267
35
36 03370 036226
37 03371 034227
38 03372 156400
39 03373 054313
40 03374 020037
41 03375 024036
42 03376 035000
43 03377 116415
44 03380 000410
45
46 03403 101551
47 03404 000247
48 03405 000245
49 03406 000050
50 03407 000000
51 03410 104710
52 03411 151400
53 03412 014313
54 03413 000753
55
56 03414 006231
57

I TEST "BAM"
I INITIALIZE TEST....
SETUP 2
JSR #ENTLO
LOA 1,#BND
LOA 2,#BREG
INC 2,1
INC 2,3
INC 3,3
STA 1,BAMSIZ
LOA 0,#JSR #BAMXR
STA 0,0,2
LOA 0,#BAM
STA 0,1,2
SUBD 0,0
JSR #+2
JMP BAMXR
STA 3,BAMXR
LOA 1,BAMSIZ
MOVH 1,1
STA 1,BAMSIZ
JMP #BBEG
IFILL BUFFER WITH
IJSR/BAM
IFIX C(CAC0-3) FOR
INEXT "BAM" INST
IEND OF BUFFER
IRETURN TO BUFFER
ICHECK C(BUFFER)
IFOR ACCURACY
I"ADON GOOD BAD"
I"SHOULD BE ZERU, C(CAC0-3)*"
I"ADDRESS GOOD BAD FAILED"
I"NEXT WORD IS DIFFERENT
ICOUNT WORDS
ITERATE CHECK
IFEXIT TEST
ITERATE TEST ROUTINE.....

```



10054 ECL2P

```

01
02
03
04 03415 006230
05 03416 000005
06 03417 020241
07 03420 126400
08 03421 044840
09 03422 122210
10
11
12 03427 006231
13
14
15 03430 006230
16 03431 000005
17 03432 020241
18 03433 126400
19 03434 044840
20 03435 152400
21 03436 142210
22
23
24 03443 006231
25
26
27 03444 006230
28 03445 000005
29 03446 020241
30 03447 105200
31 03450 131800
32 03451 155600
33 03452 102010
34 03453 126110
35 03454 152210
36 03455 176310
37 03456 176310
38 03457 000401
39 03460 105415
40 03461 156414
41
42 03466 034241
43 03467 172415
44 03470 146414
45
46
47 03475 006231
48

```

```

SZB41
SETUP 5
JSR #ENTLN
5
LDA 0,0
SUB 1,1
STA 1,TEM
SZA 1,0
EKROR
LOOP
JSR #ENTLN

SZB11
SETUP 5
JSR #ENTLN
5
LDA 0,0
ADCR 1,1
STA 1,TEM
SUB 2,0
SZA 2,0
EKROR
LOOP
JSR #ENTLN

SZB21
SETUP 5
JSR #ENTLN
5
LDA 0,0
MOV 0,1
MOV 1,2
MOV 2,3
MOV 0,0
RTZ 1,1
SZA 2,0
SZB0 3,3
SZB0 3,3
JMP +1
SUB# 0,1,SZR
SUB# 2,3,SZR
EKROR
LDA 3,0
SUB# 3,2,SZR
SUB# 2,1,SZR
EKROR
LOOP
JSR #ENTLN

```

```

TEST "SZB"
INITIALIZE TEST....
JST C(TEM) TO ZERO
JAND POINT TEM VIA
JBIT COUNT
ITERATE TEST ROUTINE....

TEST "SZB"
INITIALIZE TEST....
JBIT 0=(0) OTHER BITS=(1).
JPRINT C(TEM) VIA
JBIT COUNT
JSHOULD SKIP
ITERATE TEST ROUTINE....

TEST "BIT GROUP"
INITIALIZE TEST....
JBIT INSTRUCTIONS SHOULD
JNOT CHANGE THE CONTENTS OF
JANY AC
JARE AC=3 UNCHANGED?
JNO, ERROR
ITERATE TEST ROUTINE....

```

```

LDA 0,0
SUB# 1,1
STA 1,TEM
SETUP 40
JSR #ENTLN
40
SZA 0,0
JMP SZB11
EKROR
LOOP
JSR #ENTLN
INC 0,0
LDA 1,TEM
MOV# 1,1,SZR
JMP SZB3

LDA 1,0
ADCR 2,2
STA 2,TEM
SETUP 40
JSR #ENTLN
40
SZA 1,1
EKROR
LOOP
JSR #ENTLN
INC 1,1
LDA 2,TEM
MOV# 2,2,SZR
JMP SZB4

SETUP 40
JSR #ENTLN
40
RAND
JSR #ENTLN
STA 0,TEM
LDA 2,0
SZA 1,1
SZA 2,2
MOV# 3,3,SKP
MOV# 3,3
INC 2,2
MOV# 1,1,SZR
JMP SZB41
SUB# 3,0,SZR
EKROR
LOOP
JSR #ENTLN

```

```

JSHIFT A (1) BIT THROUGH
J(TEM). THIS BIT IS POINTED
TO VIA SZB. THUS "SZB"
JSHOULD NEVER SKIP
JINITIALIZE TEST....
J(CAC1)=C(TEM)
J(CAC2)=POINTER TO C(TEM)
ITERATE TEST ROUTINE....

JSHIFT A (0) BIT THROUGH
J(TEM). THIS BIT IS POINTED
TO VIA SZB. THUS "SZB"
JSHOULD NEVER SKIP
JINITIALIZE TEST....
J(CAC1)=C(TEM)
J(CAC2)=POINTER TO C(TEM)
ITERATE TEST ROUTINE....

JTEST "SZB"
JINITIALIZE TEST....
J(CAC0)=RANDOM #
JA RANDOM NUMBER IS STORED
JIN LOCATION TEM. THE
J"SZB" INSTRUCTION WITH A
JPOINTER IN C(AC2) IS USED TO
JASSEMBLE A LINE WORD IN C(AC3)

J(CAC0)=CORRECT
J(CAC2)=POINTER TO TEM
J(CAC3)=RESULT OF SZB INSTRUCTIONS
ITERATE TEST ROUTINE....

```

10055 ECL20

```

01
02
03 03357 000230
04 03358 000040
05
06 03361 000234
07 03362 040240
08 03363 034241
09 03364 120520
10 03365 170210
11 03366 151141
12 03367 151120
13 03370 175400
14 03371 175124
15 03372 000773
16 03373 142414
17
18
19 03300 000231
20
21
22 03601 000230
23 03602 000020
24 03603 102420
25 03604 040240
26 03605 030750-
27
28 03606 020044-
29 03607 100110
30 03610 142210
31 03611 000401
32
33 03612 101004
34 03613 000774
35 03614 020240
36 03615 101002
37 03616 101004
38
39
40 03623 000231
41
42
43 03624 000230
44 03625 000020
45 03626 102040
46 03627 040240
47 03630 020044-
48 03631 030050-
49 03632 100110
50 03633 102210
51 03634 101004
52 03635 000775
53 03636 020240
54 03637 101002
55 03640 100014
56
57
58 03645 000231

```

10056 ECL20

```

01
02
03 03646 020226 SZB01
04 03647 040313 SZB01:
05
06 03650 000230
07 03651 000002
08
09 03652 000234
10 03653 030044-
11 03654 042313
12 03655 034035-
13 03656 110110- SZB02:
14 03657 172210
15 03660 125241
16 03661 125220
17 03662 151004
18 03663 000773
19 03664 032313 SZB03:
20 03665 142415
21 03666 142414
22
23
24 03673 000231
25 03674 010313
26 03675 020313
27 03676 024227
28 03677 106414 SZB04:
29 03700 000750
30
31 03701 024227 SZB05:
32 03702 030226
33
34 03703 140400
35 03704 100410
36 03705 165710
37 03706 050313
38 03707 044240 SZB09:
39
40 03710 000230
41 03711 000002
42
43 03712 000234
44 03713 024240
45 03714 030226
46 03715 100510
47 03716 133000
48 03717 041000
49 03720 050275
50 03721 024240
51 03722 034226
52 03723 152520
53
54
55

```

10055 ECL20

```

/TEST "SZB"
/INITIALIZE TEST.....
/SEE PREVIOUS TEST
/((AC0)=RANDOM #
/INITIALIZE TEST.....
/IN THE DATA BUFFER,
/((AC0)=RANDOM #
/INTERROGATE THIS NUMBER
/VIA THE "SZB" INSTRUCTION
/BIT COUNT CHANGES FROM
/17 TO 0 WITH C(AC3)
/POINTING INDIRECT TO THE
/WORD UNDER TEST
/((AC3)=WORD ADDRESS
/((AC2)=WORD FROM MEMORY
/((AC1)=ASSEMBLY VIA SZB
/((AC0)=CORRECT, ORIGINAL WORD
/ITERATE TEST ROUTINE.....
/ADVANCE TO NEXT BUFFER WORD
/TEST "SZB"
/STORE A RANDOM NUMBER
/IN THE BUFFER. THIS WORD IS
/POINTED TO VIA C(BBEG) AND
/FA BIT COUNT ASSEMBLE A
/LIKE WORD IN A C(AC2) BY
/SCANNING THIS WORD WITH A
/"SZB" INSTRUCTION. THE COMPUTED
/WORD IS CHECKED WITH THE
/INITIALIZE TEST.....
/ORIGINAL AND THE CONTENTS
/((AC0)=RANDOM #
/OF MEMORY.

```

```

10058 ECL20
01
02 03724 166210 SZR92: SZB 3,1
03 03725 151040 INC 1,1
04 03726 125400 MOV 2,2,SNC
05 03727 151160 MVL 2,2,SNC
06 03728 000774 JMP SZR92
07 03731 036875 LDA 3,0,TEM
08 03732 116415 SUR# 0,1,SZR
09 03733 112414 SUB# 0,1,2,SZR
10 ERROR
11 LOOP
12 03740 006231 JSR #ENTLO
13 03741 044240 STA 1,TEM
14 03742 030313 LDA 2,0,AMS17
15 03743 046414 SUB# 2,1,SZK
16 03744 000744 JMP SZR91

SZBA:
17
18 03745 006231 SZBA1: SETUP 5
19 03746 000005 JSR #ENTIN
20 03747 030226 LDA 2,0,BEG
21 03748 020034 MVA 0,0,2
22 03750 020034 STA 0,0,2
23 03751 041000 LDA 0,0,JSR 0,3
24 03752 020033 INC 3,3
25 03753 041001 INC 3,3
26 03754 155400 LDA 1,BEND
27 03755 175400 SUB 2,1
28 03756 024227 LDA 1,BEND
29 03757 145400 INC 3,3
30 03760 102120 AOCZL 0,0
31 03761 040240 STA 0,TEM
32 03762 102400 SUB 0,0
33 03763 113710 RAM
34 03764 030227 SZBA1: LDA 2,BEND
35 03765 020030 LDA 0,0,JSR 0,2
36 03766 041377 STA 0,-1,2
37 03767 041376 STA 0,-2,2
38 03770 004402 JSR #ZBA4
39 03771 000413 JMP SZBA4
40 03772 171000 MOV 3,2
41 03773 004402 JSR #*2
42 03774 000404 JMP SZBA3
43 03775 020030 LDA 0,0,TEM
44 03776 024031 LDA 1,1,17
45 03777 002226 SZBA2: JMP #BEG
46
47
48
49 04064 006231 SZBA3: ERROR
50 LOOP
51 04065 010240 SZBA4: JSR #ENTLO
52 04070 020240 ISZ TEM
53 04071 100714 LDA 0,TEM
54 04072 000763 SUB# 0,1,SZK
55 JMP SZBA1

```

```

10058 ECL20
01
02 04065 006230 SZBB1: SETUP 5
03 04066 000005 JSR #ENTIN
04 04067 030226 LDA 2,0,BEG
05 04068 020034 MVA 0,0,2
06 04069 020034 STA 0,0,2
07 04071 041000 LDA 0,0,JSR 1,1
08 04072 020027 INC 3,3
09 04073 041001 INC 3,3
10 04074 155400 LDA 1,BEND
11 04075 175400 SUB 2,1
12 04076 024227 LDA 1,BEND
13 04077 145400 INC 3,3
14 04080 102400 RAM
15 04081 113710 LDA 2,BEND
16 04082 030227 SZBB1: LDA 0,0,JSR 0,3
17 04083 020033 STA 0,-1,2
18 04084 041377 LDA 1,BEG
19 04085 041000 SUB 2,2
20 04086 024226 LDA 1,BEG
21 04087 155400 JSR #BEG
22 04088 030226 SZBB2: LDA 0,0,TEM
23 04089 020227 LDA 2,0,BEG
24 04092 030226 SUB# 2,0
25 04093 145400 ADD 0,2
26 04094 113000 SUR# 1,2,SZR
27 04095 132654 ERROR
28 LOOP
29 JSR #ENTLO
30 04042 006231 SZBC1: LDA 1,REND
31 04043 024227 SZBC1: LDA 2,0,BEG
32 04044 030226 STA 2,TEM
33 04045 050040 SUB 2,1
34 04046 146400 INC 2,3
35 04047 155400 LDA 0,0,177377
36 04048 020026 MVA 0,0,2
37 04049 020026 STA 0,0,2
38 04051 041000 RAM
39 04052 102400 LDA 1,REND
40 04053 113710 SUB# 0,1,SZK
41 04054 024227 SZBC1: SETUP 4
42 04055 006230 JSR #ENTIN
43 04056 000004 LDA 3,0,7
44 04057 030025 STA 2,TEM
45 04058 030024 SUB# 2,3
46 04059 030024 ERROR
47 04061 155210 LOOP
48
49 04060 006231 JSR #ENTLO
50 04061 010240 ISZ TEM
51 04062 020240 LDA 0,TEM
52 04063 100714 SUB# 0,1,SZK
53 04064 000763 JMP SZBA1

```

```

10057 ECL20
01
02 03724 166210 SZR92: SZB 3,1
03 03725 151040 INC 1,1
04 03726 125400 MOV 2,2,SNC
05 03727 151160 MVL 2,2,SNC
06 03728 000774 JMP SZR92
07 03731 036875 LDA 3,0,TEM
08 03732 116415 SUR# 0,1,SZR
09 03733 112414 SUB# 0,1,2,SZR
10 ERROR
11 LOOP
12 03740 006231 JSR #ENTLO
13 03741 044240 STA 1,TEM
14 03742 030313 LDA 2,0,AMS17
15 03743 046414 SUB# 2,1,SZK
16 03744 000744 JMP SZR91

SZBA:
17
18 03745 006231 SZBA1: SETUP 5
19 03746 000005 JSR #ENTIN
20 03747 030226 LDA 2,0,BEG
21 03748 020034 MVA 0,0,2
22 03750 020034 STA 0,0,2
23 03751 041000 LDA 0,0,JSR 0,3
24 03752 020033 INC 3,3
25 03753 041001 INC 3,3
26 03754 155400 LDA 1,BEND
27 03755 175400 SUB 2,1
28 03756 024227 LDA 1,BEND
29 03757 145400 INC 3,3
30 03760 102120 AOCZL 0,0
31 03761 040240 STA 0,TEM
32 03762 102400 SUB 0,0
33 03763 113710 RAM
34 03764 030227 SZBA1: LDA 2,BEND
35 03765 020030 LDA 0,0,JSR 0,2
36 03766 041377 STA 0,-1,2
37 03767 041376 STA 0,-2,2
38 03770 004402 JSR #ZBA4
39 03771 000413 JMP SZBA4
40 03772 171000 MOV 3,2
41 03773 004402 JSR #*2
42 03774 000404 JMP SZBA3
43 03775 020030 LDA 0,0,TEM
44 03776 024031 LDA 1,1,17
45 03777 002226 SZBA2: JMP #BEG
46
47
48
49 04064 006231 SZBA3: ERROR
50 LOOP
51 04065 010240 SZBA4: JSR #ENTLO
52 04070 020240 ISZ TEM
53 04071 100714 LDA 0,TEM
54 04072 000763 SUB# 0,1,SZK
55 JMP SZBA1

```

10059 ECL20

01  
02  
03

BTR:

SETUP 20  
JSR #ENTIN

TEST "BTO"  
INITIALIZE TEST....  
SHOULD SET BIT 0  
OF WORD TEM.

04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43

04 04073 006230  
05 04074 000020  
06 04075 102400  
07 04076 040240  
08 04077 176820  
09 04100 024032  
10 04101 126010  
11 04102 030240  
12 04103 101003  
13 04104 156404  
14  
15  
16 04111 006231  
17  
18  
19 04112 006230  
20 04113 000020  
21 04114 102400  
22 04115 040240  
23 04116 176820  
24 04117 030024  
25 04120 152010  
26 04121 024240  
27 04122 101003  
28 04123 136414  
29  
30  
31 04130 006231  
32  
33 04131 006230  
34 04132 000040  
35 04133 176820  
36 04134 102400  
37 04135 020023  
38 04136 020023  
39 04140 102010  
40 04141 175124  
41 04142 080775  
42 04143 024240  
43 04144 124014  
44  
45  
46 04151 006231  
47  
48  
49

10060 ECL20

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43

B13:

SETUP 440  
JSR #ENTIN

TEST "BTO"  
INITIALIZE TEST....  
RANDOM NUMBER IS STORED  
IN C(TEM). FOR EACH (1) BIT  
IS GENERATED AND A "BTO"  
IS EXECUTED. C(TEM) SHOULD NOT  
BE CHANGED BY THE SEVERAL  
"BTO" INSTRUCTIONS.  
I WILL ITERATE FOR EACH (1) BIT.  
C(AC2)=BTO RESULT  
C(AC1)=CORRECT  
C(CARRY) SHOULD REMAIN (1).  
ITERATE TEST ROUTINE....

04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43

04 04152 006230  
05 04153 000440  
06 04154 006234  
07 04155 105045  
08 04156 000776  
09 04157 040240  
10 04160 034241  
11 04161 110510  
12 04162 176010  
13 04163 101004  
14 04164 003774  
15 04165 030240  
16 04166 101002  
17 04167 140414  
18  
19  
20 04174 006231  
21  
22 04175 006230  
23 04176 000040  
24  
25 04177 006234  
26 04200 104000  
27 04201 125045  
28 04202 000775  
29 04203 040240  
30 04204 034241  
31 04205 136510  
32 04206 176010  
33 04207 125004  
34 04208 100774  
35 04211 024240  
36 04212 101002  
37 04213 124014  
38  
39  
40 04220 006231  
41  
42  
43

B14:

SETUP 44  
JSR #ENTIN

TEST "BTO"  
INITIALIZE TEST....  
RANDOM NUMBER IS STORED  
IN C(TEM). FOR EACH (0) BIT  
IS GENERATED AND A "BTO"  
IS EXECUTED. C(TEM) SHOULD NOT  
BE CHANGED BY THE SEVERAL  
"BTO" INSTRUCTIONS.  
I WILL ITERATE FOR EACH (0) BIT.  
C(AC1)=BTO RESULT  
C(CARRY) SHOULD REMAIN (1).  
ITERATE TEST ROUTINE....

04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43

04 04175 006230  
05 04176 000040  
06 04177 006234  
07 04200 104000  
08 04201 125045  
09 04202 000775  
10 04203 040240  
11 04204 034241  
12 04205 136510  
13 04206 176010  
14 04207 125004  
15 04208 100774  
16 04211 024240  
17 04212 101002  
18 04213 124014  
19  
20 04174 006231  
21  
22 04175 006230  
23 04176 000040  
24  
25 04177 006234  
26 04200 104000  
27 04201 125045  
28 04202 000775  
29 04203 040240  
30 04204 034241  
31 04205 136510  
32 04206 176010  
33 04207 125004  
34 04208 100774  
35 04211 024240  
36 04212 101002  
37 04213 124014  
38  
39  
40 04220 006231  
41  
42  
43

10059 ECL20

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49

BTR:

SETUP 20  
JSR #ENTIN

TEST "BTO"  
INITIALIZE TEST....  
SHOULD SET BIT 15  
OF WORD TEM

04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43

04 04073 006230  
05 04074 000020  
06 04075 102400  
07 04076 040240  
08 04077 176820  
09 04100 024032  
10 04101 126010  
11 04102 030240  
12 04103 101003  
13 04104 156404  
14  
15  
16 04111 006231  
17  
18  
19 04112 006230  
20 04113 000020  
21 04114 102400  
22 04115 040240  
23 04116 176820  
24 04117 030024  
25 04120 152010  
26 04121 024240  
27 04122 101003  
28 04123 136414  
29  
30  
31 04130 006231  
32  
33 04131 006230  
34 04132 000040  
35 04133 176820  
36 04134 102400  
37 04135 020023  
38 04136 020023  
39 04140 102010  
40 04141 175124  
41 04142 080775  
42 04143 024240  
43 04144 124014  
44  
45  
46 04151 006231  
47  
48  
49

10060 ECL20

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43

B11:

SETUP 20  
JSR #ENTIN

TEST "BTO"  
INITIALIZE TEST....  
SHOULD SET BIT 15  
OF WORD TEM

04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43

04 04073 006230  
05 04074 000020  
06 04075 102400  
07 04076 040240  
08 04077 176820  
09 04100 024032  
10 04101 126010  
11 04102 030240  
12 04103 101003  
13 04104 156404  
14  
15  
16 04111 006231  
17  
18  
19 04112 006230  
20 04113 000020  
21 04114 102400  
22 04115 040240  
23 04116 176820  
24 04117 030024  
25 04120 152010  
26 04121 024240  
27 04122 101003  
28 04123 136414  
29  
30  
31 04130 006231  
32  
33 04131 006230  
34 04132 000040  
35 04133 176820  
36 04134 102400  
37 04135 020023  
38 04136 020023  
39 04140 102010  
40 04141 175124  
41 04142 080775  
42 04143 024240  
43 04144 124014  
44  
45  
46 04151 006231  
47  
48  
49

10059 ECL20

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49

BTR:

SETUP 20  
JSR #ENTIN

TEST "BTO"  
INITIALIZE TEST....  
SHOULD SET BIT 15  
OF WORD TEM

04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43

04 04073 006230  
05 04074 000020  
06 04075 102400  
07 04076 040240  
08 04077 176820  
09 04100 024032  
10 04101 126010  
11 04102 030240  
12 04103 101003  
13 04104 156404  
14  
15  
16 04111 006231  
17  
18  
19 04112 006230  
20 04113 000020  
21 04114 102400  
22 04115 040240  
23 04116 176820  
24 04117 030024  
25 04120 152010  
26 04121 024240  
27 04122 101003  
28 04123 136414  
29  
30  
31 04130 006231  
32  
33 04131 006230  
34 04132 000040  
35 04133 176820  
36 04134 102400  
37 04135 020023  
38 04136 020023  
39 04140 102010  
40 04141 175124  
41 04142 080775  
42 04143 024240  
43 04144 124014  
44  
45  
46 04151 006231  
47  
48  
49

10060 ECL20

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43

B11:

SETUP 20  
JSR #ENTIN

TEST "BTO"  
INITIALIZE TEST....  
SHOULD SET BIT 15  
OF WORD TEM

04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43

04 04073 006230  
05 04074 000020  
06 04075 102400  
07 04076 040240  
08 04077 176820  
09 04100 024032  
10 04101 126010  
11 04102 030240  
12 04103 101003  
13 04104 156404  
14  
15  
16 04111 006231  
17  
18  
19 04112 006230  
20 04113 000020  
21 04114 102400  
22 04115 040240  
23 04116 176820  
24 04117 030024  
25 04120 152010  
26 04121 024240  
27 04122 101003  
28 04123 136414  
29  
30  
31 04130 006231  
32  
33 04131 006230  
34 04132 000040  
35 04133 176820  
36 04134 102400  
37 04135 020023  
38 04136 020023  
39 04140 102010  
40 04141 175124  
41 04142 080775  
42 04143 024240  
43 04144 124014  
44  
45  
46 04151 006231  
47  
48  
49

10061 ECL2P

```

01
02
03
04 04221 006230
05 04222 000004
06 04223 102400
07 04224 024222
08 04225 000245
09 04226 100400
10 04227 044313
11 04228 100400
12 04229 100400
13 04230 010000
14 04231 010000
15 04232 113710
16 04233 034226
17 04234 024313
18 04235 100410
19 04236 100410
20 04237 100410
21 04238 100410
22 04239 034226
23 04240 100410
24 04241 100410
25 04242 000775
26 04243 034226
27 04244 100410
28 04245 100410
29 04246 021400
30 04247 100415
31 04248 000414
32
33
34 04250 101350
35 04251 000250
36 04252 000272
37 04253 000245
38 04254 000246
39 04255 000000
40 04256 175400
41 04257 151404
42 04258 000700
43
44 04267 006231

```

```

BT5:
BT5.1:
BT5.2:
BT5.3:
BT5.4
BT5.5
BT5.6
BT5.7
BT5.8
BT5.9
BT5.10
BT5.11
BT5.12
BT5.13
BT5.14
BT5.15
BT5.16
BT5.17
BT5.18
BT5.19
BT5.20
BT5.21
BT5.22
BT5.23
BT5.24
BT5.25
BT5.26
BT5.27
BT5.28
BT5.29
BT5.30
BT5.31
BT5.32
BT5.33
BT5.34
BT5.35
BT5.36
BT5.37
BT5.38
BT5.39
BT5.40
BT5.41
BT5.42
BT5.43
BT5.44

```

```

SETUP 4.
JSR #ENTLN
4.
SUB #0,0
LDA 1,BEND
LDA 2,BBEG
SUB 2,1
STA 1,BAMSIZ
INC 2,3
STA 0,0,2
DSZ 0,2
BAH
LDA 2,BBEG
LDA 0,BAMSIZ
HXL 1,0
SBI 1,0
RTO 2,0
MOV #0,0,SZR
JMP 1,3
LDA 2,BBEG
LDA 0,BAMSIZ
HXL 1,2
HXR 1,2
NEG 2,2
COM# 0,0,SNR
JMP #0,0,SNR
LDA 1,BBEG
SUB 3,1
NEG 1,1
BAMER
#BT0H1
AC3
M1
AC0
AC1
0
INC 3,3
INC 2,2,SZR
JMP #BT6.2
LOOP
JSR #ENTLN

```

```

BT6:
BT6.1:
BT6.2:
BT6.3:
BT6.4
BT6.5
BT6.6
BT6.7
BT6.8
BT6.9
BT6.10
BT6.11
BT6.12
BT6.13
BT6.14
BT6.15
BT6.16
BT6.17
BT6.18
BT6.19
BT6.20
BT6.21
BT6.22
BT6.23
BT6.24
BT6.25
BT6.26
BT6.27
BT6.28
BT6.29
BT6.30
BT6.31
BT6.32
BT6.33
BT6.34
BT6.35
BT6.36
BT6.37
BT6.38
BT6.39
BT6.40
BT6.41
BT6.42
BT6.43
BT6.44

```

```

BT6.1:
BT6.2:
BT6.3:
BT6.4
BT6.5
BT6.6
BT6.7
BT6.8
BT6.9
BT6.10
BT6.11
BT6.12
BT6.13
BT6.14
BT6.15
BT6.16
BT6.17
BT6.18
BT6.19
BT6.20
BT6.21
BT6.22
BT6.23
BT6.24
BT6.25
BT6.26
BT6.27
BT6.28
BT6.29
BT6.30
BT6.31
BT6.32
BT6.33
BT6.34
BT6.35
BT6.36
BT6.37
BT6.38
BT6.39
BT6.40
BT6.41
BT6.42
BT6.43
BT6.44

```

10062 ECL20

```

01
02
03
04 04270 006230
05 04271 000004
06 04272 102400
07 04273 024227
08 04274 030226
09 04275 100400
10 04276 044313
11 04277 100400
12 04278 010000
13 04279 010000
14 04280 113710
15 04281 030226
16 04282 020313
17 04283 101410
18 04284 100410
19 04285 100410
20 04286 100410
21 04287 000775
22 04288 034226
23 04289 030313
24 04290 111410
25 04291 111510
26 04292 150400
27 04293 021400
28 04294 100415
29 04295 000414
30 04296 024226
31 04297 100400
32 04298 124400
33
34 04327 101350
35 04328 000250
36 04329 000272
37 04330 000245
38 04331 000246
39 04332 000000
40 04333 175400
41 04334 151404
42 04335 000700
43
44 04340 006231

```

```

TEST "BT0"
INITIALIZE TEST....
ISET MEMORY TO ALL
ONES. THEN USE BT0 TO
ISET EACH BIT OF BUFFER
IAGAIN TO ONES.
ISET MEMORY TO ALL
ONES.
ISTART AT TOP AND
ITERATE FOR EACH BIT.
ICHECK THE BUFFER FOR
ONES.
ISET TO FAIL TO SET
ONE OR MORE BITS OF MEMORY.
IWORD*20+FAILED BITS
IBIT COUNT
I"BT0 ADDR GOOD BAD WORD"
ITERATE TEST ROUTINE....

```

```

SETUP 4.
JSR #ENTLN
4.
SUB #0,0
LDA 1,BEND
LDA 2,BBEG
SUB 2,1
STA 1,BAMSIZ
INC 2,3
STA 0,0,2
DSZ 0,2
BAH
LDA 2,BBEG
LDA 0,BAMSIZ
HXL 1,0
SBI 1,0
RTO 2,0
MOV #0,0,SZR
JMP 1,3
LDA 2,BBEG
LDA 0,BAMSIZ
HXL 1,2
HXR 1,2
NEG 2,2
COM# 0,0,SNR
JMP #0,0,SNR
LDA 1,BBEG
SUB 3,1
NEG 1,1
BAMER
#BT0H1
AC3
M1
AC0
AC1
0
INC 3,3
INC 2,2,SZR
JMP #BT6.2
LOOP
JSR #ENTLN

```

```

BT6:
BT6.1:
BT6.2:
BT6.3:
BT6.4
BT6.5
BT6.6
BT6.7
BT6.8
BT6.9
BT6.10
BT6.11
BT6.12
BT6.13
BT6.14
BT6.15
BT6.16
BT6.17
BT6.18
BT6.19
BT6.20
BT6.21
BT6.22
BT6.23
BT6.24
BT6.25
BT6.26
BT6.27
BT6.28
BT6.29
BT6.30
BT6.31
BT6.32
BT6.33
BT6.34
BT6.35
BT6.36
BT6.37
BT6.38
BT6.39
BT6.40
BT6.41
BT6.42
BT6.43
BT6.44

```

```

BT6.1:
BT6.2:
BT6.3:
BT6.4
BT6.5
BT6.6
BT6.7
BT6.8
BT6.9
BT6.10
BT6.11
BT6.12
BT6.13
BT6.14
BT6.15
BT6.16
BT6.17
BT6.18
BT6.19
BT6.20
BT6.21
BT6.22
BT6.23
BT6.24
BT6.25
BT6.26
BT6.27
BT6.28
BT6.29
BT6.30
BT6.31
BT6.32
BT6.33
BT6.34
BT6.35
BT6.36
BT6.37
BT6.38
BT6.39
BT6.40
BT6.41
BT6.42
BT6.43
BT6.44

```

```

BT6.1:
BT6.2:
BT6.3:
BT6.4
BT6.5
BT6.6
BT6.7
BT6.8
BT6.9
BT6.10
BT6.11
BT6.12
BT6.13
BT6.14
BT6.15
BT6.16
BT6.17
BT6.18
BT6.19
BT6.20
BT6.21
BT6.22
BT6.23
BT6.24
BT6.25
BT6.26
BT6.27
BT6.28
BT6.29
BT6.30
BT6.31
BT6.32
BT6.33
BT6.34
BT6.35
BT6.36
BT6.37
BT6.38
BT6.39
BT6.40
BT6.41
BT6.42
BT6.43
BT6.44

```

```

BT6.1:
BT6.2:
BT6.3:
BT6.4
BT6.5
BT6.6
BT6.7
BT6.8
BT6.9
BT6.10
BT6.11
BT6.12
BT6.13
BT6.14
BT6.15
BT6.16
BT6.17
BT6.18
BT6.19
BT6.20
BT6.21
BT6.22
BT6.23
BT6.24
BT6.25
BT6.26
BT6.27
BT6.28
BT6.29
BT6.30
BT6.31
BT6.32
BT6.33
BT6.34
BT6.35
BT6.36
BT6.37
BT6.38
BT6.39
BT6.40
BT6.41
BT6.42
BT6.43
BT6.44

```

```

BT6.1:
BT6.2:
BT6.3:
BT6.4
BT6.5
BT6.6
BT6.7
BT6.8
BT6.9
BT6.10
BT6.11
BT6.12
BT6.13
BT6.14
BT6.15
BT6.16
BT6.17
BT6.18
BT6.19
BT6.20
BT6.21
BT6.22
BT6.23
BT6.24
BT6.25
BT6.26
BT6.27
BT6.28
BT6.29
BT6.30
BT6.31
BT6.32
BT6.33
BT6.34
BT6.35
BT6.36
BT6.37
BT6.38
BT6.39
BT6.40
BT6.41
BT6.42
BT6.43
BT6.44

```

10063 ECL20

01  
02  
03

BTO27:

```

04 04331 006230  SETUP 10
05 04332 000010  JSR #ENTIN
06 04333 030241  LDA 2,00
07 04334 126000  AUC 1,1
08 04335 042400  STA 1,1,TEM
09 04336 125400  INC 1,1
10 04337 132110  BIT 1/2
11 04338 034240  LDA 3,TEM
12 04339 102520  ADCR 0,0
13 04340 116414  SUB# 0,3,SZR
14 ERROR
15 JSR #ENTLO
16 04357 006231  ITERATE TEST ROUTINE.....
17
18 04358 006230  FTEST "BTZ"
19 04359 000010  INITIALIZE TEST.....
20 04360 030241  LDA 2,00
21 04361 126000  AUC 1,1
22 04362 042400  STA 1,1,TEM
23 04363 125400  INC 1,1
24 04364 132110  BIT 1/2
25 04365 034240  LDA 3,TEM
26 04366 102520  ADCR 0,0
27 04367 116414  SUB# 0,3,SZR
28 04371 174814  ERROR
29 LOOP
30 JSR #ENTLO
31 04376 006231  ITERATE TEST ROUTINE.....
32
33
34 04377 006230  FTEST "BTZ"
35 04378 000010  INITIALIZE TEST.....
36 04379 030241  LDA 2,TEM
37 04380 126400  SUB 1,1
38 04381 042400  STA 1,1,TEM
39 04382 146010  LDA 2,1,TEM
40 04383 034240  LDA 3,TEM
41 04384 102520  SUB# 0,3,SZR
42 04385 162414  ERROR
43 LOOP
44 JSR #ENTLO
45 04414 006231  ITERATE TEST ROUTINE.....
46
47

```

10064 ECL20

01  
02  
03

BTZ1:

```

04 04415 006230  SETUP 20
05 04416 030240  JSR #ENTIN
06 04417 102000  AUC 0,0
07 04418 042000  STA 0,TEM
08 04419 176200  ADCR 3,3
09 04420 02421  LDA 1,00
10 04421 129110  BIT 1/1
11 04422 030240  LDA 2,TEM
12 04423 101002  MOV 0,0,SIC
13 04424 156414  SUB# 2,3,SZR
14 ERROR
15 JSR #ENTLO
16 04433 006231  ITERATE TEST ROUTINE.....
17
18 04434 006230  FTEST "BTZ"
19 04435 000020  INITIALIZE TEST.....
20 04436 102000  AUC 0,0
21 04437 042000  STA 0,TEM
22 04438 176120  ADCR 3,3
23 04439 030242  LDA 2,0,TEM
24 04440 152110  BIT 2,2
25 04441 024240  LDA 1,TEM
26 04442 101002  MOV 0,0,SIC
27 04443 136414  SUB# 1,3,SZR
28 ERROR
29 LOOP
30 JSR #ENTLO
31 04452 006231  ITERATE TEST ROUTINE.....
32
33
34 04453 006230  FTEST "BTZ"
35 04454 000040  INITIALIZE TEST.....
36 04455 176200  SUBZL 3,3
37 04456 102000  ADC 0,0
38 04457 042000  STA 0,TEM
39 04458 020243  LDA 0,0,R
40 04459 101002  INC 0,R
41 04460 152110  RTZ 0,R
42 04461 175124  MOVZL 3,3,SZR
43 04462 003775  JMP 0,-3
44
45 04463 024240  LDA 1,TEM
46 04464 125004  MOV 1,1,SZR
47 ERROR
48 LOOP
49 04473 006231  ITERATE TEST ROUTINE.....
50

```

10065 ECL20

01  
02  
03

BTO28:

```

04 04531 006230  SETUP 10
05 04532 000010  JSR #ENTIN
06 04533 030241  LDA 2,00
07 04534 126000  AUC 1,1
08 04535 042400  STA 1,1,TEM
09 04536 125400  INC 1,1
10 04537 132110  BIT 1/2
11 04538 034240  LDA 3,TEM
12 04539 102520  ADCR 0,0
13 04540 116414  SUB# 0,3,SZR
14 ERROR
15 JSR #ENTLO
16 04557 006231  ITERATE TEST ROUTINE.....
17
18 04558 006230  FTEST "BTZ"
19 04559 000010  INITIALIZE TEST.....
20 04560 030241  LDA 2,00
21 04561 126000  AUC 1,1
22 04562 042400  STA 1,1,TEM
23 04563 125400  INC 1,1
24 04564 132110  BIT 1/2
25 04565 034240  LDA 3,TEM
26 04566 102520  ADCR 0,0
27 04567 116414  SUB# 0,3,SZR
28 04571 174814  ERROR
29 LOOP
30 JSR #ENTLO
31 04576 006231  ITERATE TEST ROUTINE.....
32
33
34 04577 006230  FTEST "BTZ"
35 04578 000010  INITIALIZE TEST.....
36 04579 030241  LDA 2,TEM
37 04580 126400  SUB 1,1
38 04581 042400  STA 1,1,TEM
39 04582 146010  LDA 2,1,TEM
40 04583 034240  LDA 3,TEM
41 04584 102520  SUB# 0,3,SZR
42 04585 162414  ERROR
43 LOOP
44 JSR #ENTLO
45 04414 006231  ITERATE TEST ROUTINE.....
46
47

```

10066 ECL20

01  
02  
03

BTO29:

```

04 04631 006230  SETUP 8
05 04632 000010  JSR #ENTIN
06 04633 030241  LDA 2,TEM
07 04634 126400  SUB 1,1
08 04635 042400  STA 1,1,TEM
09 04636 146010  LDA 2,1,TEM
10 04637 034240  LDA 3,TEM
11 04638 102520  SUB# 0,3,SZR
12 04639 162414  ERROR
13 LOOP
14 JSR #ENTLO
15 04657 006231  ITERATE TEST ROUTINE.....
16
17
18 04658 006230  FTEST "BTZ"
19 04659 000040  INITIALIZE TEST.....
20 04660 030241  LDA 2,TEM
21 04661 126400  SUBZL 3,3
22 04662 102000  ADC 0,0
23 04663 042000  STA 0,TEM
24 04664 176120  ADCR 3,3
25 04665 030242  LDA 2,0,TEM
26 04666 152110  BIT 2,2
27 04667 024240  LDA 1,TEM
28 04668 101002  MOV 0,0,SIC
29 04669 136414  SUB# 1,3,SZR
30 ERROR
31 LOOP
32 JSR #ENTLO
33 04678 006231  ITERATE TEST ROUTINE.....
34
35
36 04679 006230  FTEST "BTZ"
37 04680 000040  INITIALIZE TEST.....
38 04681 176200  SUBZL 3,3
39 04682 102000  ADC 0,0
40 04683 042000  STA 0,TEM
41 04684 020243  LDA 0,0,R
42 04685 101002  INC 0,R
43 04686 152110  RTZ 0,R
44 04687 175124  MOVZL 3,3,SZR
45 04688 003775  JMP 0,-3
46
47 04689 024240  LDA 1,TEM
48 04690 125004  MOV 1,1,SZR
49 ERROR
50 LOOP
51 04699 006231  ITERATE TEST ROUTINE.....
52

```

10065 ECL20

```

01
02
03
04 04474 000230
05 04475 000040
06
07 04476 006234
08 04477 103045
09 04478 000776
10 04501 000250
11 04502 034241
12 04503 105110
13 04504 176110
14 04505 101004
15 04506 000774
16 04507 030240
17 04510 101002
18 04511 101014
19
20
21 04516 006231
22
23
24 04517 006230
25 04520 000040
26
27 04521 006234
28 04522 100000
29 04523 103045
30 04524 000775
31 04525 034241
32 04527 136510
33 04530 176110
34 04531 125004
35 04532 000774
36 04533 024240
37 04534 101002
38 04535 100414
39
40
41
42 04542 006231

```

```

10066 ECL20
01
02
03 04043 005230
04 04044 000040
05 04045 102400
06 04046 004227
07 04047 000256
08 04050 100400
09 04051 040313
10 04052 100400
11 04053 041000
12 04054 113710
13 04055 034226
14 04056 024313
15 04057 100410
16 04060 100410
17 04061 100210
18 04062 100400
19 04063 102414
20 04064 000775
21 04065 034226
22 04066 111510
23 04067 100400
24 04070 021400
25 04071 100015
26 04072 000414
27
28
29
30 04073 024226
31 04074 100400
32 04075 124400
33
34 04080 101350
35 04081 000250
36 04082 000272
37 04083 000245
38 04084 000246
39 04085 000000
40 04086 175400
41 04087 151404
42 04010 000750
43
44 04011 000231

```

```

/TEST "BTZ"
/INITIALIZE TEST....
/IA RANDOM NUMBER IS STORED
/IN C(TEM), FOR EACH (I) BIT
/IN THE WORD A "BTZ" POINTER
/IS GENERATED AND A "BTZ"
/EXCHANGED TO ALL TEMS BY
/THE SEVERAL "BTZ" INSTRUCTIONS
/ WILL ITERATE FOR EACH (I) BIT

/IC(AC2)=BTO RESULT
/IC(AC1)=ORIGINAL C(TEM)
/IC(CARRY) SHOULD REMAIN (I)

/ITERATE TEST ROUTINE....

/TEST "BTZ"
/INITIALIZE TEST....
/IA RANDOM NUMBER IS STORED
/IN C(TEM), FOR EACH (0) BIT
/IN THE NUMBER, A POINTER IS
/GENERATED AND A "BTZ" INST-
/RUCTION EXECUTED. THUS AFTER
/THE SEVERAL BTZ INSTRUCTIONS
/ THE WORD IN MEMORY SHOULD
/ NOT BE CHANGED.
/ITERATE FOR EACH 0 BIT IN C(TEM)

/IC(AC1)=BTZ RESULT
/IC(AC0)=ORIGINAL AND CORRECT
/IC(CARRY) SHOULD REMAIN (I)

/ITERATE TEST ROUTINE....

```

```

SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
MOV 0,1,SNR
JMP BTZ5
STA 0,TEM
LDA 3,B0
RTZ 3,3
MOV 0,0,JSZR
JMP #*4
LDA 2,TEM
MOV 0,0,JSZC
MOV #2,JSZR
ERROR
LOOP
JSR #ENTLU

SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
MOV 0,1,SNR
JMP BTZ4
STA 0,TEM
LDA 3,B0
RTZ 3,3
MOV 1,1,JSZR
JMP #*4
LDA 1,TEM
MOV 0,0,JSZC
SUB# 0,1,JSZC
ERROR
LOOP
JSR #ENTLU

```

```

/TEST "BTO"
/INITIALIZE TEST....

SETUP 4.
JSR #ENTIN
4.
SUB 0,0
LDA 1,BAND
LDA 2,BBEG
SUB 2,1
STA 1,BANDSIZ
INC 2,3
RTA 0,0,2
RAM
LDA 3,BBEG
LDA 1,BANDSIZ
MUL 1,1
DXYR 4,1
RTD 3,1
INC 1,1
SUB# 1,0,JSZK
JMP #*3
LDA 3,BBEG
HXR 1,2
NEG 2,2
LDA 0,0,3
COM# 0,0,SNR
JMP BTZ54

LDA 1,BBEG
SUB 3,1
NEG 1,1
BAKER
#BTOM1
ACS
M1
AC0
AC1
0
INC 3,3
INC 2,2,JSZR
JMP BTZ53
LOOP
JSK #ENTLU
/ITERATE TEST ROUTINE....

```

```

/TEST "BTO"
/INITIALIZE TEST....

/IBTO FAILED TO SET ONE
/ FOR MORE BITS OF MEMORY.
/ IWORD *20* FAILED BIT TIME
/ BIT COUNT IN C(CAO)
/ "BTO ADDR GOOD BAD WORD

/ITERATE TEST ROUTINE....

```

```

/TEST "BTO"
/INITIALIZE TEST....

/ITERATE TEST ROUTINE....

```









```

10073 ECL20
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
STA 0,TEM
SUBZL 3,3
LOA 1,80
SZBO 1,1
MOV 1,1
INC 1,1
MVL 3,3,SNC
JMP SZ21
LOA 2,TEM
COM# 2,2,SNK
SUB# 0,0,SZK
ERROR
LOOP
JSR #ENTLU

ITERATE TEST ROUTINE,....

TEST "SZBO"
INITIALIZE TEST,....

IC(AC0)=RANDOM #
ISTORE RANDOM # IN C(TEM)
IPRINT TO LOCATION TEM AND
IUSE SZBO TO ASSEMBLE A
ILIKE WORD IN C(AC3).

IFOR EACH (1) BIT IN
IRANDOM,EXECUTE A
ISZBO INSTRUCTION TO
ITHE SAME BIT IN LOC
ITEM".

IFAIL TO SKIP,....

IC(AC1)=SZBO RESULT IN MEMORY,
IC(AC2)=ORIGINAL NUMBER,

ITERATE TEST ROUTINE,....

SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
STA 0,TEM
SUBZL 3,3
LOA 1,80
SZBO 1,1
MOV 1,1
INC 1,1
MVL 3,3,SNC
JMP SZ21
LOA 2,TEM
COM# 2,2,SNK
SUB# 0,0,SZK
ERROR
LOOP
JSR #ENTLU

ITERATE TEST ROUTINE,....

TEST "SZBO"
INITIALIZE TEST,....

IC(AC0)=RANDOM #
ISTORE RANDOM # IN FIRST
IBUFFER LOCATION, C(AC1)
IWILL PRINT AT THIS WORD
IFINDIRECTLY, C(AC2) WILL SCAN
IFEACH BIT POSITION, ASSEMBLE
ITHE NUMBER IN C(AC3) BY
ISZBO".
ITERATE 16, TIMES.

INORD IN MEMORY SHOULD
INDW BE ALL ONES,
IC(AC2)=SZBO RESULT IN MEMORY,
IC(AC3)=SZBO ASSEMBLY,

ITERATE TEST ROUTINE,....

```

```

10074 ECL20
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

```

```

SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
STA 0,TEM
SUBZL 3,3
LOA 1,80
SZBO 1,1
MOV 1,1
INC 1,1
MVL 3,3,SNC
JMP SZ21
LOA 2,TEM
COM# 2,2,SNK
SUB# 0,0,SZK
ERROR
LOOP
JSR #ENTLU

ITERATE TEST ROUTINE,....

TEST "SZBO"
INITIALIZE TEST,....

IC(AC0)=RANDOM #
ISTORE RANDOM # IN FIRST
IBUFFER LOCATION, C(AC1)
IWILL PRINT AT THIS WORD
IFINDIRECTLY, C(AC2) WILL SCAN
IFEACH BIT POSITION, ASSEMBLE
ITHE NUMBER IN C(AC3) BY
ISZBO".
ITERATE 16, TIMES.

INORD IN MEMORY SHOULD
INDW BE ALL ONES,
IC(AC2)=SZBO RESULT IN MEMORY,
IC(AC3)=SZBO ASSEMBLY,

ITERATE TEST ROUTINE,....

```

```

SZ21

```

```

SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
STA 0,TEM
SUBZL 3,3
LOA 1,80
SZBO 1,1
MOV 1,1
INC 1,1
MVL 3,3,SNC
JMP SZ21
LOA 2,TEM
COM# 2,2,SNK
SUB# 0,0,SZK
ERROR
LOOP
JSR #ENTLU

ITERATE TEST ROUTINE,....

```

```

TEST "SZBO"
INITIALIZE TEST,....

```

```

IC(AC0)=RANDOM #
ISTORE RANDOM # IN C(TEM)
IPRINT TO LOCATION TEM AND
IUSE SZBO TO ASSEMBLE A
ILIKE WORD IN C(AC3).

IFOR EACH (1) BIT IN
IRANDOM,EXECUTE A
ISZBO INSTRUCTION TO
ITHE SAME BIT IN LOC
ITEM".

IFAIL TO SKIP,....

IC(AC1)=SZBO RESULT IN MEMORY,
IC(AC2)=ORIGINAL NUMBER,

ITERATE TEST ROUTINE,....

```

```

SZ31

```

```

SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
STA 0,TEM
SUBZL 3,3
LOA 1,80
SZBO 1,1
MOV 1,1
INC 1,1
MVL 3,3,SNC
JMP SZ31
LOA 2,TEM
COM# 2,2,SNK
SUB# 0,0,SZK
ERROR
LOOP
JSR #ENTLU

ITERATE TEST ROUTINE,....

TEST "SZBO"
INITIALIZE TEST,....

IC(AC0)=RANDOM #
ISTORE RANDOM # IN FIRST
IBUFFER LOCATION, C(AC1)
IWILL PRINT AT THIS WORD
IFINDIRECTLY, C(AC2) WILL SCAN
IFEACH BIT POSITION, ASSEMBLE
ITHE NUMBER IN C(AC3) BY
ISZBO".
ITERATE 16, TIMES.

INORD IN MEMORY SHOULD
INDW BE ALL ONES,
IC(AC2)=SZBO RESULT IN MEMORY,
IC(AC3)=SZBO ASSEMBLY,

ITERATE TEST ROUTINE,....

```

```

SETUP 40
JSR #ENTIN
40
RAND
JSR #ENTRA
STA 0,TEM
SUBZL 3,3
LOA 1,80
SZBO 1,1
MOV 1,1
INC 1,1
MVL 3,3,SNC
JMP SZ31
LOA 2,TEM
COM# 2,2,SNK
SUB# 0,0,SZK
ERROR
LOOP
JSR #ENTLU

ITERATE TEST ROUTINE,....

```

```

TEST "SZBO"
INITIALIZE TEST,....

```

```

IC(AC0)=RANDOM #
ISTORE RANDOM # IN C(TEM)
IPRINT TO LOCATION TEM AND
IUSE SZBO TO ASSEMBLE A
ILIKE WORD IN C(AC3).

IFOR EACH (1) BIT IN
IRANDOM,EXECUTE A
ISZBO INSTRUCTION TO
ITHE SAME BIT IN LOC
ITEM".

IFAIL TO SKIP,....

IC(AC1)=SZBO RESULT IN MEMORY,
IC(AC2)=ORIGINAL NUMBER,

ITERATE TEST ROUTINE,....

```

```

10075 ECL20
01
02
03 05344 006230
04 05345 006040
05
06 05346 006234
07 05347 022226
08 05350 024044
09 05351 050017
10 05352 176060
11 05353 104110
12 05354 146310
13 05355 175060
14 05356 175203
15 05357 000774
16 05360 026226
17 05361 162415
18 05362 124014
19
20
21 05367 006231
22 05370 000405
23 05371 105372
24 05372 105373
25 05373 105374
26 05374 100226
27

10076 ECL20
01
02 05375 006230
03 05376 000010
04 05377 024227
05 05400 006226
06 05401 146400
07 05402 105400
08 05403 102400
09 05404 041000
10 05405 113710
11 05406 030226
12 05407 024227
13 05410 034044-SZ60:
14 05411 141110 SZ61:
15 05412 156310
16 05413 000415
17 05414 175004
18 05415 000774
19 05416 035000
20 05417 174014
21
22 05424 151400
23 05425 146414
24 05426 000702
25 05427 101001
26
27
28 05434 006231
29
30
31 05435 006230
32 05436 000001
33 05437 024227
34 05440 030226
35 05441 146400
36 05442 044240
37 05443 050273
38
39 05444 006234
40 05445 111000
41 05446 101120
42 05447 101220
43 05450 024240
44 05451 122422
45 05452 000777
46 05453 123000
47 05454 151112
48 05455 103240
49 05456 024226
50 05457 123000
51 05460 042273
52 05461 010273
53 05462 020273
54 05463 024227
55 05464 100414
56 05465 000757
57
58 05466 006231

```

```

I TEST "SZ00"
I INITIALIZE TEST.....
I CLEAR C(BUFFER)

I WITH A CLEARED BUFFER,
I SET IT ALL ONES, A BIT
I AT A TIME.

I FAIL TO SKIP

I C(AC3)=WORD FROM MEMORY.
I C(AC2)=ADDRESS OF THE WORD.
I C(AC1)=END OF BUFFER.

I C(AC2)=WORD POINTER.
I C(AC3)=BIT POINTER.
I "SZ00" FAILED TO SKIP.

I ITERATE TEST ROUTINE.....
I FOR THE SEVERAL TEST
I INITIALIZE TEST....

I TO FOLLOW:
I FILL EACH WORD OF THE
I BUFFER WITH A PSEUDO
I RANDOM ADDRESS. THESE
I POINTERS ARE ALL WITHIN
I THE RANGE OF THE BUFFER
I C(AC0)=RANDOM #
I AND MAY BE INDIRECT.

I C(AC0) 1=15=RANDOM
I C(AC1)=BUFFER SIZE.
I MAKE C(AC0) MODULO
I BUFFER SIZE.

I RANDOMLY SET THE
I INDIRECT BIT.

I STORE WORD IN BUFFER.
I POINT TO NEXT WORD.
I TEST FOR LAST.

I ITERATE
I EXIT
I ITERATE TEST ROUTINE.....

```

```

SETUP 8.
JSR #ENTLN
8.
LOA 1,BEG
LOA 2,BEG
SUB 2,1
INC 2,3
SUB 0,0
STA 0,0,2
RAM
LOA 2,BEG
LOA 1,BEG
LOA 3,END
SCL 1,3
SZ00 2,3
MOV 3,3,SZR
JMP SZ61
LOA 3,0,2
COM# 3,3,SZK
EVRD
INC 2,2
SUB# 2,1,SZK
JMP SZ60
MOV 0,0,SRP
ERROR
LOOP
JSR #ENTLN

SETUP 1
JSR #ENTLN
1
LOA 1,BEG
LOA 2,BEG
SUB 2,1
STA 1,TEM
STA 2,TEM1
RAND
JSR #ENTRA
MOV 0,2
MOVZL 0,0
MOVZB 0,0
LOA 1,TEM
SUBZ 1,0,SZC
JMP -1
ADD 1,0
MOVZL# 2,2,SZC
ADDR 0,0
LOA 1,BEG
ADD 1,0
STA 0,TEM1
ISZ TEM1
LOA 0,TEM1
SUB# 0,1,SZK
JMP SZ71
LOOP
JSR #ENTLN

SETUP 1
JSR #ENTLN
1
LOA 1,BEG
LOA 2,BEG
SUB 2,1
STA 1,TEM
STA 2,TEM1
RAND
JSR #ENTRA
MOV 0,2
MOVZL 0,0
MOVZB 0,0
LOA 1,TEM
SUBZ 1,0,SZC
JMP -1
ADD 1,0
MOVZL# 2,2,SZC
ADDR 0,0
LOA 1,BEG
ADD 1,0
STA 0,TEM1
ISZ TEM1
LOA 0,TEM1
SUB# 0,1,SZK
JMP SZ71
LOOP
JSR #ENTLN

SETUP 1
JSR #ENTLN
1
LOA 1,BEG
LOA 2,BEG
SUB 2,1
STA 1,TEM
STA 2,TEM1
RAND
JSR #ENTRA
MOV 0,2
MOVZL 0,0
MOVZB 0,0
LOA 1,TEM
SUBZ 1,0,SZC
JMP -1
ADD 1,0
MOVZL# 2,2,SZC
ADDR 0,0
LOA 1,BEG
ADD 1,0
STA 0,TEM1
ISZ TEM1
LOA 0,TEM1
SUB# 0,1,SZK
JMP SZ71
LOOP
JSR #ENTLN

```

```

I TEST "SZ00"
I INITIALIZE TEST.....
I C(AC0)=RANDOM #
I THIS TEST IS SIMILAR TO
I THE PREVIOUS TEST
I WITH THE EXCEPTION OF
I MULTI LEVEL DEFER.

I C(AC0)=ORIGINAL RANDOM
I C(AC1)#

I ITERATE TEST ROUTINE.....
I POINTS TO FIRST WORD
I IN THE DATA BUFFER.

```

```

SETUP 40
JSR #ENTLN
40
RAND
JSR #ENTRA
STA 0,0,BEG
LOA 1,END
LOA 2,END
SUBZ 3,3
SBI 1,1
SZ00 2,1
MOV 3,3
MOVH 3,3,SHC
JMP #4
LOA 1,0,BEG
SUB# 3,0,SHK
COM# 1,1,SZK
ERROR
LOOP
JSR #ENTLN
JMP SZ6
#-1
#-1
#-1
0BEG

```

```

10077 ECL2A
01
02
03
04 05467 020226
05 05470 040240 SZR:
06 05471 006230
07 05472 006005
08 05473 030240
09
10 05474 030444
11 05475 030000
12 05476 170500
13 05477 170500
14 05500 000431
15 05501 031000
16 05502 032000
17 05503 151132
18 05504 030773
19 05505 035000
20 05506 030000
21 05507 024044
22 05510 040000
23 05511 020000
24 05512 125120
25 05513 125200
26 05514 132414
27 05515 136415
28 05516 000433
29 05517 020000
30 05520 100414
31 05521 000774
32 05522 034273
33 05523 030274
34 05524 020240
35 05525 100240
36 05526 024044
37 05527 104110
38 05530 106310
39 05531 101000
40 05532 131200
41 05533 130000
42 05534 030773
43 05535 020274
44 05536 120400
45 05537 024274
46 05540 120000
47 05541 100414
48
49 05546 020273
50 05547 022274
51
52 05550 000231 SZR:
53 05551 020240
54 05552 101400
55 05553 030227
56 05554 112434
57 05555 000713

LVA 0,0,BEG
STA 0,TEM
SETUP 5
JSR #ENTIN
LVA 2,TEM
LVA 2,TEM
LVA 2,TEM
STA 2,20
SUBZL 3,3
INCL 3,3, SZC
JMP SZR1
LVA 2,0,2
LVA 2,0,2
MOVZL 2,2, SZC
JMP *-5
LVA 3,0,2
LVA 0,20
LVA 1,20
LVA 1,20
LVA 1,000
MOVZL 1,1
MOVZL 1,1
MOVZL 1,1
SUB# 1,0, SZK
SUB# 1,0, SZK
SUB# 1,0, SZK
JMP SZR1
LVA 1,20
SUB# 0,1, SZR
JMP *-10
STA 3,TEM1
STA 2,TEM2
LVA 0,TEM
AODD 0,0
LVA 1,20
SBI 1,1
SZD 0,1
MOVE 0,0
MOVZ 2,2
MOVZ 1,1, SZR
JMP *-5
LVA 1,TEM2
INCL 1,1
LVA 1,TEM2
MOV 1,1, SZC
SUB# 2,3, SZK
ERROR
LVA 0,TEM1
STA 0,TEM2
LOOP
JSR #ENTLO
LVA 0,TEM
INC 0,0
LVA 2, BEND
SUBZ 0,2, SZR
JMP SZR-1

MULTI LEVEL INDIRECT
ADDRESSING OF "SZ00"
INITIALIZE TEST,....
IPOINT INTO BUFFER
IFIND THE FINAL ADDRESS
IFAND VALUE IN THE INDIRECT
IFCHAIN, C(AC2) IS FIRST
IFADDRESS AT WHICH TO
ISTART LOOKING.
IFIND INDIRECT CHAIN IS
IF15, EXIT TO SZR1.
IAUTO INDEX REGISTERS.
IFARE USED AS VARIABLES.
IFMAKE SURE THAT FINAL
IFDATA WILL NOT POINT
IFTO ANY SPOT IN THE CHAIN.
IFIF IT OSES, EXIT TO SZR1.
IFC(AC2)=FINAL ADDRESS
IFC(AC3)=DATA AT THAT
IFADDRESS.
IFVALUE OF WORD SELECTED.
IFADDRESS OF THIS WORD.
IFMAKE INDIRECT POINTER
IFAND RESET C(CARRY).
IFASSEMBLE A WORD IN
IFC(AC2) VIA SKIP ON
IF"SZ00" INSTRUCTION.
IFC(CARRY)=1 IF C(MEMORY)=17777
IFC(AC0)=INITIAL POINTER
IFC(AC1)=FINAL ADDRESS
IFC(AC2)=SZ00 ASSEMBLY
IFC(AC3)=ORIGINAL/CORRECT
IFRESTORE WORD IN MEMORY.
IFITERATE TEST ROUTINE,....
IFADVANCE TO NEXT.

10078 ECL20
01
02
03
04 05558 000230
05 05557 000100
06 05560 034050
07 05561 152400
08 05562 050240
09 05563 102620
10 05564 172310
11 05565 000410
12 05566 101400
13 05567 101224
14 05570 000774
15 05571 020240
16 05572 100014
17
18 05577 101001 SZR1:
19
20
21 05604 000231
22
23
24 05605 000230 SZ10:
25 05606 000100
26 05607 034050
27 05610 102040
28 05611 120440
29 05612 104040
30 05613 101400
31 05614 120205
32 05615 000410
33 05616 102310
34 05617 000774
35
36 05624 000407
37 05625 030240 SZ11:
38 05626 150014
39
40
41 05633 000231

SETUP 100
JSR #ENTIN
LVA 3,TEM
STA 2,TEM
SUBZ 0,0
SZ0 3,2
JMP SZR1
INC 2,2
MOVZ 0,0, SZR
JMP *-4
LVA 0,TEM
CQ# 0,0, SZK
ERROR
MOV 0,0, SKP
ERROR
LOOP
JSR #ENTLO

SETUP 100
JSR #ENTIN
LVA 3,TEM
ACCU 0,0
SUBD 1,1
STA 0,TEM
INC 0,0
MOVZ 1,1, SNR
JMP SZ11
SZ0 3,0
JMP *-4
ERROR
JMP SZ12
LVA 2,TEM
CQ# 2,2, SZK
ERROR
LOOP
JSR #ENTLO

TEST "SZ00"
INITIALIZE TEST,....
IC(AC3)=POINTER TO TEM
IPOINT TO BIT 0,1,2 ETC.
IEXIT LOOP
ISHOULD NEVER SKIP.
ITERATE TEST ROUTINE,....

```

```

10079 ECL20
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

10080 ECL20
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

LDB01
04 05634 006230
05 05635 000010
06 05636 102640
07 05637 040240
08 05640 020016-
09 05641 102710
10 05642 024065-
11 05643 122415
12 05644 101003
13
14
15 05651 006231
16
17
18 05652 006230
19 05653 000010
20 05654 102640
21 05655 040240
22 05656 020016-
23 05657 102710
24 05658 101005
25 05659 101003
26
27
28 05666 006231
29
30
31 05667 006230
32 05670 000100
33
34 05671 006234
35 05672 040240
36 05673 024016-
37 05674 132710
38 05675 125420
39 05676 130710
40 05677 145800
41 05700 167003
42 05701 122414
43
44
45 05706 006231

LDB02
04 05707 020226
05 05710 040240
06
07
08 05713 006234
09 05714 042240
10 05715 034240
11 05716 175120
12 05717 165710
13 05720 175400
14 05721 172710
15 05722 125300
16 05723 147003
17 05724 106414
18
19
20 05731 006231
21 05732 010240
22 05733 020240
23 05734 034227
24 05735 116414
25 05736 000753
26

LDB03
04 05711 006230
05 05712 000005
06
07
08 05713 006234
09 05714 042240
10 05715 034240
11 05716 175120
12 05717 165710
13 05720 175400
14 05721 172710
15 05722 125300
16 05723 147003
17 05724 106414
18
19
20 05731 006231
21 05732 010240
22 05733 020240
23 05734 034227
24 05735 116414
25 05736 000753
26

LDB04
04 05711 006230
05 05712 000005
06
07
08 05713 006234
09 05714 042240
10 05715 034240
11 05716 175120
12 05717 165710
13 05720 175400
14 05721 172710
15 05722 125300
16 05723 147003
17 05724 106414
18
19
20 05731 006231
21 05732 010240
22 05733 020240
23 05734 034227
24 05735 116414
25 05736 000753
26

LDB05
04 05711 006230
05 05712 000005
06
07
08 05713 006234
09 05714 042240
10 05715 034240
11 05716 175120
12 05717 165710
13 05720 175400
14 05721 172710
15 05722 125300
16 05723 147003
17 05724 106414
18
19
20 05731 006231
21 05732 010240
22 05733 020240
23 05734 034227
24 05735 116414
25 05736 000753
26

LDB06
04 05711 006230
05 05712 000005
06
07
08 05713 006234
09 05714 042240
10 05715 034240
11 05716 175120
12 05717 165710
13 05720 175400
14 05721 172710
15 05722 125300
16 05723 147003
17 05724 106414
18
19
20 05731 006231
21 05732 010240
22 05733 020240
23 05734 034227
24 05735 116414
25 05736 000753
26

LDB07
04 05711 006230
05 05712 000005
06
07
08 05713 006234
09 05714 042240
10 05715 034240
11 05716 175120
12 05717 165710
13 05720 175400
14 05721 172710
15 05722 125300
16 05723 147003
17 05724 106414
18
19
20 05731 006231
21 05732 010240
22 05733 020240
23 05734 034227
24 05735 116414
25 05736 000753
26

LDB08
04 05711 006230
05 05712 000005
06
07
08 05713 006234
09 05714 042240
10 05715 034240
11 05716 175120
12 05717 165710
13 05720 175400
14 05721 172710
15 05722 125300
16 05723 147003
17 05724 106414
18
19
20 05731 006231
21 05732 010240
22 05733 020240
23 05734 034227
24 05735 116414
25 05736 000753
26

```

10081 ECL20

```

01
02
03
04 05737 006230
05 05740 000200
06
07 05741 006234
08 05742 020015
09 05743 131440
10 05744 120710
11 05745 152710
12 05746 135300
13 05747 187002
14 05750 152414
15
16
17 05755 006231
18
19 05756 020226
20 05757 040240
21
22 05760 006230
23 05761 000005
24
25 05762 006234
26 05763 042240
27 05764 030240
28 05765 151120
29 05766 155400
30 05767 140710
31 05770 170710
32 05771 125300
33 05772 137000
34 05773 020240
35 05774 122415
36 05775 116414
37
38
39 06002 006231
40 06003 010240
41 06004 020240
42 06005 034227
43 06006 116414
44 06007 000751

```

L084:

```

SETUP 200
JSR #ENTIN
200
RAND
JSR #ENTINA
LDA 1,#KANKAN
INCO 1,2
LOR 1,1
LOR 2,2
MOV 1,3
ADD 2,3,SZC
SUB# 0,3,SZK
ERROR
LOOP
JSR #ENTLU

```

L085:

```

LDA 0,#BREG
STA 0,#TEM
SETUP 5
JSR #ENTIN
5
RAND
JSR #ENTRA
STA 0,#TEM
LDA 2,#TEM
MOVZL 2,2
INC 2,3
LOR 2,1
LOR 3,3
MOV 1,1
ADD 1,3
LDA 1,#TEM
SUB# 1,0,SJK
SUB# 0,3,SZK
ERROR
LOOP
JSR #ENTLU
ISZ TEM
LUA 0,#TEM
LUA 3,#END
SUB# 0,3,SZK
JMP L085

```

10082 ECL20

```

01
02
03
04 06010 006230
05 06011 000020
06 06012 004002
07 06013 177400
08 06014 170120
09 06015 162710
10 06016 024005
11 06017 122415
12 06020 101002
13
14
15 06025 006231
16
17
18 06026 006230
19 06027 000020
20 06030 004002
21 06031 000377
22 06032 175120
23 06033 171400
24 06034 156710
25 06035 024005
26 06036 136415
27 06037 101002
28
29
30 06044 006231
31
32
33 06045 006230
34 06046 000100
35
36 06047 006234
37 06050 040240
38 06051 115000
39 06052 020010
40 06053 105400
41 06054 112710
42 06055 122710
43 06056 101300
44 06057 143300
45 06060 116414
46
47
48 06065 006231
49

```

L086:

```

SETUP 20
JSR #ENTIN
20
JSR #*2
MOVZL 3,3
LOR 3,0
LDA 1,#077
SUB# 1,0,SJK
MOV 0,2,SZC
ERROR
LOOP
JSR #ENTLU

```

L087:

```

SETUP 20
JSR #ENTIN
20
JSR #*2
MOVZL 3,3
INC 3,2
LOR 2,3
LDA 1,#077
SUB# 1,0,SJK
MOV 0,10,SZC
ERROR
LOOP
JSR #ENTLU

```

L088:

```

SETUP 100
JSR #ENTIN
100
RAND
JSR #ENTRA
STA 0,#TEM
MOV 0,3
LDA 0,#TEM+TEM
INC 0,1
LOR 0,2
LOR 1,0
MOV 0,0
ADD 2,0
SUB# 0,3,SZK
ERROR
LOOP
JSR #ENTLU

```

```

/TEST "LOB"
/INITIALIZE TEST,....
/POINT TO WORD=1
/LEFT BYTE,
/AC(AC1)=CORRECT
/AC(AC0)=LOB RESULT
/AC(CARRY) SHOULD BE 0.
/ITERATE TEST ROUTINE,....

/TEST "LOB"
/INITIALIZE TEST,....
/POINT TO THE 377,
/AC(AC1)=CORRECT
/AC(AC3)=LOB RESULT
/AC(CARRY) SHOULD BE 0.
/ITERATE TEST ROUTINE,....

/TEST "LOB"
/INITIALIZE TEST,....
/AC(AC0)=RANDOM #
/AC(AC2)=HIGH PART
/AC(AC0)=LOW PART
/AC(AC0)=LOB RESULT
/AC(AC3)=CORRECT
/ITERATE TEST ROUTINE,....

```





18085 ECL2P

```
01  
02  
03  
04 06162 066230  
05 06163 080005  
06 06164 020016=  
07 06165 126400  
08 06166 152400  
09 06167 176400  
10 06170 190010  
11 06171 130015  
12 06172 175004  
13  
14  
15 06177 066231  
16  
17  
18 06200 066230  
19 06201 080005  
20 06202 034010=  
21 06203 102400  
22 06204 126400  
23 06205 152400  
24 06206 170010  
25 06207 190015  
26 06210 151004  
27  
28  
29 06215 066231  
  
STB41  
SETUP 5  
JSR #ENTIN  
LDA 0,#TEM*TEM  
SUB 1,1  
SUB 2,2  
SUB 3,3  
SUB 0,0  
ADD# 1,2,SNR  
MOV 3,3, SZR  
ERROR  
LOOP  
JSR #ENTLO  
  
ITERATE TEST ROUTINE.....  
  
STB51  
SETUP 5  
JSR #ENTIN  
LDA 3,#TEM*TEM  
SUB 0,0  
SUB 1,1  
SUB 2,2  
SUB 3,3  
SUB 3,3  
SUB 3,3  
SUB 2,2, SZR  
MOV 2,2, SZR  
ERROR  
LOOP  
JSR #ENTLO  
  
ITERATE TEST ROUTINE.....
```

18086 ECL2P

```
01  
02  
03 06215 066230  
04 06217 080200  
05  
06 06220 066234  
07 06221 024016=  
08 06222 115300  
09 06223 131400  
10 06224 147010  
11 06225 153010  
12 06226 054240  
13 06227 162414  
14  
15 06234 066231  
16  
17  
18 06235 066230  
19 06236 080100  
20 06237 066234  
21 06238 080200  
22 06239 066234  
23 06240 080240  
24 06241 105000  
25 06242 034013=  
26 06243 167010  
27 06244 034240  
28 06245 152414  
29  
30  
31 06252 066231  
32  
33 06253 066230  
34 06254 080100  
35  
36 06255 066234  
37 06256 105020  
38 06257 024016=  
39 06258 114000  
40 06259 147010  
41 06261 183000  
42 06262 183000  
43 06263 107010  
44 06264 034240  
45 06265 121300  
46 06266 116415  
47 06267 101002  
48  
49  
  
STB61  
SETUP 400  
JSR #ENTIN  
RAND  
JSR #ENTRA  
LOA 1,#TEM*TEM  
MOV# 0,3  
INC 1,2  
STB 1,3  
STB 2,0  
LOA 3,TEM  
SUB# 3,0, SZK  
ERROR  
LOOP  
JSR #ENTLO  
  
ITERATE TEST ROUTINE.....  
  
STB71  
SETUP 100  
JSR #ENTIN  
RAND  
JSR #ENTRA  
STA 0,TEM  
MOV 0,1  
LOA 3,#TEM*TEM+1  
STB 3,1  
LOA 2,TEM  
SUB# 1,2, SZK  
ERROR  
LOOP  
JSR #ENTLO  
  
ITERATE TEST ROUTINE.....  
  
STB81  
SETUP 100  
JSR #ENTIN  
RAND  
JSR #ENTRA  
MOVZ 0,1  
LOA 0,#TEM*TEM  
INC 0,2  
STB 2,1  
MOVS 1,1  
STB 0,1  
LOA 3,TEM  
MOVS 1,0  
SUB# 0,0, SNR  
MOV 0,0, SZC  
ERROR  
LOOP
```

10087 ECL20

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

LDA 0, RREG  
STA 0, TEM  
SETUP 3,  
JSR #ENTIN  
3,  
RAND  
JSR #ENTRA  
LDA 2, TEM  
MOVZL 2, 2  
MOVS 0, 3  
STA 2, 3  
SIB 2, 2  
MOVZL 2, 2  
LDA 3, 0, 2  
SUB# 0, 1, SZK  
ERROR  
LOOP  
JSR #ENTLO  
ISZ TEM  
LDA 0, TEM  
LDA 1, BEND  
SUB# 0, 1, SZK  
JMP ST89

ITEST "STB"  
INITIALIZE TEST.....  
STORE RANDOM #  
VIA STB  
STORE LEFT BYTE  
STORE RIGHT BYTE  
IC(AC3)=STB RESULT  
IC(AC0)=CURRENT  
IC(AC2)=0 IN BUFFER.  
ITERATE TEST ROUTINE....  
NEXT BUFFER WORD  
TEST FOR END OF  
BUFFER.

10088 ECL20

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39

LDA 0, RREG  
STA 0, TEM  
SETUP 3,  
JSR #ENTIN  
3,  
RAND  
JSR #ENTRA  
MOV 0, 2  
LDA 0, TEM  
MOVZL 0, 0  
INC 0, 1  
SIB 1, 2  
MOVS 2, 2  
SIB 0, 2  
MOVZL 0, 3  
LDA 0, 0, 3  
MOVS 2, 2  
SUB# 2, 0, SZK  
ERROR  
LOOP  
JSR #ENTLO  
ISZ TEM  
LDA 0, TEM  
LDA 1, BEND  
SUB# 0, 1, SZK  
JMP STB90

ITEST "STB"  
INITIALIZE TEST.....  
IC(AC0)=RANDOM #  
STORE A WORD IN MEMORY  
VIA BYTE STORE. CHECK  
FOR ACCURACY.  
STORE RIGHT BYTE  
STORE LEFT BYTE  
IC(AC3)=MEMORY ADDRESS  
IC(AC0)=STB RESULT  
IC(AC2)=CORRECT  
ITERATE TEST ROUTINE....  
NEXT BUFFER WORD  
TEST FOR LAST.  
ITEST "STB"  
INITIALIZE TEST.....  
STORE RIGHT BYTE  
LOAD IT BACK.  
ITERATE TEST ROUTINE....

```
10089 ECL20
01 *****EGGS & DIRT DATA BLOCKS*****
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

; *****EGGS & DIRT DATA BLOCKS*****
;
; DTOS AUTO MOVE SWITCH
; PRIMARY DEVICE CODE TO BE TESTED
; CAT SWITCH, SET IF CAT LOADED
; PASS COUNT, # OF TIMES TO RUN
; RETURN POINT TO RESTORE DTOS
; DEFAULT SWITCH REGISTER

EGGS:
WB 05437 000000 AUTO: 0
WB 06444 000000 DEVI: 0
WB 06441 000000 CATSW: 0
WB 06442 000000 PCNT: 0
WB 06443 000000 PTRN: 0
WB 06444 000000 SMREG: 0

WB 06445 000100 VCTAB: 100
WB 06549 006545 PGEND: PKGEN0
WB 06546 047503 .TXT /COPYRIGHT (C) DSC,1974,76
WB 06450 044520
WB 044522
WB 044107
WB 020124
WB 041450
WB 020451
WB 043504
WB 026103
WB 034461
WB 032067
WB 033454

WB 06562 000000 ALL RIGHTS RESERVED/
WB 065114
WB 051040
WB 043511
WB 052110
WB 020123
WB 042523
WB 042523
WB 043102
WB 042105
WB 000000

WB 06575 141705 DIRT: .TXT IECLIPSE201
WB 144714
WB 051520
WB 131305
WB 131060
WB 050000
WB 06603 000000
WB 06604 000000
WB 06605 175772
WB 06606 000000
WB 06607 000000
WB 06610 000000
WB 06611 000000
WB 06612 000000

WB .END DTOSB

INTERNAL COUNT DONE?
INDEX, LOOP PASS
IF YES, BUMP PASS COUNT
IF PASS CNT > 65K
RESTORE INTERNAL COUNT
IF CHECK SWITCH 4
IF SET, DO NOT
IF PRINT PASS COUNT
IF PRINT PASS COUNT
IF STANDAND DTOS RETURN
IF AUTO MODE?
IF NO, RE-DO TEST
IF YES, PASS COUNT ZERO?
IF NO, BACK TO TEST
IF YES, BACK TO DTOS

END OF TEST
IF RESTART TEST...
IF START CAT THEN TEST

; *** END OF TEST ROUTINES ***
```

0091 ECL20  
01 105371  
02 102256  
03 817777  
04 160000  
05 004777  
06 005017  
07 000007  
08 173777  
09 125400  
10 132210  
11 000017  
12 005000  
13 005400  
14 102210  
15 100313  
16 113710  
17 000270  
18 002266  
19 100317  
20 100316  
21 000110  
22 000000  
23 002100  
24 000000  
25 177700  
26 000240  
27 177770  
28 000212  
29 104400  
30 022000  
31 000000  
32 000011  
33 000031  
34 000144  
35 000100  
36 177740  
37 000010  
38 000140  
39 000377  
40 000200  
41 000400  
42 001777

0092 ECL20  
AC0 000245  
AC1 000246  
AC2 000247  
AC3 000250

|       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|
| 13/37 | 20/06 | 20/26 | 21/21 | 21/34 | 22/05 | 22/16 |
| 22/41 | 23/35 | 35/25 | 36/20 | 37/38 | 40/45 | 41/21 |
| 45/21 | 51/47 | 52/48 | 51/37 | 52/37 | 56/37 | 57/36 |
| 13/36 | 21/22 | 21/35 | 22/28 | 22/43 | 23/23 | 35/26 |
| 36/21 | 37/39 | 40/46 | 41/22 | 51/48 | 61/38 | 62/38 |
| 66/38 | 67/37 |       |       |       |       |       |
| 13/39 | 21/23 | 21/36 | 22/29 | 22/45 | 23/24 | 35/27 |
| 36/22 | 37/40 | 40/47 | 41/23 | 45/19 | 45/20 | 51/49 |
| 51/55 | 51/57 | 52/47 |       |       |       |       |
| 13/40 | 22/47 | 23/30 | 35/20 | 36/15 | 37/33 | 40/40 |
| 41/16 | 43/40 | 43/46 | 44/41 | 44/47 | 45/18 | 46/36 |
| 46/40 | 47/25 | 49/32 | 50/40 | 51/46 | 51/50 | 51/54 |
| 51/56 | 52/06 | 52/49 | 53/11 | 53/23 | 53/42 | 53/46 |
| 54/12 | 54/27 | 54/50 | 55/18 | 55/39 | 55/57 | 56/23 |
| 57/11 | 57/40 | 58/29 | 58/49 | 59/15 | 59/30 | 59/47 |
| 60/20 | 60/41 | 61/34 | 61/35 | 62/34 | 62/35 | 63/15 |
| 63/30 | 63/44 | 64/15 | 64/30 | 64/48 | 65/20 | 65/41 |
| 66/34 | 66/35 | 67/33 | 67/34 | 68/22 | 68/48 | 69/44 |
| 71/01 | 71/20 | 71/40 | 71/59 | 72/46 | 73/19 | 73/19 |
| 74/19 | 74/23 | 74/44 | 75/20 | 76/22 | 76/27 | 77/49 |
| 78/18 | 78/20 | 78/36 | 78/40 | 79/14 | 79/27 | 79/44 |
| 80/19 | 81/16 | 81/38 | 82/14 | 82/29 | 82/47 | 83/14 |
| 83/28 | 83/44 | 84/15 | 85/14 | 85/28 | 86/15 | 86/30 |
| 86/49 | 87/20 | 88/21 | 88/36 |       |       |       |
| 10/20 | 43/48 | 44/49 |       |       |       |       |
| 16/12 | 23/25 | 90/05 |       |       |       |       |
| 13/32 | 53/06 | 53/17 | 53/29 | 53/42 | 54/03 | 54/19 |
| 54/40 | 55/08 | 59/11 | 60/32 | 63/06 | 63/21 | 64/09 |
| 65/11 | 65/32 | 69/05 | 71/50 | 72/07 | 73/09 | 74/11 |
| 13/33 | 64/24 |       |       |       |       |       |
| 13/35 | 72/42 |       |       |       |       |       |
| 14/29 | 46/13 | 46/29 | 46/51 | 49/04 | 49/22 | 50/04 |
| 50/38 | 50/41 |       |       |       |       |       |
| 14/30 | 46/14 | 46/30 | 46/52 | 49/05 | 49/23 | 50/05 |
| 14/31 | 46/15 | 46/25 | 46/53 | 49/06 | 49/17 | 50/06 |
| 14/32 | 46/16 | 46/54 | 49/07 | 50/07 |       |       |
| 47/14 | 48/03 |       |       |       |       |       |
| 48/29 |       |       |       |       |       |       |
| 50/16 | 50/25 |       |       |       |       |       |
| 50/23 | 50/49 |       |       |       |       |       |
| 50/22 | 50/29 |       |       |       |       |       |
| 48/35 | 48/40 |       |       |       |       |       |
| 48/42 |       |       |       |       |       |       |
| 49/03 |       |       |       |       |       |       |
| 49/05 |       |       |       |       |       |       |
| 49/18 |       |       |       |       |       |       |
| 49/24 |       |       |       |       |       |       |
| 49/20 | 49/20 | 49/20 |       |       |       |       |
| 49/26 | 50/10 | 49/31 |       |       |       |       |
| 8/20  | 45/17 | 46/35 | 46/39 | 47/24 | 49/31 | 50/39 |
| 51/45 | 51/53 | 52/45 | 61/33 | 62/33 | 66/33 | 67/32 |
| 29/43 | 46/40 | 49/32 |       |       |       |       |
| 30/09 | 46/45 | 49/37 |       |       |       |       |
| 30/18 | 46/50 | 50/03 |       |       |       |       |
| 30/26 | 46/50 |       |       |       |       |       |
| 30/37 | 47/25 | 50/40 |       |       |       |       |
| 31/09 | 51/46 |       |       |       |       |       |
| 31/24 | 45/18 | 51/04 | 52/46 |       |       |       |
| 13/56 | 51/14 | 51/15 |       |       |       |       |

MC

MC



0095 ECL20

BTZ3 004476 65/06 65/09 98/07  
 BTZ4 004521 65/26 65/30  
 BTZ5 004543 66/02  
 BTZ51 004555 66/13  
 BTZ52 004565 66/21  
 BTZ53 004579 66/24  
 BTZ54 004606 66/26  
 BTZ55 004612 67/02  
 BTZ61 004625 67/14  
 BTZ62 004641 67/26  
 BTZ63 004657 67/28  
 BTZ64 004673 67/33  
 CATSM 006441 12/31  
 CHAR1 001250 25/42  
 CHAR2 001277 27/30  
 CHAR3 001314 27/40  
 CHAR4 001325 27/47  
 CHARE 002253 27/57  
 CHORZ 002254 13/44  
 CHRXY 001332 13/45  
 CRY 002251 27/05  
 DEY 006440 13/41  
 DIRT 006575 90/06  
 DIV0 002046 12/05  
 DIV1 002077 36/02  
 DIVH1 001615 31/60  
 DMOR 006432 89/29  
 OTOSB 002268 12/43  
 EDIV 002250 39/03  
 EDIVH 001515 30/56  
 E669 006437 12/16  
 EN0 006372 89/02  
 ENTBE 000233 13/24

0096 ECL20

ENTL0 000231 86/03 86/19 85/34 87/06 88/03 88/30  
 ENTRA 000234 13/22 35/34 36/29 37/30 37/30 40/54 41/30  
 ENTR 000235 43/47 44/48 45/27 47/13 50/27 51/43 52/56  
 ERK 001020 53/12 53/24 53/47 54/13 54/28 54/51 55/19  
 ERR1 001055 55/40 55/56 56/24 57/12 57/49 58/30 58/50  
 ERR2 001071 59/31 59/48 60/21 60/42 61/44 62/44 65/21  
 ERR3 001074 63/31 63/45 64/31 64/41 64/49 65/21  
 ERRA 001109 65/42 66/44 67/43 68/23 68/49 69/45 71/02  
 ERRT 000212 71/21 71/41 71/60 72/47 73/20 74/24 74/45  
 ERRT0R 000220 75/21 76/28 77/52 78/31 78/41 79/30  
 EPC 000252 79/28 79/45 80/20 81/17 81/30 82/16  
 ERMSG 001337 82/48 82/48 83/15 83/26 83/45 84/16  
 ERN 001020 86/16 86/31 87/21 88/22 88/39 89/15  
 ERR1 001055 13/25 17/15 17/15 24/20 27/07 27/30  
 ERRA 001109 56/49 56/49 56/73 74/07 74/30 75/06  
 ERRT 000212 68/34 73/08 81/07 81/25 82/36 86/06  
 ERRT0R 000220 87/08 88/08 88/36 89/17 89/17 23/04  
 ERMSG 001337 13/42 22/04 22/15 22/15 23/19 23/19  
 ERN 001020 22/31 29/06 29/06 29/06 29/06 29/06  
 ERR1 001055 23/04 23/11 23/17 23/17 23/17 23/17  
 ERRA 001109 23/07 23/07 23/28 23/28 23/28 23/28  
 ERRT 000212 22/25 23/28 23/28 23/28 23/28 23/28  
 ERRT0R 000220 13/24 22/04 22/15 22/15 22/15 22/15  
 EPC 000252 7/22 43/39 43/41 43/41 44/46 44/46  
 ERMSG 001337 53/41 53/49 54/11 54/26 54/49 55/17  
 ERN 001020 55/56 56/22 57/16 57/47 58/28 58/48  
 ERR1 001055 59/29 59/46 60/19 60/40 63/14 63/29  
 ERRA 001109 64/14 64/29 64/47 65/19 65/40 66/21  
 ERRT 000212 69/43 70/08 71/19 71/39 71/56 71/88  
 ERRT0R 000220 73/18 74/18 74/22 74/43 75/19 76/21  
 EPC 000252 77/48 78/17 78/19 78/35 78/39 79/13  
 ERMSG 001337 79/43 80/18 81/15 81/37 82/13 82/46  
 ERN 001020 83/43 83/43 83/43 84/14 85/13 85/27  
 ERR1 001055 86/29 86/48 87/19 88/20 88/37 88/37  
 ERRA 001109 10/30 43/07 44/08 44/08 44/08 44/08  
 ERRT 000212 11/05 18/35 18/35 18/35 18/35 18/35  
 ERRT0R 000220 14/27 43/58 46/10 46/17 46/43 47/04  
 EPC 000252 48/42 49/10 49/35 50/36 50/36 50/36  
 ERMSG 001337 13/17 15/24 16/03 16/17 16/17 21/24  
 ERN 001020 13/11 16/03 16/03 16/03 16/03 16/03  
 ERR1 001055 13/21 20/06 20/20 20/20 20/20 20/20  
 ERRA 001109 13/10 21/33 21/33 21/33 21/33 21/33  
 ERRT 000212 13/09 16/07 16/07 16/07 16/07 16/07  
 ERRT0R 000220 25/10 18/45 18/45 18/45 18/45 18/45  
 EPC 000252 11/16 15/16 15/16 15/16 15/16 15/16  
 ERMSG 001337 11/15 18/41 18/41 18/41 18/41 18/41  
 ERN 001020 11/15 18/41 18/41 18/41 18/41 18/41

ENTER 000232

ENTER 000232 41/16 45/18 48/40 48/40 48/40 48/40  
 ENTLN 000230 51/48 51/54 51/48 51/48 51/48 51/48  
 ENTR 000232 53/11 53/23 53/11 53/23 53/11 53/23  
 ENTRA 000234 55/16 55/39 55/16 55/39 55/16 55/39  
 ENTR 000235 58/49 59/15 58/49 59/15 58/49 59/15  
 ERK 001020 63/30 63/44 63/30 63/44 63/30 63/44  
 ERR1 001055 68/22 68/48 68/22 68/48 68/22 68/48  
 ERR2 001071 71/59 72/46 71/59 72/46 71/59 72/46  
 ERR3 001074 76/22 76/27 76/22 76/27 76/22 76/27  
 ERRA 001109 79/14 79/27 79/14 79/27 79/14 79/27  
 ERRT 000212 82/29 82/47 82/29 82/47 82/29 82/47  
 ERRT0R 000220 85/28 86/15 85/28 86/15 85/28 86/15  
 EPC 000252 41/04 42/02 41/04 42/02 41/04 42/02  
 ERMSG 001337 51/02 52/03 51/02 52/03 51/02 52/03  
 ERN 001020 54/35 55/03 54/35 55/03 54/35 55/03  
 ERR1 001055 58/03 58/43 58/03 58/43 58/03 58/43  
 ERRA 001109 81/04 82/04 81/04 82/04 81/04 82/04  
 ERRT 000212 64/34 65/04 64/34 65/04 64/34 65/04  
 ERRT0R 000220 69/03 70/03 69/03 70/03 69/03 70/03  
 EPC 000252 74/04 74/27 74/04 74/27 74/04 74/27  
 ERMSG 001337 78/24 78/04 78/24 78/04 78/24 78/04  
 ERN 001020 82/04 82/18 82/04 82/18 82/04 82/18  
 ERR1 001055 85/04 85/18 85/04 85/18 85/04 85/18  
 ERRA 001109 83/52 83/52 83/52 83/52 83/52 83/52  
 ERRT 000212 83/18 83/52 83/18 83/52 83/18 83/52  
 ERRT0R 000220 83/18 83/52 83/18 83/52 83/18 83/52  
 EPC 000252 13/21 13/21 13/21 13/21 13/21 13/21  
 ERMSG 001337 43/04 43/04 43/04 43/04 43/04 43/04  
 ERN 001020 53/15 53/27 53/15 53/27 53/15 53/27  
 ERR1 001055 55/43 56/06 55/43 56/06 55/43 56/06  
 ERRA 001109 59/04 59/34 59/04 59/34 59/04 59/34  
 ERRT 000212 63/19 63/34 63/19 63/34 63/19 63/34  
 ERRT0R 000220 65/24 66/03 65/24 66/03 65/24 66/03  
 EPC 000252 71/05 71/05 71/05 71/05 71/05 71/05  
 ERMSG 001337 75/03 76/02 75/03 76/02 75/03 76/02  
 ERN 001020 79/18 79/31 79/18 79/31 79/18 79/31  
 ERR1 001055 82/33 82/33 82/33 82/33 82/33 82/33  
 ERRA 001109 82/33 82/33 82/33 82/33 82/33 82/33  
 ERRT 000212 82/33 82/33 82/33 82/33 82/33 82/33  
 ERRT0R 000220 82/33 82/33 82/33 82/33 82/33 82/33

089/ ECL24

0898 ECL20

|        |        |       |       |       |       |       |       |       |       |       |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| ISF    | 000004 | 11/13 | 18/36 | 21/10 | 21/31 | 22/27 | 13/04 | 17/25 | 24/37 | 39/03 |
| ISTK   | 000221 | 13/12 | 20/11 | 21/10 | 21/31 |       | 11/14 | 39/47 | 40/15 | 39/11 |
| ITK    | 000206 | 12/52 | 20/12 | 21/05 | 21/11 |       | 13/49 | 40/04 |       |       |
| ITNCT  | 000207 | 12/53 | 20/16 | 21/05 | 21/36 |       | 35/07 |       |       |       |
| ITREC  | 000211 | 12/55 | 17/12 | 21/07 | 21/36 | 22/21 | 31/39 | 35/33 |       |       |
| ITRER  | 000210 | 12/54 | 20/15 | 21/07 | 21/40 | 22/23 | 31/39 | 35/33 |       |       |
| ITRET  | 000201 | 12/46 | 20/07 | 21/08 |       |       | 31/52 | 35/24 | 37/37 | 41/26 |
| ITRP   | 000222 | 13/13 | 35/11 | 40/31 | 41/07 |       | 31/52 | 35/28 | 40/48 | 41/24 |
| ISAN   | 000223 | 13/14 |       |       |       |       | 12/47 | 15/19 |       |       |
| ISRE   | 000328 | 14/36 |       |       |       |       | 12/29 | 15/05 |       |       |
| J8R3   | 000325 | 14/35 |       |       |       |       | 14/15 | 15/29 | 36/24 | 38/17 |
| K6     | 000264 | 13/54 | 46/41 | 46/46 | 47/30 | 49/39 | 14/16 | 39/47 | 48/23 | 49/49 |
| KCLRF  | 001770 | 16/09 | 16/20 | 23/22 | 33/47 |       | 14/18 | 36/25 | 38/12 | 38/20 |
| L0B4   | 005634 | 79/17 |       |       |       |       | 39/48 | 40/24 | 48/50 | 41/26 |
| L0B1   | 005652 | 79/17 |       |       |       |       | 14/17 | 35/31 | 36/26 | 38/13 |
| L0B2   | 005667 | 78/30 |       |       |       |       | 39/33 | 39/49 | 48/22 | 40/51 |
| L0B3   | 005711 | 80/04 | 80/25 |       |       |       | 14/18 | 38/09 | 39/38 | 40/63 |
| L0B4   | 005737 | 81/23 | 81/44 |       |       |       | 12/30 | 24/29 | 35/13 | 36/21 |
| L0B5   | 005766 | 82/03 |       |       |       |       | 14/11 | 24/29 | 35/13 | 36/16 |
| L0B6   | 006010 | 82/03 |       |       |       |       | 37/20 | 37/30 | 40/33 | 40/41 |
| L0B7   | 006028 | 82/17 |       |       |       |       | 14/12 | 24/30 | 35/14 | 36/17 |
| L0B8   | 006045 | 82/32 |       |       |       |       | 37/23 | 37/35 | 48/34 | 40/42 |
| L00P   | 000000 | 7/05  | 35/33 | 36/28 | 37/29 | 41/29 | 14/13 | 24/31 | 35/15 | 36/18 |
|        |        | 43/46 | 44/47 | 45/26 | 47/12 | 50/26 | 37/26 | 37/35 | 40/35 | 40/43 |
|        |        | 53/11 | 53/23 | 53/46 | 54/12 | 54/27 | 14/14 | 24/23 | 24/32 |       |
|        |        | 55/39 | 55/57 | 56/23 | 57/11 | 57/48 | 9/02  |       |       |       |
|        |        | 59/15 | 59/30 | 59/47 | 60/20 | 60/41 | 9/11  |       |       |       |
|        |        | 63/15 | 63/30 | 63/44 | 64/15 | 64/30 | 29/02 | 49/21 | 20/18 | 22/32 |
|        |        | 65/41 | 65/43 | 67/42 | 68/22 | 68/48 | 12/48 | 16/11 | 20/18 | 22/32 |
|        |        | 71/50 | 71/40 | 71/59 | 72/46 | 73/19 | 12/49 | 49/02 | 89/08 | 89/23 |
|        |        | 75/08 | 75/27 | 76/57 | 77/51 | 78/20 | 12/50 | 49/02 | 89/08 | 89/23 |
|        |        | 79/07 | 79/44 | 80/19 | 81/16 | 81/38 | 14/37 | 49/02 | 89/08 | 89/23 |
|        |        | 82/47 | 83/14 | 83/28 | 83/44 | 84/15 | 90/08 |       |       |       |
|        |        | 86/15 | 86/30 | 87/20 | 88/21 | 88/38 | 14/37 |       |       |       |
|        |        | 13/22 | 21/04 |       |       |       | 13/10 | 25/04 |       |       |
| L0P1   | 000737 | 21/13 |       |       |       |       | 25/44 |       |       |       |
| L0P2   | 000767 | 21/20 | 21/37 |       |       |       | 25/26 |       |       |       |
| L0P3   | 000771 | 21/40 | 21/40 |       |       |       | 25/32 |       |       |       |
| L0PRE  | 000213 | 12/57 | 21/04 |       |       |       | 25/07 | 25/17 | 25/29 | 25/41 |
| M1     | 000272 | 14/02 | 15/36 | 21/09 | 21/49 | 67/35 | 13/48 | 25/05 | 25/45 |       |
| MAXLO  | 000215 | 13/05 | 15/25 |       |       |       | 16/07 | 38/28 |       |       |
| M0R    | 002100 | 37/03 |       |       |       |       | 38/35 | 38/36 |       |       |
| M0R1   | 002102 | 37/06 |       |       |       |       | 14/31 | 38/39 |       |       |
| M0R2   | 002115 | 37/18 |       |       |       |       | 21/25 | 38/28 | 38/42 |       |
| M0R3   | 002130 | 37/18 |       |       |       |       | 11/12 | 33/12 |       |       |
| M0R4   | 002145 | 37/22 | 37/28 | 37/32 | 37/32 |       | 35/12 | 38/03 | 40/12 |       |
| M0R5   | 002146 | 37/31 | 37/44 |       |       |       | 14/19 | 38/03 | 38/14 |       |
| M0CK   | 002163 | 35/17 | 38/12 | 38/16 | 38/51 | 41/13 | 13/09 | 25/14 |       |       |
| M0CK1  | 000174 | 38/19 | 38/22 | 38/25 | 38/49 |       | 13/04 | 13/05 | 13/18 | 15/14 |
| M0CKR  | 000164 | 38/23 | 37/19 | 36/14 | 37/32 | 40/39 | 49/18 | 49/20 |       |       |
| M0M4   | 001632 | 32/13 | 37/33 |       |       |       | 13/59 | 38/45 | 39/37 |       |
| M0RET  | 000311 | 14/28 | 38/16 | 38/25 | 38/26 | 38/52 | 39/08 | 40/32 |       |       |
| M0RENT | 000011 | 12/23 | 15/09 | 16/05 | 24/47 |       | 39/24 | 39/22 |       |       |
| M0S1Z  | 001754 | 16/04 |       |       |       |       | 39/24 | 39/22 |       |       |
| M0SKE  | 000256 | 13/47 | 25/50 | 25/59 |       |       | 39/33 | 39/33 |       |       |
| M0SS   | 001233 | 13/11 | 25/49 |       |       |       | 39/23 | 39/36 |       |       |
| M0SS1  | 001230 | 25/02 | 26/01 |       |       |       | 7/34  |       |       |       |

MC

MC

MC

MC

MC

MC

MC

MC

MC

MC

MC

MC

MC

MC

MC

MC

MC

MC

MC

MC

MC

0099 ECL20

PSMUL 002306  
RAN 000276

RAND 000012 MC

RANST 000170 MC  
REST 001121  
RTRN 000440  
SC 000002  
SOLV 002335  
SDIV1 002345  
SDIV2 002366  
SDVH1 001651  
SETSM 001772  
SETUP 000005 MC

SF 000043

SIZE 001144

SL 000042

SMCK 002215

SMH1 001666

SMUL0 002370

SMUL1 002377

SMUL2 002421

SP 000040

SHAN 000277

STACK 000277 MC

STAKT 000063

STR0 000066

STR1 000104

STR2 000122

STR3 000142

STR4 000162

STR5 000200

STR6 000216

STR7 000235

STR8 000276

STR9 000326

STR00 000326

STR91 000356

STK 000046

SWME5 000533

SWREG 000444

SZ01 000161

SZ1 000166

SZ10 000505

0100 ECL20

SZ11 000520

SZ12 000533

SZ22 000544

SZ21 000522

SZ31 000527

SZ31 000527

SZ32 000504

SZ33 000511

SZ34 000531

SZ34 000520

SZ4 000530

SZ41 000527

SZ5 000544

SZ51 000553

SZ52 000571

SZ6 000537

SZ60 000540

SZ61 000541

SZ62 000530

SZ63 000534

SZ7 000535

SZ71 000544

SZ8 000571

SZ81 000551

SZ9 000556

SZ91 000500

SZ90 000410

SZ81 000430

SZ82 000444

SZ83 000500

SZ83 000511

SZ84 000526

SZ80 000535

SZ81 000543

SZ85 000557

SZ86 000565

SZ87 000501

SZ88 000546

SZ89 000550

SZ82 000556

SZ83 000564

SZ84 000577

SZ89 000701

SZ91 000710

SZ92 000724

SZ93 000741

SZ81 000745

SZ81 000764

SZ82 000777

SZ83 000800

SZ84 000804

SZ85 000805

SZ801 000822

SZ802 000830

SZ803 000835

SZ8C 000843

SZ8C1 000855

SZ0 0008142

78/32

78/36

73/02

74/03

74/11

74/16

74/16

74/19

74/23

74/26

74/35

75/02

75/11

75/09

75/22

76/01

76/13

76/18

76/18

76/20

76/27

76/36

76/38

77/05

77/07

77/14

78/03

78/11

53/03

53/14

53/26

54/05

54/10

54/12

54/32

54/34

54/42

55/02

55/10

55/21

55/42

56/03

56/05

56/13

56/19

56/28

56/31

56/39

57/02

57/13

57/18

57/34

57/45

57/42

57/39

58/02

58/16

58/22

58/27

58/32

58/42

58/54

71/45

77/5:



0101 ECL20

TEM 000240

|              |       |       |       |       |       |       |
|--------------|-------|-------|-------|-------|-------|-------|
| 13/31        | 42/07 | 42/14 | 43/02 | 43/06 | 43/32 | 43/42 |
| 43/49        | 43/53 | 44/03 | 44/07 | 44/13 | 44/43 | 44/50 |
| 44/54        | 53/00 | 53/19 | 54/05 | 54/15 | 54/21 | 54/30 |
| 54/39        | 55/07 | 55/25 | 55/26 | 55/35 | 55/46 | 55/48 |
| 55/53        | 56/38 | 56/44 | 56/50 | 57/13 | 57/31 | 57/43 |
| 58/34        | 58/46 | 58/51 | 58/52 | 59/07 | 59/09 | 59/11 |
| 59/22        | 59/24 | 59/29 | 59/38 | 59/39 | 59/44 | 59/49 |
| 60/16        | 60/31 | 60/37 | 60/06 | 60/11 | 60/23 | 60/27 |
| 63/36        | 63/38 | 63/40 | 64/07 | 64/11 | 64/22 | 64/26 |
| 64/38        | 64/45 | 65/10 | 65/16 | 65/31 | 65/37 | 66/36 |
| 68/45        | 69/07 | 69/41 | 70/07 | 70/58 | 71/07 | 71/09 |
| 71/15        | 71/26 | 71/28 | 71/35 | 71/49 | 71/53 | 72/05 |
| 72/41        | 73/07 | 73/10 | 74/09 | 74/20 | 76/36 | 76/43 |
| 77/04        | 77/06 | 77/34 | 77/53 | 78/06 | 78/08 | 78/15 |
| 78/20        | 78/20 | 78/37 | 79/07 | 79/08 | 79/21 | 79/22 |
| 79/35        | 79/36 | 80/03 | 80/09 | 80/10 | 80/21 | 80/22 |
| 81/26        | 81/26 | 81/27 | 81/34 | 81/40 | 81/41 | 82/37 |
| 82/39        | 83/07 | 83/08 | 83/10 | 83/21 | 83/22 | 83/24 |
| 83/34        | 83/37 | 83/40 | 84/08 | 84/09 | 84/11 | 85/06 |
| 85/20        | 86/07 | 86/12 | 86/23 | 86/25 | 86/27 | 86/39 |
| 86/44        | 87/04 | 87/10 | 87/22 | 87/23 | 88/03 | 88/10 |
| 88/23        | 88/24 | 88/32 | 88/32 | 88/32 | 88/32 | 88/32 |
| 14/05        | 43/31 | 44/32 | 56/49 | 57/07 | 76/37 | 76/51 |
| 14/03        | 42/08 | 42/22 | 42/23 | 42/24 | 42/24 | 42/24 |
| 76/52        | 76/53 | 77/32 | 77/49 | 77/49 | 77/49 | 77/49 |
| 14/04        | 43/30 | 43/41 | 44/31 | 44/42 | 44/42 | 44/42 |
| 77/45        | 77/50 | 77/50 | 77/50 | 77/50 | 77/50 | 77/50 |
| 11/06        | 18/17 | 46/16 | 46/44 | 47/03 | 48/31 | 48/43 |
| 14/28        | 45/57 | 50/12 | 50/12 | 50/12 | 50/12 | 50/12 |
| 49/11        | 49/36 | 49/36 | 49/36 | 49/36 | 49/36 | 49/36 |
| 7/48         | 7/48  | 18/04 | 18/04 | 18/04 | 18/04 | 18/04 |
| TKP 000516   | MC    | 13/13 | 13/13 | 13/13 | 13/13 | 13/13 |
| TKST 000124  | MC    | 8/03  | 8/03  | 8/03  | 8/03  | 8/03  |
| UFLIM 000253 | MC    | 9/26  | 9/26  | 9/26  | 9/26  | 9/26  |
| UPLIN 000324 | MC    | 14/34 | 14/34 | 14/34 | 14/34 | 14/34 |
| VCTAR 000445 | MC    | 9/12  | 9/12  | 9/12  | 9/12  | 9/12  |
| VEC 000665   | MC    | 19/04 | 19/04 | 19/04 | 19/04 | 19/04 |
| VECTO 000331 | MC    | 10/02 | 10/02 | 10/02 | 10/02 | 10/02 |
| VKET 000705  | MC    | 19/04 | 19/04 | 19/04 | 19/04 | 19/04 |
| ZERUA 000151 | MC    | 8/13  | 8/13  | 8/13  | 8/13  | 8/13  |
| *BRAN 001113 | MC    | 13/26 | 13/26 | 13/26 | 13/26 | 13/26 |
| *EGS 000010- | MC    | 12/22 | 12/22 | 12/22 | 12/22 | 12/22 |
| *KAND 001115 | MC    | 13/25 | 13/25 | 13/25 | 13/25 | 13/25 |
| *KANS 001133 | MC    | 13/14 | 13/14 | 13/14 | 13/14 | 13/14 |



**DataGeneral**

---

---

**DIAGNOSTIC  
LISTING**

---

---

LISTING

096-000242-02

PROGRAM

EXERCISER FOR ECLIPSE  
PART 4

TAPE

095-000227-02

ABSTRACT

'ECLIPSE21' IS AN EXERCISER PROGRAM USED TO TEST THE RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF THE ECLIPSE COMPUTER. 'ECLIPSE21' EXERCISES THE EXTENDED INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES OF ITS RELIABLE OPERATION.



```

0001 ECL21 MACRO REV 03.00      14:54:57 08/06/76
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

*****
? NAME1 ECLIPSE21.SR          PART NUMBER: 094-000626
?
? DESCRIPTION: ECLIPSE EXERCISER, PART 4
? REVISION HISTORY:
?
? REV.      DATE
? 00      08/02/74
? 01      12/29/74
? 02      08/06/76
?
? COPYRIGHT (C) DATA GENERAL CORPORATION. 1974. 1976
? ALL RIGHTS RESERVED.
*****

10002 ECL21
02
03
04
05
06
07

-TITL  ECL21
/ECLIPSE21
/ECLIPSE21 - CONTINUATION OF ECLIPSE20
/PART 4 OF EXERCISER FOR ECLIPSE
/

```

10883 ECL21

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47

EXERCISER FOR ECLIPSE1 PART 4

PROGRAM NAME

ECLIPSE21

GENERAL DESCRIPTION

'ECLIPSE21' IS AN EXERCISER PROGRAM USED TO TEST THE RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF THE ECLIPSE COMPUTER. 'ECLIPSE21' EXERCISES THE EXTENDED INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES OF ITS RELIABLE OPERATION.

THE INSTRUCTIONS EXERCISED ARE AS FOLLOWS:

PSH,POP,PSHR,POPJ,SAVE,RTN,AND POPB

LOCATIONS 200 TO 205 IN PAGE 0 ARE FIXED FOR ECLIPSE21 PROGRAM.

LOCATION 203 KEEPS TRACK OF NUMBER OF PASSES RUN THROUGH ECLIPSE21 PROGRAM.

LOCATION 201 KEEPS TRACK OF THE TEST RUNNING AT PRESENT AND IS USEFUL FOR DEBUG WHEN LOOPING OCCURS IN THE PROGRAM.

LOCATION 202 CONTAINS THE STARTING ADDRESS OF ECLIPSE21 PROGRAM.

LOCATION 208 IS USED BY DTOS.

LOCATION 204 KEEPS TRACK OF INTERNAL PASS COUNT WHICH IS FIXED BY LOCATION 205.

FIRST PASS THROUGH ECLIPSE21 TEST WILL RUN SUPERFAST.

NEXT PASSES WILL RUN SLOWER AS EACH TEST IS RUN SEVERAL TIMES TO RUN ALL RANDOM NUMBER COMBINATIONS.

MACHINE REQUIREMENTS

ECLIPSE PROCESSOR

4K READ-WRITE MEMORY

CONSOLE EQUIPMENT

10884 ECL21

14.

SWITCH SETTINGS

THIS PROGRAM USES DATA SWITCHES AS FOLLOWS

SW'0' - USE CONTENTS OF 'SWREG' IF 0

SW'1' - USE DATA SWITCHES IF 1

SW'2' - LOOP ON FAILING TEST IF 0

SW'3' - PROCEED TO NEXT TEST IF 1

SW'4' - OUTPUT TO TTY IF 0

SW'5' - INHIBIT PRINTING TO TTY IF 1

SW'6' - DO NOT PRINT X ERRORS IF 0

SW'7' - PRINT FAILURE RATE IF 1

SW'8' - PRINT PASS COUNT IF 0

SW'9' - INHIBIT PRINTING PASS COUNT IF 1

SW'0' - INHIBIT OUTPUT TO LINE PRINTER IF 0

SW'1' - OUTPUT TO LINE PRINTER IF 1

EDTOS PRESUMES THAT ALL SWITCHES ARE 0. RETURN IS MADE TO THE OPER SYS UPON AN ERROR AFTER FIRST PRINTING THE FAILURE REPORT. IN STAND ALONE MODE, THE OPERATOR IS ASKED TO PRESET THESE SWITCHES (THE MACHINE HALTS FOR HIM TO DO SO) AND PRESS CONTINUE. THIS PRESET VALUE CAN BE CHANGED AT ANY TIME BY SETTING DATA SWITCH #0 TO A ONE. THE PROGRAM WILL THEN FOLLOW ACTUAL SETTINGS.

STAND ALONE STARTING ADDRESS = 208

IF 'CAT1' OR 'KITTEN' WAS LOADED FROM DTOS AND RESTR1 WAS NEEDED, THEN USE AS FOLLOWS:

STARTING ADDR = 170 (FOR START WITH NO 'CAT1')

STARTING ADDR = 171 (FOR START WITH 'CAT1')

MONITOR LOCATION 203 TO CHECK THE CURRENT PASS COUNT

MONITOR LOCATION X6000 TO MAKE SURE THAT 'CAT1' OR 'KITTEN' IS RUNNING. IN CASES WHERE PROGRAM IS STARTED WITH 'CAT1' OR 'KITTEN' LOCATION X6000 WILL SHOW A PATTERN CHANGING FROM ZEROS TO ALL ONES TO AN INC/SNAP PATTERN.

(X= THE NUMBER OF THE HIGHEST MEMORY MODULE IN THE SYSTEM AND MAY BE A VALUE 0 - 7)

10005 ECL21

```

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

```

OPERATING PROCEDURE/OPERATOR INPUT  
-----

15. LOAD THE PROGRAM VIA THE BINARY LOADER OR INSERT A  
PRELOADED MEMORY MODULE.  
15.1 SET SWITCHES TO 200.  
15.2 PRESS START.  
15.3 PROGRAM WILL HALT AFTER PRINTING THE MESSAGE  
15.4 "SET DATA SWITCHES AND PRESS CONTINUE".  
15.4.1 SET DATA SWITCHES AND PRESS CONTINUE.  
15.4.2 SET REQUIRED SWITCH SETTINGS AND PRESS CONTINUE.  
15.4.3 THE PROGRAM WILL RUN UNTIL MANUALLY STOPPED. IN CASE  
OF MALFUNCTIONING, THE PROGRAM WILL PRINT ERROR  
MESSAGE AND TAKE APPROPRIATE ACTION AS PER THE SW  
SETTINGS.

16. PROGRAM OUTPUT/ERROR DESCRIPTION  
-----

16.1 FOR ANY ERRORS DETECTED, THE PROGRAM WILL PRINT ERROR  
16.1.1 REPORT OR X FAILURES DEPENDING UPON THE SW SETTINGS.  
16.2 ERROR REPORT CONSISTS OF ALL ACCUMULATORS, CARRY,  
16.2.1 RELOCATED PROGRAM COUNTER OF THE TEST THAT IS FAILING  
16.2.2 AND PC IN THE LISTING AT THE TIME OF FAILURE.  
16.3 THE PROGRAM WILL LOOP IN THE TEST THAT IS FAILING IF  
16.3.1 SW"1" IS 0.  
16.4 THE PRINTING OF ERROR REPORT CAN BE ABORTED BY SETTING  
16.4.1 SW"2" TO 1.  
16.5 IF LOOPING OCCURS IN THE PROGRAM, STOP THE COMPUTER  
16.5.1 AND CHECK LOCATION 201 TO FIND OUT THE TEST THAT WAS  
16.5.2 RUNNING BEFORE THE LOOPING OCCURRED.

10006 ECL21

```

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

```

PROGRAM DESCRIPTION/THEORY OF OPERATION  
-----

17. EACH TEST IS COMPLETE IN ITSELF, SO THE PROGRAM CAN  
17.1 BE STARTED FROM ANY TEST WITHOUT CAUSING ANY  
INITIALIZATION ERRORS.  
17.2 WHEN IECLIPSE21 IS STARTED AT LOCATION 200 OR BY  
DTOS, IT WILL SIZE UP THE MEMORY AND WILL PRINT  
THE TOP OF THE MEMORY.  
17.2.1 AFTER SETTING UP THE SWITCHES AND PRESSING CONTINUE,  
17.2.2 THE EXERCISER WILL RUN THE FIRST PASS VERY FAST. IN  
17.2.3 THE FIRST PASS EACH TEST IS RUN ONLY ONCE. ALL OTHER  
17.2.4 PASSES WILL TAKE MORE TIME AS EACH TEST IS RUN  
17.2.5 ACCORDING TO THE TEST COUNT SPECIFIED IN EACH TEST.  
17.2.6 REFER TO THE LISTING TO FIND OUT THE INFORMATION  
17.2.7 ABOUT EACH TEST.

RESTRICTIONS/MISC  
-----

18. CERTAIN INSTRUCTIONS LIKE BLM, XCY, BAM, ETC.,  
18.1 DO ALLOW INTERRUPTS TO OCCUR DURING THEIR  
EXECUTION. THIS FEATURE OF THOSE INSTRUCTIONS IS  
NOT CHECKED IN THIS TEST.

```

10007 ECL21
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

*** MACRO DEFINITIONS ***

MACRO LOOP
JSR @ENTLO
X

MACRO SETUP
JSR @ENTIN
A1
X

MACRO RAND
JSR @ENTRA
IC(CA0)=RANDOM #
X

MACRO BRAND
JSR @ENTRB
X

MACRO ERROR
JMP *+2
**
JMP *+3
**
STA 3,AC3
**
JSR @ENTER
X

MACRO STACK
JSR @ISTK
A1
X

MACRO PSHSP
SETUP 20
STACK PSHSA1
MOV 0,0
PSH A1,A1
LDA 0,0P
LDA 1,0BEG
MOV 0,0,SZC
SUB 0,1,SZR
PSHSA1: ERROR
X

MACRO TRAPER
JSR @ITRP
A1
A2
A3
A4
X

ITERATE TEST ROUTINE....

INITIALIZE TEST....

INITIALIZE STACK
IF AULT ADDRESS IS A1

IF A1 ON THE STACK
IF TEST THE STACK POINTER
IF FOR A INCREMENT OF 0N1
IC(CA0)=STACK POINTER+C(40)
IC(CA1)=CORRECT
IC(CARRY) SHOULD NOT CHANGE

INIT FOR TRAP INSTRUCTION
IF STACK FAULT, GO TO A1
IF TRAP ORIGIN IN CONTENTS OF A2
IF THE CAP NUMBER IS A3
IF SUBROUTINE ADDRESS IS A4
IF STACK POINTERS SET TO C(0BEG)

```

10008 FCL21

```

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

TRST
LDA 0,0BEG
LDA 1,A1,A1
ADD 0,1
XOP A1,A1,A2
SUB 2,3,SNR
SUB 1,2,S7R
ERROR
X

ZERDAG
SUB 0,0
SUB 1,1
SUB 2,2
SUB 3,3
X

BAMER
STA 3,AC3
**
JSR @ENTBE
X

MDERR
STA 3,AC3
**
JSR @ENTBE
X

RANST
JSR @I,RAN
RANDOM # TO AC0-2
X

```



10089 ECL21

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

```

```

.MACRO OVLIM ,+2
LDA 0,1,+2
MOVZ 0,0,SKP
A2
STA 0,0,FF
STA 0,0,FF
LDA 0,0,1
STA 0,0,SP
STA 0,0,SL
STA 0,0,FF
X

.MACRO OVT1
SETUP 000
OVLIM 000,OV10A1
LDA 1,0,1
ADD 0,1
STA 1,0,SL
SAVE 0
ERROR
LOOP
X

.MACRO UFLIM
LDA 0,0,SPBK
STA 0,0,SP
LDA 0,0,BEG
STA 0,0,SL
LDA 0,0,1
STA 0,0,FF
X

.MACRO OYTRP
LDA 0,0,BEG
STA 0,0,SP
STA 0,0,SL
LDA 1,0,05
STA 1,0,00
LDA 1,0,04
STA 1,0,0F
JMP ,+3
A2
A1
X

```

```

;THE ADDRESS TO GO TO
;ION OVERFLOW IS +2.
;SET C(AC0),STACK, AND
;FRAME POINTERS TO
;THE CONTENTS OF A1.
;TEST OF STACK OVERFLOW.

;STACK LIMIT IS A1
;GREATER THEN STACK POINTER.
;OVERFLOW SHOULD OCCURE.

;SETUP STACK IN PAGE 0
;TO UNDER FLOW ON ANY
;POP,POP,RTH,OR POPB.
;INSTRUCTION, THE FAULT
;LOCATION IS A1.

;TEST OVERFLOW OF THE STACK
;ON THE CAP INSTRUCTION.

;IF NO OV GO TO A2.
;IF OV GO TO A1.

```

10010 ECL21

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

```

```

.MACRO VECTOR
JBR 00VEC
VP
A2
A1
0
X

.MACRO 0FILL
LDA 2,0,BEG
INC 0,3
LDA 1,0,BEND
SUB 2,1
LDA 0,0,1
STA 0,0,02
BLM
X

.MACRO AGAIN
ISZ TEM
LDA 1,0,BEND
LDA 0,0,110
SUB 0,1
LDA 0,0,TEM
ADCM 0,1,0SR
JMP ,+1
X

.MACRO FIND
LDA 0,0,00
STA 2,0
SURZL 3,3
INCZL 3,3,0ZC
JMP ,+1
LDA 0,0,02
STA 0,0,00
MOVZLW 2,2,0ZC
JMP ,+5
LDA 0,0,02
LDA 0,0,00
LDA 1,0,00
STA 1,0,00
LDA 1,0,00
MOVZL 1,1
MOVZL 1,1
SUBW 1,2,0SR
SUBW 1,3,0SR
JMP ,+1
LDA 1,0,00
SUBW 0,1,0SR
JMP ,+10
X

```

```

;DISPATCH TABLE AT LOCATION
;VCYAB... ENTRY 0 SET
;TO ADDRESS A1, OTHERS
;TO ADDRESS A2.

;FILL THE BUFFER WITH
;A1.

;ADVANCE TO NEXT BUFFER
;LOCATION AND TEST FOR
;END OF BUFFER-110. IF NOT
;END OF BUFFER GO TO A1.

;FIND THE FINAL ADDRESS
;AND VALUE IN THE INDIRECT
;CHAIN. C(AC2)=FIRST
;ADDRESS AT WHICH TO
;START LOOKING.
;IF INDIRECT CHAIN IS
;A15, EXIT TO A1.

;AUTO INDEX REGISTERS
;ARE USED AS VARIABLES.

;MAKE SURE THAT FINAL
;DATA WILL NOT POINT
;TO ANY SPOT IN THE CHAIN.
;IF IT DOES, EXIT TO A1.

;C(AC2)=FINAL ADDRESS
;C(AC3)=DATA AT THAT
;ADDRESS.

```

```

10012 ECL21
01
02
03
04 000000
05 000000 000000 .LOC 0
06 000000 000000 DIRT
07 000000
08 000000 .LOC 40
09 000000
10 000000 000000
11 000000 000000
12 000000 000000
13 000000 000000 *+1
14 000000 000000 HALT
15 000000 000000 .LOC 45
16 000000 000000 EGGS
17
18
19 000000-000010 .ZREL
20 000000-000010 .BLK 8.
21
22 00010-000120 EGGS: EGGS
23 00011-000000 MEMTOP: 0
24 00012-000000 ICAT: 0
25
26
27
28 000170 .LOC 170
29 00170 102441 OFF: SURO 0,0,SKP
30 00171 102000 ON: ADC 0,0
31 00172 142470 ESTA 0,CATSW
32 00173 005735
33 00174 002175 JMP 0,+1
34 00175 000533 SNMESS
35
36
37
38
39 00176 000002 .LOC 176
40 00176 000002 .BLK 2
41
42
43
44 00200 002202 DT000: JMP 200
45 00201 000000 ITRET: 0
46 00202 000500 BGNADR:
47 00203 000000 NSTRY
48 00204 000000 PASS1: 0
49 00205 000001 PASSIN: 1
50 00206 000001 PASSVL: 1
51
52 00207 000000 ITR: 0
53 00207 000000 ITRCT: 0
54 00210 000000 ITRER: 0
55 00211 000000 ITRER: 0
56 00212 000000 ERRET: 0
57 00213 000000 LOPRET: 0
58

```

```

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

```

```

*****DIAGNOSTIC PROGRAM PREAMBLE*****
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58

```

10013 ECL21

```

01 1 *****LOCAL ZREL*****
02
03
04 00214 006334 MINLOC: PRGEND+100
05 00215 007234 MAXLOC: PRGEND+1000
06
07
08 / POINTERS TO SUBROUTINES
09 00216 001172 IPOCT: POCT
10 00217 001160 IPOEC: POEC
11 00220 001233 INESS: MESS
12 00221 000646 ISTK1 STK
13 00222 000616 ITRP: TRP
14 00223 001133 I.RAN: .RAN
15 00224 001144 ISIZE: SIZE
16
17 00225 002014 IBEG: BEGIN
18 00226 006334 BBEG: PRGEND+100
19 00227 006634 BEND: PRGEND+400
20
21 00230 000766 ENTIN: INIT
22 00231 000727 ENYLO: LOP
23 00232 001014 ENTER: ERRB
24 00233 001003 ENYBE: ERRA
25 00234 001115 ENTRAI: .RAND
26 00235 001113 ENTRBI: .BRAN
27
28 / TEMPORARYS FOR TESTS AND SUBROUTINES
29
30 00240 .LOC 240
31 00240 000000 TEM:
32 00241 005000 BO: 249+20
33 00242 005017 B1: 249+20+17
34 00243 004777 BH1: 249+20-1
35 00244 005020 B20: 249+20+20
36
37 00245 000000 AC0:
38 00246 000000 AC1:
39 00247 000000 AC2:
40 00250 000000 ACS:
41 00231 000000 CBY:
42 00252 000000 EPC:
43
44 00253 000000 CHARET:
45 00254 000000 CHORZ1:
46 00255 000000 PDERET:
47 00256 000000 MESRET:
48
49 00257 000000 MTEM:
50 00260 000000 BOK0:
51 00261 000000 BOK1:
52 00262 000000 BOK2:
53 00263 000000 BOK3:
54 00264 000000 K01:
55 00265 000000 BANSUM:
56 00266 000000 BANRET:
57 00267 000000 BANXZ:
58 00270 000000 BANXR:
59 00271 000000 PSCRVI:

```

10014 ECL21

```

01
02 00272 177777 MI:
03 00273 000000 TEM1:
04 00274 000000 TEM2:
05 00275 000000 TEM3:
06
07 00276 000000 RAN:
08 00277 000000 SRAN:
09
10
11 00300 000000 ORIGR:
12 00301 000000 ORIG1:
13 00302 000000 ORIG2:
14 00303 000000 ORIG3:
15 00304 000000 OK0:
16 00305 000000 OK1:
17 00306 000000 OK2:
18 00307 000000 OK3:
19 00310 000000 PHULR:
20 00311 000000 MDRET:
21 00312 000000 PDIVR:
22
23
24 00313 000000 RANRZ:
25 00314 000000 RAPT1:
26 00315 000000 BANT0:
27 00316 000000 FROM:
28 00317 000000 T00:
29 00320 000000 BAC0:
30 00321 000000 BAC1:
31 00322 000000 BAC2:
32 00323 000000 BAC3:
33
34 00324 006134 UP1
35 00325 005400 JSR3:
36 00326 005400 JSR2:
37 00327 000010 PRK:

```

10015 ECL21

```

01
02 00328 000000 VCTAB:
03 00329 000000 JSR:
04 00330 000000 JSR:
05 00331 000000 JSR:
06 00332 000000 JSR:
07 00333 000000 JSR:
08 00334 000000 JSR:
09 00335 000000 JSR:
10 00336 000000 JSR:
11 00337 000000 JSR:
12 00338 000000 JSR:
13 00339 000000 JSR:
14 00340 000000 JSR:
15 00341 000000 JSR:
16 00342 000000 JSR:
17 00343 000000 JSR:
18 00344 000000 JSR:
19 00345 000000 JSR:
20 00346 000000 JSR:
21 00347 000000 JSR:
22 00348 000000 JSR:
23 00349 000000 JSR:
24 00350 000000 JSR:
25 00351 000000 JSR:
26 00352 000000 JSR:
27 00353 000000 JSR:
28 00354 000000 JSR:
29 00355 000000 JSR:
30 00356 000000 JSR:
31 00357 000000 JSR:
32 00358 000000 JSR:
33 00359 000000 JSR:
34 00360 000000 JSR:
35 00361 000000 JSR:
36 00362 000000 JSR:
37 00363 000000 JSR:

```

10016 ECL21

```

01
02 00364 000000 VCTAB:
03 00365 000000 JSR:
04 00366 000000 JSR:
05 00367 000000 JSR:
06 00368 000000 JSR:
07 00369 000000 JSR:
08 00370 000000 JSR:
09 00371 000000 JSR:
10 00372 000000 JSR:
11 00373 000000 JSR:
12 00374 000000 JSR:
13 00375 000000 JSR:
14 00376 000000 JSR:
15 00377 000000 JSR:
16 00378 000000 JSR:
17 00379 000000 JSR:
18 00380 000000 JSR:
19 00381 000000 JSR:
20 00382 000000 JSR:
21 00383 000000 JSR:
22 00384 000000 JSR:
23 00385 000000 JSR:
24 00386 000000 JSR:
25 00387 000000 JSR:
26 00388 000000 JSR:
27 00389 000000 JSR:
28 00390 000000 JSR:
29 00391 000000 JSR:
30 00392 000000 JSR:
31 00393 000000 JSR:
32 00394 000000 JSR:
33 00395 000000 JSR:
34 00396 000000 JSR:
35 00397 000000 JSR:
36 00398 000000 JSR:
37 00399 000000 JSR:

```

10017 ECL21

```

01
02 00400 000000 VCTAB:
03 00401 000000 JSR:
04 00402 000000 JSR:
05 00403 000000 JSR:
06 00404 000000 JSR:
07 00405 000000 JSR:
08 00406 000000 JSR:
09 00407 000000 JSR:
10 00408 000000 JSR:
11 00409 000000 JSR:
12 00410 000000 JSR:
13 00411 000000 JSR:
14 00412 000000 JSR:
15 00413 000000 JSR:
16 00414 000000 JSR:
17 00415 000000 JSR:
18 00416 000000 JSR:
19 00417 000000 JSR:
20 00418 000000 JSR:
21 00419 000000 JSR:
22 00420 000000 JSR:
23 00421 000000 JSR:
24 00422 000000 JSR:
25 00423 000000 JSR:
26 00424 000000 JSR:
27 00425 000000 JSR:
28 00426 000000 JSR:
29 00427 000000 JSR:
30 00428 000000 JSR:
31 00429 000000 JSR:
32 00430 000000 JSR:
33 00431 000000 JSR:
34 00432 000000 JSR:
35 00433 000000 JSR:
36 00434 000000 JSR:
37 00435 000000 JSR:

```

10018 ECL21

```

01
02 00436 000000 VCTAB:
03 00437 000000 JSR:
04 00438 000000 JSR:
05 00439 000000 JSR:
06 00440 000000 JSR:
07 00441 000000 JSR:
08 00442 000000 JSR:
09 00443 000000 JSR:
10 00444 000000 JSR:
11 00445 000000 JSR:
12 00446 000000 JSR:
13 00447 000000 JSR:
14 00448 000000 JSR:
15 00449 000000 JSR:
16 00450 000000 JSR:
17 00451 000000 JSR:
18 00452 000000 JSR:
19 00453 000000 JSR:
20 00454 000000 JSR:
21 00455 000000 JSR:
22 00456 000000 JSR:
23 00457 000000 JSR:
24 00458 000000 JSR:
25 00459 000000 JSR:
26 00460 000000 JSR:
27 00461 000000 JSR:
28 00462 000000 JSR:
29 00463 000000 JSR:
30 00464 000000 JSR:
31 00465 000000 JSR:
32 00466 000000 JSR:
33 00467 000000 JSR:
34 00468 000000 JSR:
35 00469 000000 JSR:
36 00470 000000 JSR:
37 00471 000000 JSR:

```

```

10015 ECL21
01 000500
02
03
04
05 00500 02577 NSTR1 IORST
06 00501 006224 JSR
07 00502 150400 NEG
08 00503 150000 COM
09 00504 050011 STA
10 00505 122470 ELDA
11 00506 005422 MOV
12 00507 101005 JMP
13 00510 000415 NOCAT
14 00511 122470 ELDA
15 00512 005522 LDA
16 00513 024052 ADD
17 00514 123000 ADD
18 00515 112033 ADCZ#
19 00516 000407 JMP
20
21 00517 132400 SUB
22 00520 024051 LDA
23 00521 147000 ADD
24 00522 044012 STA
25 00523 050215 STA
26 00524 000407 JMP
27
28 00525 024050 NOCAT: LDA
29 00526 132400 SUB
30 00527 126400 SUB
31 00530 146470 ESTA
32 00537 005377 STA
33 00532 050215 STA
34

```

```

*****SITE SYSTEM & RESERVE MEMORY*****
LOC 500
*****
NSTR1 IORST
PISIZE 2,2
STORE ADDRESS OF LAST... 2,2
LOCATION IN MEMTOP. 2,2
TEST IF DPOS SET CATSM 0,CATSM
IF 0 CAT WAS NOT LOADED. 0,0,SNR
IF CAT WAS LOADED 0,PRGEND
TEST FOR SUFFICIENT MEMORY TO RUN CAT 1,*,1777
MEMORY TO RUN CAT 1,0
NOCAT# 0,2,SNR
NOCAT
SUB 1,2
LDA 1,*,400
ADD 2,1
I,CAT 1,I,CAT
2,MAXLOC STA
SMESS STA
JMP STA
LDA 1,*,200
SUB 1,2
I,CATSM ESTA
STA 2,MAXLOC

```

```

*****OUTPUT STRT MESSAGE & READ SWITCHES*****
01
02
03 00533 006220 SMESS: JSR
04 00534 001750 LDA
05 00535 024011 MOV
06 00536 101040 NOVO
07 00537 006210 JSR
08 00540 006220 JSR
09 00541 001770 SUR
10 00542 126400 STA
11 00543 044203 STA
12
13 00544 126470 ELDA
14 00546 005361 MOV
15 00547 125004 JMP
16 00547 000414 JMP
17
18 00550 006220 JSR
19 00551 001772 JSR
20 00552 006220 JSR
21 00553 001770 HALT
22 00554 063077 JMP
23 00555 000401 JMP
24 00556 000477 READS
25 00557 142470 ESTA
26 00557 005353 ESTA
27 00551 000402 JMP

```

```

:PRINT SIZE OF MEMORY
1, MEMTOP
0, 0
PIPOCT
0, SMESS
1, 1
1, PASS
1, AUTO
1, RUNNING IN AUTO MODE?
1, 1, SZR
START
0, SMESS
0, SMESS
0, *1
0, SMREG
START

```

```

:READ NEW STATE OF SWITCHES

```

```

:PRINT SET SWITCHES MESS.

```

```

10017 ECL21
01
02 00562 006012" START: JSR @ICAT
03
04
05
06
07
08
09
10
11 00553 102400 REG1: SUB 0,0
12 00564 040210 STA 0,TRRER
13
14
15 00565 006234 RAND
16 00566 005300 JSR @ENTRA
17 00567 030215 MOVS 0,1
18 00568 142422 LDA 2,MAXLOC
19 00569 000777 SUBZ 2,0,SZC
20 00570 143000 JMP -1
21 00571 146422 SUBZ 2,1,SZC
22 00572 000777 ADD 2,0
23 00573 147000 JMP -1
24 00574 000777 ADD 2,1
25 00575 147000
26 00576 030214 BEG2: LDA 2,MINLOC
27 00577 142433 SUBZ# 2,0,SNC
28 00578 000763 JMP REG1
29 00579 146433 SUBZ# 2,1,SNC
30 00580 000761 JMP BEG1
31 00581 106433 SUBZ# 0,1,SNC
32 00582 104710 XCH 0,1
33 00583 131000 BEG3: MOV 1,2
34 00584 112400 SUB 0,2
35 00585 034847" LDA 3,5377
36 00586 172433 SUBZ# 3,2,SNC
37 00587 000752 JMP BEG1
38 00588 040226 STA 0,IBEG
39 00589 044227 STA 1,BEND
40 00590 002401 JMP 0,+1
41 00591 002014 BEGIN
42
10010 ECL21
01
02
03
04 00616 023401 TRP: LDA 0,0,1,3
05 00617 040040 STA 0,SP
06 00618 040041 AND FRAME POINTER
07 00619 020227 LDA 0,BEND
08 00620 024046" LDA 1,0,140
09 00621 122400 SUB 1,0
10 00622 040042 STA 0,SL
11 00623 021400 LDA 0,0,3
12 00624 040043 LDMY FAULT AND
13 00625 040045 FLDATING FAULT
14 00626 033401 YTRAP ORIGIN ADDRESS
15 00627 024845" LDA 1,0,10
16 00628 133000 ADD 1,2
17 00629 050044 STA 2,70
18 00630 024044" LDA 1,0,32,
19 00631 041000 STA 0,0,2
20 00632 151400 INC 2,2
21 00633 125404 INC 1,1,SZR
22 00634 000775 JMP -3
23 00635 025402 LDA 1,2,3
24 00636 133000 ADD 1,2
25 00637 025403 LDA 1,0,3
26 00638 045340 STA 1,0,32,,2
27 00639 001404 JMP 4,5
28
29
30
31
32 00640 020226 STK: LDA 0,0BEG
33 00641 100110 SBI 1,0
34 00642 040040 AND FRAME POINTER
35 00643 040041 YPOINT TO THE DATA
36 00644 040004 STA 0,ISP
37 00645 020227 LDA 0,BEND
38 00646 122400 LDA 1,0,100
39 00647 122400 SUB 1,0
40 00648 040042 STA 0,SL
41 00649 040006 LDA 0,0,3
42 00650 021400 STA 0,FF
43 00651 040045 STA 0,SF
44 00652 040043 STA 0,ISF
45 00653 040007 JMP 1,5
46 00654 001401

```

```

? TRAPER FAULT, ORIGIN, TRAP#, SUBROUTINE ADDRESS
? INITIALIZE A STACK,....
? MAKE STACK POINTER
? AND FRAME POINTER
? POINT TO THE DATA
? BUFFER -1.
? BUFFER END -100 IS
? THE STACK LIMIT
? SETUP STACK FAULT
? AND FLOATING FAULT

```

```

? *****INITIALIZING ROUTINES*****
? THESE ROUTINES GENERATE APPROPRIATLY SIZED RANDOM BUFFERS
? FOR USE BY THE TEST PROGRAMS.
? START OF PROGRAM
? MAKE C(AC0-1) MODULO
? MEMORY SIZE
? BUFFER ADDRESS MUST BE
? GREATER THAN END OF
? PROGRAM
? C(AC0)<C(AC1)
? C(AC2)=SIZE OF BUFFER
? BUFFER SIZE MUST BE
? LARGER OF GREATER
? BEGIN OF BUFFER
? END OF BUFFER

```

10019 ECL21

```

01
02
03
04 00665 054420  ISETUP A VECTOR TABLE.....
05 00666 033400  ICALL+1=ORIGIN POINTER
06 00667 021401  ICALL+2=ERROR ADDRESS
07 00668 041000  ICALL+3=CORRECT ENTRY
08 00669 155000  ICALL+4=ENTRY NUMBER
09 00670 024043*  ITABLE IS FILLED WITH ERR
10 00671 133710  IRETURNS..
11 00672 034411  IGOOD ENTRY
12 00673 021402  IPOSITION (DEVICE CODE).
13 00674 025403  ITHE GOOD ENTRY.
14 00675 133000  ADD 1,2
15 00676 041300  STA 0,-100,2
16 00677 062077  T088Y
17 00678 102000  AOC 0,0
18 00679 062077  MSKO 0
19 00680 001404  JMP 4,5
20 00681 000000  VRET: 0

```

10020 ECL21

```

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

```

\*\*\*\*\*TEST UTILITY SUBROUTINES\*\*\*\*\*

I SUBROUTINE TO INITIALIZE A TEST LOOP

INIT: INC 3,3 ITEST LOOP INITIALIZER  
STA 3,ITRET ISAVE RETURN LOCATION  
STA 0,AC0 ISAVE CONTENTS OF ACM

LDA 0,-1,3 IGET # OF ITERATIONS  
STA 0,ITR ISET ITER. VALUE  
STA 0,ITRCL ISET ITER. COUNT

SUB 3,3 ICLEAR ERROR SWITCH  
STA 3,ITRER ICLEAR ERROR COUNT  
STA 3,ITREC

LDA 3,PASS ITEST FOR FIRST PASS  
MOV 3,S,JSZR

JMP INIT1

SUBZL 3,3 ITHIS IS 1ST PASS  
STA 3,ITR ISET ITERATIONS FOR  
STA 3,ITRCL I1 LOOP ONLY.

LDA 0,AC0 IPRESTORE ACS AND  
JMP 0,ITRET IEXIT TO TEST

10021 ECL21

! SUBROUTINE TO TERMINATE A TEST LOOP

```

01 STA 3,LOPRET
02 DSZ ITRCT
03 LOP3
04 00727 054213
05 00730 014207
06 00731 00044A
07 00732 034210
08 00733 175905
09 00734 002213
10 00735 034206
11 00736 054207
12
13 00737 074477 LOP1:
14 00740 175112
15 00741 000403
16 00742 136470
17 00743 005170
18 00744 177100
19 00745 177103
20 00746 000421
21 00747 040245
22 00750 044246
23 00751 050247
24 00752 000200
25 00753 001725
26 00754 102480
27 00755 024211
28 00756 040211
29 00757 030042
30 00760 143710
31 00761 030206
32 00762 153710
33 00763 000217
34 00764 020245
35 00765 024246
36 00766 030247
37 00767 170400 LOP2:
38 00770 054211
39
40 00771 034210 LOP3:
41 00772 175004
42 00773 074477
43 00774 175112
44 00775 000403
45 00776 136470
46 00777 005170
47 01000 177113
48 01001 002201
49 01002 002213

```

10022 ECL21

! LONG AND SHORT FORM ERROR ROUTINES

```

01 STA 3,EPC
02 STA 0,AC0
03 SUBCL 0,0
04 STA 0,CRY
05 01003 054252
06 01004 040245
07 01005 102500
08 01006 040251
09 01007 021400
10 01010 175400
11 01011 101004
12 01012 000775
13 01013 000403
14
15 01014 054252
16 01015 040245
17 01016 102500
18 01017 040251
19 01020 054212
20 01021 010211
21 01022 010211
22
23 01022 020210
24 01023 110033
25 01024 000404
26
27 01025 054210
28 01026 044246
29 01027 050247
30 01030 000220
31 01031 001137
32 01032 024203
33 01033 125420
34 01034 000217
35 01035 000220
36 01036 001732
37 01037 101020
38 01040 024251
39 01041 000217
40 01042 101040
41 01043 024245
42 01044 000210
43 01045 024246
44 01046 000216
45 01047 024247
46 01050 000216
47 01051 024248
48 01052 000216
49 01053 024252
50 01054 000216
51

```

```

1023 ECL21
01
02
03 0105 03052 ERR1 LDA Z,EPC
04 0106 02052 ERR1 ISZ EPC
05 0107 02052 ERR1 LDA 0,ERR1
06 0108 142027 ADCZ 2,0,5BN
07 0109 000413 JMP ERR1
08
09 0102 03500 ERR2 LDA 3,0,2
10 0103 175113 JMP 3,3,5NC
11 0104 000485 JMP MOVLM
12 0105 05402 STA ERR2
13 0106 000220 STA 3,4,2
14 0107 000000 JSR PINESS
15 0108 000765 JMP 0
16
17 01071 025400 ERR2: LDA 1,0,3
18 01072 000216 JSR 0,POCT
19 01073 000762 JMP ERR1
20
21 01074 000220 ERR3: JSR 0,INESS
22 01075 001770 KCRLF
23 01076 024240 LDA 1,AC1
24 01077 030247 LDA 2,AC2
25 01100 136470 LDA 3,AUTO
26 005025 MOV 0,5,SNR
27 01102 175005 JMP ERR4
28 01103 000405
29
30 01104 002077 TORST
31 01105 034010- LDA 3,EGGS
32 01106 035404 LDA 3,4,3
33 01107 001400 JMP 0,3
34
35 01110 020245 ERR4: LDA 0,AC0
36 01111 034250 LDA 3,AC3
37 01112 002212 JMP 0,ERR1

```

```

1024 ECL21
01
02
03
04
05 01113 020276 .BRANS: LDA 0,RAN
06 01114 000405 JMP .RAND*4
07 01115 020276 .RAND: LDA 0,RAN
08 01116 024210 LDA 1,ITRER
09 01117 125004 MOV 1,1,SZR
10 01120 001400 JMP 0,3
11 01121 105000 MOV 0,1
12 01122 125410 HXL 2,1
13 01123 107000 ADD 0,1
14 01124 125120 MOVZL 1,1
15 01125 125120 MOVZL 1,1
16 01126 123000 ADD 1,0
17 01127 024041- LDA 1,33031
18 01130 123000 ADD 1,0
19 01131 040276 STA 0,RAN
20 01132 001400 JMP 0,3
21
22
23 01133 054303 .RANS: STA 3,ORIG3
24 JSR RAND
25 01134 000234 JSR 0,ENTRA
26 01135 105700 INCS 0,1
27 01136 110400 NEG 0,2
28 01137 133000 ADD 1,2
29 01140 040300 STA 0,ORIG0
30 01141 044301 STA 1,ORIG1
31 01142 050302 STA 2,ORIG2
32 01143 002303 JMP 0,ORIG3
33
34
35
36
37 01144 030214 SIZE: LDA 2,MINLOC
38 01145 151400 INC 2,2
39 01146 151112 MOVLM 2,2,SZC
40 01147 000405 JMP +6
41 01150 021000 LDA 0,0,2
42 01151 051000 STA 2,0,2
43 01152 025000 LDA 1,0,2
44 01153 041000 STA 0,0,2
45 01154 132414 SUB# 1,2,SZR
46 01155 001400 JMP 0,3
47 01156 050011- STA 2,MENTOP
48 01157 000766 JMP SIZE*1
49

```

```

FARE WE IN LONG OR SHORT MODE
FOR LIST IS DONE.

LOAD ARGUMENT FROM LIST
MESSAGE PRINT?
IND. USE OCTAL PRINT

SETUP FOR MESSAGE PRINT

LOAD VALUE TO PRINT

PRESTORE AC'S
TEST, AUTO MODE?

INOP, BACK TO TEST

RETURN TO OTOS

RETURN TO TEST

```

```

/ RANDOM NUMBER GENERATOR SUBROUTINES
/ SIZE LOGICAL MEMORY
/ SIZE LOGICAL MEMORY
/ MEM IS 32K WORDS.
/ SAVE MEMORY TOP ADDRESS
/ SIZE*1

```



10025 ECL21

0026 ECL21  
01 01247 000767  
R2 JNP MESS1

```

01 ;
02 ;
03 ;*****PRINT ROUTINES*****
04 01160 17510P PDEC1      MOVL 3,3      IDECIMAL PRINT C(AC1).
05 01161 054255          STA 3,PDERET
06 01162 17520P          MOVR 3,3
07 01163 004441          JSR PDEC3
08 01164 02342R          IRESET C(CARRY) FOR ZERO SUPPRESSION
09 01165 00175P          ISET C(CARRY) IF NOT
10 01166 000144          1000.
11 01167 00012          10.
12 01170 00001          1.
13 01171 00000          0
14 01172 17510P POC1:      MOVL 3,3
15 01173 054255          STA 3,PDERET
16 01174 17520P          MOVR 3,3
17 01175 004427          JSR PDEC3
18 01176 10000R          IRESET C(CARRY) FOR ZERO SUPPRESSION
19 01177 01000R          ISET C(CARRY) IF NOT
20 01200 00100R          10000
21 01201 00010R          100
22 01202 00001R          10
23 01203 00000R          1
24 01204 00000R          0
25
26 01205 02004R-PDEC1:    LDA 0,-11
27 01206 031377          LDA 2,-1,2
28 01207 151815          MOV# 2,2,SNR
29 01210 000415          JNP PDEC3+1
30 01211 10245R          SUBC 0,0
31 01212 140452          SUBON 2,1,5ZC
32 01213 000404          JNP PDEC2
33 01214 146420          SUBZ 2,1
34 01215 01400R          INC 0,0
35 01216 000774          JNP -4
36 01217 151234 PDEC2:    MOVZ# 2,2,SZR
37 01220 152402          SUBC 2,2,5ZC
38 01221 030057-        LDA 2,0,0
39 01222 143004          ADD 2,0,FSR
40 01223 171401          INC 3,2,ASKP
41 01224 171401          INC 3,2,ASKP
42 01225 004423          JSR CHAR
43 01226 155004          MOV 2,3,SZR
44 01227 000756          JNP PDEC1
45 01230 054255          LDA 3,PDERET
46 01231 17520P          MOVR 3,3
47 01232 00140R          JNP 0,3
48
49 01233 17540R MESS:    STA 3,3
50 01234 054256          STA 3,MESRET
51 01235 031777          LDA 2,-1,3
52 01236 020047-MESS1:  LDA 0,-377
53 01237 02500R          LDA 1,0,2
54 01240 151420          INCZ 2,2
55 01241 12340R          AND 1,0
56 01242 10570R          SUBS 0,1
57 01243 004405          JSR CHAR
58 01244 121005          MOV 1,0,SNR
59 01245 002256          JNP MESRET
60 01246 004402          JSR CHAR

```

```

;PRECESS WITH TAB
;EXIT, ALL DIGITS PRINTED
;FORM THE DIGIT
;AND SET C(CARRY)
;SKIP IF LAST DIGIT
;SKIP IF ZERO SUPPRESS
;MAKE COUNT INTO ASCII
;MESSAGE PRINTER
;JSR (MESS)
;MESSAGE ADDRESS
;MESSAGE ADDRESS
;PRINT....
;SKIP IF TAB EXIT
;NEXT DIGIT
;PDERET
;JSR CHAR

```

10027 ECL21  
 01 01250 175100 CHAR1  
 02 01251 054253 STA 3,3  
 03 01252 050400 STA 2,CHRSV  
 04 01253 074477 READS 5  
 05 01254 175112 MOVL# 3,3,SZC  
 06 01255 000403 JMP \*3  
 07 01256 136470 ELOA 3,SNREG  
 08 01257 004654 LDA 2,\*22000  
 09 01258 173400 AND 3,2  
 10 01259 153120 ADDZL 2,2  
 11 01260 153200 ADDCR 2,2,SNR  
 12 01261 000435 JMP REST  
 13 01262 034047- LDA 3,\*377  
 14 01263 117725 ANDZS 0,3,SNR  
 15 01264 000432 JMP REST  
 16 01265 103804 ADD 0,0,SNR  
 17 01266 000777 JMP \*1  
 18 01267 177260 ADDCR 3,3  
 19 01268 020035- LDA 0,\*211\*400  
 20 01269 102445 SUBO 3,0,SNR  
 21 01270 000430 JMP CHAR4  
 22 01271 161340 MOVOS 3,0  
 23 01272 010254 CHAR1: ISZ CHORZ  
 24 01273 151135 MOVZL# 2,2,SNR  
 25 01274 000405 JMP CHAR2  
 26 01275 001117 DOAS 0,LPT  
 27 01276 000317 SKPBZ LPT  
 28 01277 000777 NIOC \*1  
 29 01278 000217 NIOC LPT  
 30 01279 151133 CHAR2: MOVZL# 2,2,SNR  
 31 01280 000405 JMP CHAR3  
 32 01281 061111 DOAS 0,TT0  
 33 01282 000311 SKPBZ TT0  
 34 01283 000777 JMP \*1  
 35 01284 000211 NIOC TT0  
 36 01285 175400 CHAR3: INC 3,3,SNR  
 37 01286 000762 JMP CHAR1  
 38 01287 030034- LDA 2,\*212  
 39 01288 142405 SUR 2,0,SNR  
 40 01289 040254 STA 0,CHORZ  
 41 01290 000411 REST: LDA 2,CHRSV  
 42 01291 034253 LDA 3,CHARET  
 43 01292 175200 MOVR \*3  
 44 01293 001400 JMP 0,3  
 45 01294 034254 CHAR4: LDA 3,CHORZ  
 46 01295 020033- LDA 0,\*1-8  
 47 01296 114410 TOR 0,3  
 48 01297 020032- LDA 0,\*240

0020 ECL21  
 01 01331 000740 JMP CHAR1  
 02 01332 000000 CHRSV: 0  
 03 01333 000000 CHRSV: 0

TEMP SAVE FOR AC2

01 LPT/TT0 INTERFACE ROUTINE: CHARACTER PASSED IN AC0  
 02  
 03  
 04  
 05  
 06  
 07  
 08  
 09  
 10  
 11  
 12  
 13  
 14  
 15  
 16  
 17  
 18  
 19  
 20  
 21  
 22  
 23  
 24  
 25  
 26  
 27  
 28  
 29  
 30  
 31  
 32  
 33  
 34  
 35  
 36  
 37  
 38  
 39  
 40  
 41  
 42  
 43  
 44  
 45  
 46  
 47  
 48  
 49  
 50  
 51  
 52  
 53  
 54  
 55  
 56  
 57  
 58  
 59  
 60

ISAVE RETURN ADDR.  
 ISAVE AC2  
 IREAD SWITCHES INTO AC3  
 ITEST SWITCH 0  
 ISW0 SET  
 ISW0 CLEAR, DEFAULT  
 IMASK SW2 & SW3 INTO  
 IPACK FROM AC3  
 ILEFT JUSTIFY SW2  
 ICOMPLEMENT SW2  
 INO OUTPUT, RETURN  
 IMASK CHARACTER INTO L-BYTE  
 IOF ACS, CLEAR CARRY.  
 IIF NULL CHAR. RETURN  
 IDETERMINE REQUIRED  
 ISTATE OF PARITY BIT &  
 IINSERT IT  
 ITEST FOR TAB  
 ITRUE: SETUP TAB SIMULATION  
 IRESTORE CHARACTER TO R-BYTE AC0  
 ISET CARRY, BUMP LINE COUNT.  
 ISEND TO LPT?  
 INOP. MUST BE TT0  
 IO.K. FETCH CHARACTER  
 IWAIT FOR DONE  
 ICLEAR DEVICE  
 ISEND TO TT0?  
 IO.K. SEND CHARACTER  
 IWAIT FOR DONE  
 ICLEAR DEVICE  
 IIF TABBING, AND NOT  
 IFINISHED, LOOP.  
 ITEST FOR CR/LF  
 ITRUE: ZERO LINE COUNT  
 IOTHERWISE RETURN  
 ISET UP TO TAB  
 IC(ACS) IS TWO'S COMPLEMENT  
 IOF # OF SPACES NEEDED

1029 ECL21

01 0030 ECL21 151317  
02 01333 005215 PAMES1 .TXTE I<15><12>PASS I 02 043711  
03 040520 141640 03 141640  
04 051523 046450 04 046450  
05 000240 030303 05 030303  
06 01337 005215 ERMSG1 .TXTE I<15><12>ERROR IN PASS: I 06 031455  
07 151305 120251 07 120251  
08 147722 000000 08 000000  
09 120322 005215 BAHM2: .TXTE I<15><12>GOOD C(AC0-3) I 09 01406  
10 047311 147507 10 147507  
11 050240 042317 11 042317  
12 051501 141640 12 141640  
13 035123 040450 13 040450  
14 000240 030303 14 030303  
15 051501 031455 15 031455  
16 01331 152102 BTD ADDR G000 BAD WORD <15><12> I 16 120251  
17 094717 000000 17 000000  
18 042101 005215 BAHM3: .TXTE I<15><12>BAD C(AC0-3) I 18 01447  
19 151104 040502 19 040502  
20 043411 120104 20 120104  
21 147717 024303 21 024303  
22 004504 141501 22 141501  
23 040502 026400 23 026400  
24 004504 124403 24 124403  
25 147727 000240 25 000240  
26 042322 005215 BAHM4: .TXTE I<15><12>BAM" SET C(CARRY) I 26 01457  
27 100040 041042 27 041042  
28 040502 040501 28 040501  
29 120042 120042 29 120042  
30 142523 142523 30 142523  
31 120324 120324 31 120324  
32 024303 040703 32 040703  
33 040703 151302 33 151302  
34 042101 151302 34 151302  
35 151104 124531 35 124531  
36 043411 000000 36 000000  
37 147717 040504 37 040504  
38 004504 005215 BAHM5: .TXTE I<15><12> FROM Y0 ORIG C(AC0) GOOD BAH<15><12> I 38 01472  
39 147727 040504 39 040504  
40 042322 046717 40 046717  
41 100040 152011 41 152011  
42 000012 004717 42 004717  
43 042322 147727 43 147727  
44 141501 147411 44 147411  
45 051447 144222 45 144222  
46 040640 004507 46 004507  
47 152306 024303 47 024303  
48 151305 141501 48 141501  
49 041240 124400 49 124400  
50 046501 043411 50 043411  
51 144640 147717 51 147717  
52 051510 004504 52 004504  
53 151324 040502 53 040502  
54 141525 100504 54 100504  
55 144724 000012 55 000012  
56 047317 005215 EDIVH: .TXTE I<15><12> AC2<15><12> ACW AC1 AC2<15><12> I 56 01515  
57 153640 144904 DIVX ACW AC1 AC2<15><12> I 57 01516  
58 147722 154120 58 154120  
59 043411 040411 59 040411  
60 01425 005215 <15><12>ORIG C(AC0-3) I 60 030303

| 0031 ECL21  | 0032 ECL21   | 0033 ECL21 |
|---|--|------------|
| 01 040411   | 01 144504  |            |
| 02 130703   | 02 064526  |            |
| 03 040411   | 03 141501  |            |
| 04 131303   | 04 064460  |            |
| 05 065215   | 05 141501  |            |
| 06 01527 151317 ORIG I  | 06 064601  |            |
| 07 043711   | 07 141501  |            |
| 08 060011   | 08 106602  |            |
| 09 01532 065215 BAHM7: .TXTE I<15><12>SHOULD BE ZERO, C(AC0=3)= I | 09 01626 147412 ORIG I                                       |            |
| 10 041103   | 10 147422  |            |
| 11 052117   | 11 064507  |            |
| 12 042314   | 12 060000  |            |
| 13 041240   | 13 01632 065215 MDH4: .TXTE I<15><12>DIV/MUL AC0 AC1 AC2     |            |
| 14 120305   | 14 144504  |            |
| 15 142332   | 15 127526  |            |
| 16 147722   | 16 052315  |            |
| 17 120254   | 17 064714  |            |
| 18 024303   | 18 141501  |            |
| 19 141501   | 19 064460  |            |
| 20 066400   | 20 141501  |            |
| 21 124463   | 21 064601  |            |
| 22 120275   | 22 141501  |            |
| 23 060000   | 23 01644 106602 <15><12>ORIG I                               |            |
| 24 01551 065215 BAHM8: .TXTE I<15><12> BAD                        | 24 147412  |            |
| 25 01552 042101 ADDR GOOD   | 25 144722  |            |
| 26 151104   | 26 064507  |            |
| 27 043411   | 27 060000  |            |
| 28 147717   | 28 01651 065215 SDVM1: .TXTE I<15><12>SDIV AC0 AC1 AC2       |            |
| 29 064504   | 29 042103  |            |
| 30 040502   | 30 053311  |            |
| 31 064504   | 31 040411  |            |
| 32 040502   | 32 030303  |            |
| 33 064515   | 33 040411  |            |
| 34 040700   | 34 130703  |            |
| 35 146311   | 35 040411  |            |
| 36 042305   | 36 131303  |            |
| 37 106640   | 37 01662 065215 <15><12>ORIG I                               |            |
| 38 060012   | 38 151317  |            |
| 39 01570 065215 MULM1: .TXTE I<15><12> AC0 AC1 AC2<15><12>        | 39 043711  |            |
| 40 01571 052315 MUL AC0 AC1 AC2<15><12>                           | 40 060011  |            |
| 41 064714   | 41 01666 065215 SHM1: .TXTE I<15><12>SMUL AC0 AC1 AC2        |            |
| 42 141501   | 42 040523  |            |
| 43 064460   | 43 146125  |            |
| 44 141501   | 44 040411  |            |
| 45 064861   | 45 030303  |            |
| 46 141501   | 46 040411  |            |
| 47 106602   | 47 130703  |            |
| 48 01601 147412 ORIG I  | 48 040411  |            |
| 49 144722   | 49 131303  |            |
| 50 064507   | 50 01677 065215 <15><12>ORIG I                               |            |
| 51 060000   | 51 151317  |            |
| 52 01605 065215 MULM2: .TXTE I<15><12>BAD I                       | 52 043711  |            |
| 53 040502   | 53 060011  |            |
| 54 064504   | 54 01783 065215 BLMH5: .TXTE I<15><12> GOOD BAD 8LM FAILED I |            |
| 55 000000   | 55 01784 151306 FROM TO WORD                                 |            |
| 56 01611 065215 MULM3: .TXTE I<15><12>GOOD I                      | 56 046717  |            |
| 57 147507   | 57 152011  |            |
| 58 042317   | 58 064717  |            |
| 59 060011   | 59 147727  |            |
| 60 01615 065215 DIVM1: .TXTE I<15><12>DIV AC0 AC1 AC2<15><12>     | 60 042322  |            |

0034 ECL21  
01 051523  
02 141640  
03 047317  
04 147724  
05 052516  
06 000305

0033 ECL21  
01 043411  
02 147717  
03 004504  
04 040502  
05 004504  
06 146102  
07 120115  
08 040700  
09 146311  
10 042305  
11 000240  
12 01725 005215 PERCENT: .TXTE I<15><12>X FAIL=I  
13 120245  
14 040700  
15 146311  
16 000275  
17 01732 005215 HEADER: .TXTE I<15><12><15><12>  
18 01734 005215 CRY AC0 AC1 AC2 AC3 LISTING <15><12>I  
19 01734 013003  
20 004931  
21 141501  
22 004400  
23 141501  
24 004651  
25 141501  
26 004662  
27 141501  
28 004405  
29 144714  
30 152125  
31 047311  
32 004507  
33 005215  
34 000000  
35 01754 005215 MESIZ: .TXTE I<15><12>LAST LOGICAL ADDRESS=I  
36 000714  
37 152125  
38 146240  
39 043717  
40 141711  
41 146101  
42 040040  
43 042104  
44 142702  
45 051523  
46 000275  
47 01770 005215 KCRLF: .TXTE I<15><12>I  
48 000000  
49 01772 142523 SETSM: .TXTE ISET DATA SWITCHS AND PRESS CONTINUEI  
50 120324  
51 040504  
52 040724  
53 051640  
54 144727  
55 141724  
56 051510  
57 040640  
58 042116  
59 050240  
60 142722

```

10035 ECL21
01 BEGIN:
02 I *****FIRST TEST*****
03 I
04 I
05 I
06 I
07 I
08 I
09 I
10 I
11 I
12 I
13 I
14 I
15 I
16 I
17 I
18 I
19 I
20 I
21 I
22 I
23 I
24 I
25 I
26 I
27 I
28 I
29 I
30 I
31 I
32 I
33 I
34 I
35 I
36 I
37 I
38 I
39 I
40 I
41 I
42 I
43 I
44 I
45 I
46 I
47 I
48 I
49 I
50 I
51 I
52 I
53 I
54 I
55 I
56 I
57 I
58 I

PS0:
PSHSP 0
SETUP 20
JSR #ENTIN
20
STACK PSHS0
JSR #ISTK
MOV# 0,0
PUSH AC 0 ON THE STACK
PSH 0,0
FOR A INCREMENT OF ONE
LDA 0,SP
IC(AC0)=STACK POINTER+C(40)
IC(AC1)=CORRECT
IC(CARRY) SHOULD NOT CHANGE
ERROR
LOOP
ITERATE TEST ROUTINE....

PS1:
PSHSP 1
SETUP 20
JSR #ENTIN
20
STACK PSHS1
JSR #ISTK
MOV# 0,0
PUSH AC 1 ON THE STACK
PSH 1,1
FOR A INCREMENT OF ONE
LDA 0,SP
IC(AC0)=STACK POINTER+C(40)
IC(AC1)=CORRECT
IC(CARRY) SHOULD NOT CHANGE
ERROR
LOOP
ITERATE TEST ROUTINE....

PS2:
PSHSP 2
SETUP 20
JSR #ENTIN
20
STACK PSHS2
JSR #ISTK
MOV# 0,0
PUSH AC 2 ON THE STACK
PSH 2,2
FOR A INCREMENT OF ONE
LDA 0,SP
IC(AC0)=STACK POINTER+C(40)
IC(AC1)=CORRECT
IC(CARRY) SHOULD NOT CHANGE
ERROR
LOOP
ITERATE TEST ROUTINE....

PS3:
PSHSP 3
SETUP 20
JSR #ENTIN
20
STACK PSHS3
JSR #ISTK
MOV# 0,0
PUSH AC 3 ON THE STACK
PSH 3,3
FOR A INCREMENT OF ONE
LDA 0,SP
IC(AC0)=STACK POINTER+C(40)
IC(AC1)=CORRECT
IC(CARRY) SHOULD NOT CHANGE
ERROR
LOOP
ITERATE TEST ROUTINE....

PS4:
PSHSP 4
SETUP 20
JSR #ENTIN
20
STACK PSHS4
JSR #ISTK
MOV# 0,0
PUSH AC 4 ON THE STACK
PSH 4,4
FOR A INCREMENT OF ONE
LDA 0,SP
IC(AC0)=STACK POINTER+C(40)
IC(AC1)=CORRECT
IC(CARRY) SHOULD NOT CHANGE
ERROR
LOOP
ITERATE TEST ROUTINE....

PS4A:
PSHSP 4
SETUP 20
JSR #ENTIN
20
STACK PSHS4A
JSR #ISTK
MOV# 0,0
PUSH AC 4 ON THE STACK
PSH 4,4
FOR A INCREMENT OF ONE
LDA 0,SP
IC(AC0)=STACK POINTER+C(40)
IC(AC1)=CORRECT
IC(CARRY) SHOULD NOT CHANGE
ERROR
LOOP
ITERATE TEST ROUTINE....

PS4I:
PSHSP 4
SETUP 20
JSR #ENTIN
20
STACK PSHS4I
JSR #ISTK
MOV# 0,0
PUSH AC 4 ON THE STACK
PSH 4,4
FOR A INCREMENT OF ONE
LDA 0,SP
IC(AC0)=STACK POINTER+C(40)
IC(AC1)=CORRECT
IC(CARRY) SHOULD NOT CHANGE
ERROR
LOOP
ITERATE TEST ROUTINE....

IF ANY AC IS NOT ZERO
IT WAS CHANGED VIA PSH.
I STACK FAULT ALSO COMES HERE.
ITERATE TEST ROUTINE....

```

10037 ECL21

01  
02  
03  
04 02147 006230  
05 02150 000100  
06  
07 02151 006221  
08 02152 002163  
09 02153 020000  
10 02154 030001  
11 02155 113000  
12 02156 117110  
13 02157 024000  
14 02160 034001  
15 02161 162415  
16 02162 132414  
17  
18  
19 02167 006231  
20  
21  
22 02170 006230  
23 02171 000100  
24  
25 02172 006221  
26 02173 002200  
27 02174 020000  
28 02175 030000  
29 02176 113000  
30 02177 117110  
31 02200 113110  
32 02201 107110  
33 02202 103110  
34 02203 024000  
35 02204 101003  
36 02205 132414  
37  
38  
39 02212 006231  
40

PSS:

SETUP 100  
JSR #ENTIN  
100

STACK PS6A

JSR #ISTK

PS6A

LDA 0,SP

LDA 2,#4

ADD 0,2

PSH 0,3

LDA 1,SP

LDA 3,FP

SUB# 3,0,SNR

SUB# 1,2,SZR

ERROR

LOOP

JSR #ENTLO

SETUP 100

JSR #ENTIN

100

STACK PS6A

JSR #ISTK

PS6A

LDA 0,SP

LDA 2,#10

ADD 0,2

PSH 0,3

PSH 0,2

PSH 0,1

PSH 0,0

LDA 1,SP

MOV 0,0,SNR

SUB# 1,2,SZR

ERROR

LOOP

JSR #ENTLO

TEST "PSH"  
INITIALIZE TEST....

INITIALIZE STACK  
IF AULT ADDRESS IS PSSA  
ADVANCE STACK POINTER (SP)  
BY 4 VIA "PSH", FRAME  
POINTER SHOULD NOT CHANGE.

IC(AC0)=ORIG STACK/FRAE POINTER  
IC(AC1)=STACK POINTER  
IC(AC2)=CORRECT STACK POINTER  
IC(AC3)=FRAME POINTER

ITERATE TEST ROUTINE....

TEST "PSH"  
INITIALIZE TEST....

INITIALIZE STACK  
IF AULT ADDRESS IS PS6A

14 WORDS  
13 WORDS  
12 WORDS  
11 WORD

IC(AC0)=ORIGINAL C(5P)  
IC(AC1)=STACK POINTER  
IC(AC2)=CORRECT STACK POINTER  
IC(CARRY) SHOULD BE (0).

ITERATE TEST ROUTINE....

10030 ECL21

01  
02  
03  
04 02213 006230  
05 02214 000100  
06  
07 02215 006221  
08 02216 002233  
09 02217 006221  
10 02220 002233  
11 02221 020000  
12 02222 030000  
13 02223 113000  
14 02224 103110  
15 02225 167110  
16 02226 173110  
17 02227 177110  
18 02230 024000  
19 02231 101002  
20 02232 132414  
21  
22  
23 02237 006231  
24  
25 02240 006230  
26 02241 000100  
27  
28  
29 02242 006221  
30 02243 002262  
31 02244 020000  
32 02245 030000  
33 02246 113000  
34 02247 103110  
35 02250 147110  
36 02251 153110  
37 02252 157110  
38 02253 123110  
39 02254 127110  
40 02255 133110  
41 02256 137110  
42 02257 024000  
43 02260 101002  
44 02261 132414  
45  
46  
47 02266 006231  
48

PS7:

SETUP 100  
JSR #ENTIN  
100

STACK PS7A

JSR #ISTK

PS7A

JSR #ISTK

PS7A

LDA 0,SP

LDA 2,#10

ADD 0,2

PSH 3,0

PSH 3,1

PSH 3,2

PSH 3,3

LDA 1,SP

MOV 0,0,SZC

SUB# 1,2,SZR

ERROR

LOOP

JSR #ENTLO

SETUP 100

JSR #ENTIN

100

STACK PS8A

JSR #ISTK

PS8A

LDA 0,SP

LDA 2,#20

ADD 0,2

PSH 2,0

PSH 2,1

PSH 2,2

PSH 2,3

PSH 1,0

PSH 1,1

PSH 1,2

PSH 1,3

LDA 1,SP

MOV 0,0,SZC

SUB# 1,2,SZR

ERROR

LOOP

JSR #ENTLO

TEST "PSH"  
INITIALIZE TEST....

INITIALIZE STACK  
IF AULT ADDRESS IS PS7A  
ADVANCE STACK POINTER (SP)  
BY 4 VIA "PSH", FRAME  
POINTER SHOULD NOT CHANGE.

IC(AC0)=ORIG TOTAL INCREMENT  
IC(AC1)=ORIGINAL C(5P)  
IC(AC2)=CORRECT STACK POINTER  
IC(CARRY) SHOULD BE (1).

ITERATE TEST ROUTINE....

TEST "PSH"  
INITIALIZE TEST....

INITIALIZE STACK  
IF AULT ADDRESS IS PS8A  
EXPECT NO OVERFLOW

13 WORDS  
14 WORDS  
11 WORDS  
12 WORDS  
14 WORDS  
11 WORD

IC(AC0)=ORIGINAL C(5P)  
IC(AC1)=STACK POINTER  
IC(AC2)=CORRECT STACK POINTER  
IC(CARRY) SHOULD BE (1).

ITERATE TEST ROUTINE....

10039 ECL21

```

01
02
03
04 02267 006230
05 02270 006100
06
07 02271 006221
08 02272 002302
09 02273 002040
10 02274 024042
11 02275 106400
12 02276 130440
13 02277 177110
14 02300 151444
15 02301 000776
16 02302 024040
17 02303 000042
18 02304 132414
19
20
21 02311 006231
22
23
24 02312 006230
25 02313 006100
26
27 02314 006221
28 02315 002335
29 02316 020040
30 02317 024042
31 02320 106400
32 02321 105310
33 02322 131000
34 02323 111410
35 02324 113000
36 02325 124440
37 02327 147110
38 02330 157110
39 02331 123110
40 02332 137110
41 02333 125404
42 02334 000772
43 02335 024040
44 02336 132414
45
46
47
48 02343 006231
49

```

10040 ECL21

```

01
02
03
04 02344 006230
05 02345 000100
06
07 02346 006221
08 02347 002370
09 02350 020040
10 02351 024042
11 02352 106400
12 02353 105310
13 02354 131000
14 02355 111410
15 02356 113000
16 02357 124440
17 02360 133110
18 02361 117110
19 02362 113110
20 02363 107110
21 02364 103110
22 02365 107110
23 02366 125404
24 02367 000771
25 02370 024040
26 02371 132414
27
28
29 02376 006231
30
31 02377 006230
32 02378 006100
33
34
35 02401 006221
36 02402 002412
37
38 02403 006234
39 02404 030040
40 02405 103110
41 02406 025001
42 02407 034040
43 02410 122415
44 02411 156014
45
46
47 02416 006231
48
49

```

PS91

```

SETUP 100
JSR #ENTIN
100
STACK PS9A
JSR #ISTK
PS9A
LDA 0,SP
PUSH 16 WORDS ON THE
LDA 1,SL
SUB 0,1
NEGO 1,2
PSH 3,3
INCO,2,2,SZR
JMP 1,2
LDA 1,SP
LDA 2,SL
SUB# 1,2,SZR
ERROR
LOOP
JSR #ENTLO

```

PS10:

```

SETUP 100
JSR #ENTIN
100
STACK PS10A
JSR #ISTK
PS10A
LDA 0,SP
PUSH 16 WORDS ON THE
LDA 1,SL
SUB 0,1
HXR 1,1
MOV 1,2
HXL 1,2
ADD 0,2
NEGO 1,1
PSH 2,0
PSH 2,1
PSH 2,3
PSH 1,0
PSH 1,3
INC 1,1,SZR
JMP PS10B
LDA 1,SP
SUB# 1,2,SZR
ERROR
LOOP
JSR #ENTLO

```

PS10B:

```

NEGO 1,1
PSH 2,0
PSH 2,1
PSH 2,3
PSH 1,0
PSH 1,3
INC 1,1,SZR
JMP PS10B
LDA 1,SP
SUB# 1,2,SZR
ERROR
LOOP
JSR #ENTLO

```

PS11:

```

SETUP 100
JSR #ENTIN
100
STACK PS11B
JSR #ISTK
PS11B
LDA 0,SP
PUSH 16 WORDS ON THE
LDA 1,SL
SUB 0,1
HXR 1,1
MOV 1,2
HXL 1,2
ADD 0,2
NEGO 1,1
PSH 1,2
PSH 0,3
PSH 0,2
PSH 0,1
PSH 3,0
PSH 3,1
INC 1,1,SZR
JMP PS11A
LDA 1,SP
SUB# 1,2,SZR
ERROR
LOOP
JSR #ENTLO

```

PS11B:

```

LDA 1,SP
SUB# 1,2,SZR
ERROR
LOOP
JSR #ENTLO

```

PS12:

```

SETUP 100
JSR #ENTIN
100
STACK PS12A
JSR #ISTK
PS12A
RAND
JSR #ENTRA
LDA 2,SP
PSH 0,0
LDA 1,1,2
SUB# 1,0,SZR
ADDC 2,3,SZR
ERROR
LOOP
JSR #ENTLO

```

PS12A:

```

LDA 1,SP
SUB# 1,2,SZR
ERROR
LOOP
JSR #ENTLO

```

```

;TEST #PSH#
;INITIALIZE TEST....

```

```

;INITIALIZE STACK
;FAULT ADDRESS IS PS11B
;PUSH 16 WORDS ON THE
;STACK "N" TIMES AND
;THEN CHECK THE STACK
;POINTER.

```

```

;12 WORDS
;14 WORDS
;16 WORDS
;18 WORDS
;20 WORDS
;22 WORDS
;24 WORDS
;26 WORDS
;28 WORDS
;30 WORDS
;32 WORDS
;34 WORDS
;36 WORDS
;38 WORDS
;40 WORDS
;42 WORDS
;44 WORDS
;46 WORDS
;48 WORDS
;50 WORDS
;52 WORDS
;54 WORDS
;56 WORDS
;58 WORDS
;60 WORDS
;62 WORDS
;64 WORDS
;66 WORDS
;68 WORDS
;70 WORDS
;72 WORDS
;74 WORDS
;76 WORDS
;78 WORDS
;80 WORDS
;82 WORDS
;84 WORDS
;86 WORDS
;88 WORDS
;90 WORDS
;92 WORDS
;94 WORDS
;96 WORDS
;98 WORDS
;100 WORDS

```

```

;C(CARRY)=STACK POINTER
;C(CARRY)=CORRECT
;IF C(CARRY)=1,STACK FAULT

```

```

;ITERATE TEST ROUTINE....

```

```

;TEST #PSH#
;INITIALIZE TEST....

```

```

;INITIALIZE STACK
;FAULT ADDRESS IS PS12A

```

```

;C(CARRY)=RANDOM #
;PUSH C(ACB) ON THE
;STACK (ACTUALLY THE BUFFER)
;C(ACB)=VALUE PUSHED
;C(ACB)=VALUE IN MEMORY
;C(ACB)=ORIGINAL STACK POINTER
;C(ACB)=NEW STACK POINTER
;ITERATE TEST ROUTINE....

```



```

10041 ECL21
01
02
03
04 02417 00623F
05
06 02420 00010P
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44 02470 006231

PS13:
SETUP 100
JSR #ENTIN
100
STACK PS13A
JSR #1STK
PS13A
RAND
JSR #ENTRA
LDA 2,0P
MOV# 0,1
PSH 0,1
LDA 3,2,2
LDA 2,1,2
SUM# 0,2,SNR
SUB# 1,3,SZR
ERROR
LOOP
PS13A:
JTEST "PSH"
JINITIALIZE TEST....
JINITIALIZE STACK
JFAULT ADDRESS IS PS13A
JC(AC0)=RANDOM #
J2 WORDS PUSH
JCHECK THE WORDS IN
JMEMORY....
JC(AC2-1)=ORIGINAL
JC(AC2-3)=RESULTS
ERROR
LOOP
PS13A:
ITERATE TEST ROUTINE....

PS14:
SETUP 100
JSR #ENTIN
100
STACK PS14A
JSR #1STK
PS14A
RAND
JSR #ENTRA
MOV# 0,1
MOV# 1,2
PSH 0,2
LDA 3,0P
LDA 1,-2,3
LDA 2,-1,3
LDA 3,0,3
SUB# 0,1,SNR
SUB# 0,3,SZR
ERROR
MOV# 0,0
SUM# 0,2,SZR
ERROR
LOOP
PS14A:
JTEST "PSH"
JINITIALIZE TEST....
JINITIALIZE STACK
JFAULT ADDRESS IS PS14A
JC(AC0)=RANDOM #
JPUSH ACB=2 ON THE STACK
JORIGINAL C(AC0) TO C(AC1)
JORIGINAL C(AC1) TO C(AC2)
JORIGINAL C(AC2) TO C(AC3)
JCORRECT. C(AC1) OR C(AC3) IN ERROR
JIF STACK FAULT. C(CARRY)=1
JC(AC0)=ORIG. MIDDLE WORD
ITERATE TEST ROUTINE....

PS15:
SETUP 100
JSR #ENTIN
100
STACK PS15A
JSR #1STK
PS15A
RAND
JSR #ENTRA
INC 0,1
INC 1,2
INC 2,3 JINTO THE BUFFER.
PSH 0,3
STA 3,TEM
LDA 3,0BEG
LDA 3,0,3
SUB# 0,3,SZR
ERROR
LDA 3,0BEG
LDA 3,1,3
SUB# 1,3,SZR
ERROR
LDA 3,0BEG
LDA 3,2,3
SUB# 2,3,SZR
ERROR
LDA 3,0BEG
LDA 0,TEM
LDA 3,3,3
SUB# 0,3,SZR
ERROR
MOV 0,0,SKP
ERROR
LOOP
PS15A:
JTEST "PSH"
JINITIALIZE TEST....
JINITIALIZE STACK
JFAULT ADDRESS IS PS15A
JC(AC0)=RANDOM #
JLOAD ACB=3
JAND PUSH THEM
INC 2,3 JINTO THE BUFFER.
JFIRST AC
JC(AC0)=WORD PUSHED
JC(AC3)=VALUE IN MEMORY.
JSECOND AC
JC(AC1)=WORD PUSHED
JC(AC3)=VALUE IN MEMORY.
JTHIRD AC
JC(AC2)=WORD PUSHED
JC(AC3)=VALUE IN MEMORY.
JFOURTH AC
JC(AC0)=WORD PUSHED
JC(AC3)=VALUE IN MEMORY.
ITERATE TEST ROUTINE....

```

10043 ECL21

|    |               |                |              |             |                  |
|----|---------------|----------------|--------------|-------------|------------------|
| 01 |               |                |              | 10044 ECL21 |                  |
| 02 |               |                |              |             |                  |
| 03 |               |                |              |             |                  |
| 04 | 002546 006230 | PS16:          | SETUP 100    |             |                  |
| 05 | 002547 006100 |                | JSR #ENTIN   |             | *** TEST POP *** |
| 06 |               |                | 100          |             |                  |
| 07 | 002550 006221 | STACK PS16F    | JSR #1STK    |             |                  |
| 08 | 002551 002621 | PS16F          | JSR #1STK    |             |                  |
| 09 |               | RAND           |              |             |                  |
| 10 | 002552 006234 | JSR #ENTRA     |              |             |                  |
| 11 | 002553 006275 | STA 0,TEMP     |              |             |                  |
| 12 | 002554 006040 | LDA 2,SP       |              |             |                  |
| 13 | 002555 004842 | LDA 3,SL       |              |             |                  |
| 14 | 002556 106540 | SUBOR 2,3      |              |             |                  |
| 15 | 002557 105220 | HOVER 3,1      |              |             |                  |
| 16 | 002558 133000 | ADD 1,2        |              |             |                  |
| 17 | 002559 173900 | ADD 3,2        |              |             |                  |
| 18 | 002560 115400 | STA 1,TEM      |              |             |                  |
| 19 | 002561 044540 | INC 0,3        |              |             |                  |
| 20 | 002562 101400 | INC 3,0        |              |             |                  |
| 21 | 002563 105400 | INC 0,1        |              |             |                  |
| 22 | 002564 107110 | INC 0,1        |              |             |                  |
| 23 | 002565 101400 | INC 0,0        |              |             |                  |
| 24 | 002566 101400 | D3Z TEM        |              |             |                  |
| 25 | 002567 000772 | JMP PS16A      |              |             |                  |
| 26 | 002568 000940 | LDA 0,SP       |              |             |                  |
| 27 | 002569 112654 | SUBOR# 0,2,SZR |              |             |                  |
| 28 | 002570 000420 | JMP PS16E      |              |             |                  |
| 29 | 002571 000375 | LDA 0,TEMP     |              |             |                  |
| 30 | 002572 034220 | LDA 3,SBEG     |              |             |                  |
| 31 | 002573 101801 | MOV 0,0,SKP    |              |             |                  |
| 32 | 002574 175400 | INC 3,3        |              |             |                  |
| 33 | 002575 156415 | SUB# 2,3,SNR   |              |             |                  |
| 34 | 002576 000423 | JMP PS16G      |              |             |                  |
| 35 | 002577 101400 | INC 0,0        |              |             |                  |
| 36 | 002578 025400 | LDA 1,0,3      |              |             |                  |
| 37 | 002579 106415 | SUB# 0,1,SNR   |              |             |                  |
| 38 | 002580 000772 | JMP PS16C      |              |             |                  |
| 39 |               | ERROR          |              |             |                  |
| 40 | 002613 101801 | MOV 0,0,SKP    |              |             |                  |
| 41 | 002614 101801 | ERROR          |              |             |                  |
| 42 | 002615 101801 | ERROR          |              |             |                  |
| 43 |               | LOOP           |              |             |                  |
| 44 |               | PS16G:         | JSR #ENTLO   |             |                  |
| 45 | 002625 006231 |                |              |             |                  |
| 46 |               |                |              |             |                  |
| 01 |               |                |              | 10044 ECL21 |                  |
| 02 |               |                |              |             |                  |
| 03 |               |                |              |             |                  |
| 04 |               |                |              |             |                  |
| 05 |               |                |              |             |                  |
| 06 |               |                |              |             |                  |
| 07 | 002626 006230 | PP0:           | SETUP 40     |             |                  |
| 08 | 002627 000840 |                | JSR #ENTIN   |             |                  |
| 09 |               |                | STACK PP0A   |             |                  |
| 10 | 002630 006221 | JSR #1STK      |              |             |                  |
| 11 | 002631 002644 | PP0A           |              |             |                  |
| 12 | 002632 176920 | ADZ 3,3        |              |             |                  |
| 13 | 002633 177110 | PSH 3,3        |              |             |                  |
| 14 | 002634 103210 | POP 0,0        |              |             |                  |
| 15 | 002635 101913 | MOV# 0,0,SNR   |              |             |                  |
| 16 | 002636 100914 | CON# 0,0,SZR   |              |             |                  |
| 17 |               | ERROR          |              |             |                  |
| 18 | 002643 101901 | MOV 0,0,SKP    |              |             |                  |
| 19 |               | ERROR          |              |             |                  |
| 20 |               | LOOP           |              |             |                  |
| 21 | 002650 006231 | PP0A:          | JSR #ENTLO   |             |                  |
| 22 |               |                |              |             |                  |
| 23 |               | PP1:           | SETUP 40     |             |                  |
| 24 | 002651 006230 |                | JSR #ENTIN   |             |                  |
| 25 | 002652 000840 |                | STACK PP1A   |             |                  |
| 26 | 002653 006221 | JSR #1STK      |              |             |                  |
| 27 | 002654 002667 | PP1A           |              |             |                  |
| 28 | 002655 102940 | ADCO 0,0       |              |             |                  |
| 29 | 002656 103110 | PSH 0,0        |              |             |                  |
| 30 | 002657 177210 | POP 3,3        |              |             |                  |
| 31 | 002658 101912 | MOV# 0,0,SZR   |              |             |                  |
| 32 | 002659 101912 | CON# 3,3,SZR   |              |             |                  |
| 33 | 002661 174914 | ERROR          |              |             |                  |
| 34 |               | MOV 0,0,SKP    |              |             |                  |
| 35 | 002666 101901 | ERROR          |              |             |                  |
| 36 |               | LOOP           |              |             |                  |
| 37 |               | PP1A:          | JSR #ENTLO   |             |                  |
| 38 | 002673 006231 |                |              |             |                  |
| 39 |               | PP2:           | SETUP 400    |             |                  |
| 40 |               |                | JSR #ENTIN   |             |                  |
| 41 | 002674 006230 |                | 400          |             |                  |
| 42 | 002675 000400 |                | STACK PP2A   |             |                  |
| 43 |               |                | JSR #1STK    |             |                  |
| 44 | 002676 006221 |                | PP2A         |             |                  |
| 45 | 002677 002704 |                | RAND         |             |                  |
| 46 |               |                | JSR #ENTRA   |             |                  |
| 47 | 002700 006234 |                | PSH 0,0      |             |                  |
| 48 | 002701 103110 |                | POP 1,1      |             |                  |
| 49 | 002702 127210 |                | SUB# 0,1,SZR |             |                  |
| 50 | 002703 106414 | PP2A:          | ERROR        |             |                  |
| 51 |               |                | LOOP         |             |                  |
| 52 |               |                | JSR #ENTLO   |             |                  |
| 53 | 002710 006231 |                |              |             |                  |

10043 ECL21  
 01  
 02  
 03  
 04  
 05  
 06  
 07  
 08  
 09  
 10  
 11  
 12  
 13  
 14  
 15  
 16  
 17  
 18  
 19  
 20  
 21  
 22  
 23  
 24  
 25  
 26  
 27  
 28  
 29  
 30  
 31  
 32  
 33  
 34  
 35  
 36  
 37  
 38  
 39  
 40  
 41  
 42  
 43  
 44  
 45  
 46

TEST \*PSH\*  
 INITIALIZE TEST....  
 INITIALIZE STACK  
 \*FAULT ADDRESS IS PS16F  
 \*RANDOM #  
 \*FILL THE BUFFER VIA \*PSH\*  
 \*INSTRUCTION  
 \*FINAL VALUE OF (SP)  
 \*AC'S STORED IN THE ORDER  
 \*3-0-1  
 \*THE PSH....  
 \*THE VALUES STORED IN  
 \*MEMORY WILL ASCEND  
 \*BECAUSE OF THE INC  
 \*END OF PUSHING  
 \*CHECK THE STACK POINTER  
 \*ERROR WITH SP....  
 \*CHECK THE DATA.  
 \*END OF TEST  
 \*C(CAC3)\*STACK ADDRESS  
 \*C(CAC2)\*LAST STACK ADDRESS  
 \*C(CAC1)\*STACK DATA/ERROR  
 \*C(CAC0)\*CORRECT DATA  
 \*FINAL STACK POINTER IS WRONG  
 \*C(CAC0)\*SP:C(CAC2)=CORRECT  
 \*STACK FAULT....  
 \*ITERATE TEST ROUTINE....

SETUP 400  
 JSR #ENTIN  
 100  
 STACK PS16F  
 JSR #1STK  
 PS16F  
 RAND  
 JSR #ENTRA  
 STA 0,TEMP  
 LDA 2,SP  
 LDA 3,SL  
 SUBOR 2,3  
 HOVER 3,1  
 ADD 1,2  
 ADD 3,2  
 STA 1,TEM  
 INC 0,3  
 INC 3,0  
 INC 0,1  
 INC 0,1  
 INC 0,0  
 D3Z TEM  
 JMP PS16A  
 LDA 0,SP  
 SUBOR# 0,2,SZR  
 JMP PS16E  
 LDA 0,TEMP  
 LDA 3,SBEG  
 MOV 0,0,SKP  
 INC 3,3  
 SUB# 2,3,SNR  
 JMP PS16G  
 INC 0,0  
 LDA 1,0,3  
 SUB# 0,1,SNR  
 JMP PS16C  
 ERROR  
 MOV 0,0,SKP  
 ERROR  
 MOV 0,0,SKP  
 ERROR  
 LOOP  
 JSR #ENTLO

TEST \*POP\*  
 INITIALIZE TEST....  
 INITIALIZE STACK  
 \*FAULT ADDRESS IS PP0A  
 \*PUSH (-1) FROM C(AC3)  
 \*ONTO STACK, POP IT  
 \*BACK TO AC0  
 \*C(CAC0)\*POP RESULT  
 \*C(CARRY) SHOULD STAY (0)  
 \*MOVE 0,0,SKP  
 \*ERROR  
 \*LOOP  
 \*ITERATE TEST ROUTINE....  
 \*TEST \*POP\*  
 \*INITIALIZE TEST....  
 \*INITIALIZE STACK  
 \*FAULT ADDRESS IS PP1A  
 \*PUSH (-1) FROM C(AC0)  
 \*ONTO STACK, POP IT  
 \*BACK TO AC3  
 \*C(CAC3)\*POP RESULT  
 \*C(CARRY) SHOULD STAY (1).  
 \*MOVE 0,0,SKP  
 \*ERROR  
 \*LOOP  
 \*ITERATE TEST ROUTINE....  
 \*TEST \*POP\*  
 \*INITIALIZE TEST....  
 \*INITIALIZE STACK  
 \*FAULT ADDRESS IS PP2A  
 \*RANDOM #  
 \*PUSH A RANDOM NUMBER  
 \*ONTO STACK, POP IT BACK  
 \*C(CAC0)\*CORRECT  
 \*C(CAC1)\*POP RESULT  
 \*ERROR  
 \*LOOP  
 \*ITERATE TEST ROUTINE....

SETUP 40  
 JSR #ENTIN  
 40  
 STACK PP0A  
 JSR #1STK  
 PP0A  
 ADZ 3,3  
 PSH 3,3  
 POP 0,0  
 MOV# 0,0,SNR  
 CON# 0,0,SZR  
 ERROR  
 MOV 0,0,SKP  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP2A  
 JSR #1STK  
 PP2A  
 RAND  
 JSR #ENTRA  
 PSH 0,0  
 POP 1,1  
 SUB# 0,1,SZR  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP1A  
 JSR #1STK  
 PP1A  
 ADCO 0,0  
 PSH 0,0  
 POP 3,3  
 MOV# 0,0,SZR  
 CON# 3,3,SZR  
 ERROR  
 MOV 0,0,SKP  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP2A  
 JSR #1STK  
 PP2A  
 RAND  
 JSR #ENTRA  
 PSH 0,0  
 POP 1,1  
 SUB# 0,1,SZR  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP1A  
 JSR #1STK  
 PP1A  
 ADCO 0,0  
 PSH 0,0  
 POP 3,3  
 MOV# 0,0,SZR  
 CON# 3,3,SZR  
 ERROR  
 MOV 0,0,SKP  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP2A  
 JSR #1STK  
 PP2A  
 RAND  
 JSR #ENTRA  
 PSH 0,0  
 POP 1,1  
 SUB# 0,1,SZR  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP1A  
 JSR #1STK  
 PP1A  
 ADCO 0,0  
 PSH 0,0  
 POP 3,3  
 MOV# 0,0,SZR  
 CON# 3,3,SZR  
 ERROR  
 MOV 0,0,SKP  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP2A  
 JSR #1STK  
 PP2A  
 RAND  
 JSR #ENTRA  
 PSH 0,0  
 POP 1,1  
 SUB# 0,1,SZR  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP1A  
 JSR #1STK  
 PP1A  
 ADCO 0,0  
 PSH 0,0  
 POP 3,3  
 MOV# 0,0,SZR  
 CON# 3,3,SZR  
 ERROR  
 MOV 0,0,SKP  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP2A  
 JSR #1STK  
 PP2A  
 RAND  
 JSR #ENTRA  
 PSH 0,0  
 POP 1,1  
 SUB# 0,1,SZR  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP1A  
 JSR #1STK  
 PP1A  
 ADCO 0,0  
 PSH 0,0  
 POP 3,3  
 MOV# 0,0,SZR  
 CON# 3,3,SZR  
 ERROR  
 MOV 0,0,SKP  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP2A  
 JSR #1STK  
 PP2A  
 RAND  
 JSR #ENTRA  
 PSH 0,0  
 POP 1,1  
 SUB# 0,1,SZR  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP1A  
 JSR #1STK  
 PP1A  
 ADCO 0,0  
 PSH 0,0  
 POP 3,3  
 MOV# 0,0,SZR  
 CON# 3,3,SZR  
 ERROR  
 MOV 0,0,SKP  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP2A  
 JSR #1STK  
 PP2A  
 RAND  
 JSR #ENTRA  
 PSH 0,0  
 POP 1,1  
 SUB# 0,1,SZR  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP1A  
 JSR #1STK  
 PP1A  
 ADCO 0,0  
 PSH 0,0  
 POP 3,3  
 MOV# 0,0,SZR  
 CON# 3,3,SZR  
 ERROR  
 MOV 0,0,SKP  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP2A  
 JSR #1STK  
 PP2A  
 RAND  
 JSR #ENTRA  
 PSH 0,0  
 POP 1,1  
 SUB# 0,1,SZR  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP1A  
 JSR #1STK  
 PP1A  
 ADCO 0,0  
 PSH 0,0  
 POP 3,3  
 MOV# 0,0,SZR  
 CON# 3,3,SZR  
 ERROR  
 MOV 0,0,SKP  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP2A  
 JSR #1STK  
 PP2A  
 RAND  
 JSR #ENTRA  
 PSH 0,0  
 POP 1,1  
 SUB# 0,1,SZR  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP1A  
 JSR #1STK  
 PP1A  
 ADCO 0,0  
 PSH 0,0  
 POP 3,3  
 MOV# 0,0,SZR  
 CON# 3,3,SZR  
 ERROR  
 MOV 0,0,SKP  
 ERROR  
 LOOP  
 JSR #ENTLO

SETUP 400  
 JSR #ENTIN  
 400  
 STACK PP2A  
 JSR #1STK  
 PP2A  
 RAND  
 JSR #ENTRA  
 PSH 0,0  
 POP 1,1  
 SUB# 0,1,SZR  
 ERROR  
 LOOP  
 JSR #ENTLO

```

10045 ECL21
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

PP3:
SETUP 400
JSR #ENTIN
400
STACK PP3A
JSR #ISTK
PP3A
RAND
JSR #ENTRA
PSH 0,0
POP 2,2
SUB# 0,0,5ZR
ERROR
LOOP
JSR #ENTLO

PP4:
SETUP 100
JSR #ENTIN
100
STACK PP4A
JSR #ISTK
PP4A
LDA 0,0BEG
LDA 1,0A
ADZ 0,1
SIZ 1,SP
POP 0,0
POP 1,1
POP 2,2
POP 3,3
LDA 0,SP
LDA 1,0BEG
SUB# 0,1,5NR
MOV 0,0,5ZC
ERROR
LOOP
JSR #ENTLO

PP5:
SETUP 400
JSR #ENTIN
400
STACK PP5A
JSR #ISTK
PP5A
LDA 0,0
PSH 0,3
POP 3,0
PSH 0,3
POP 3,0
PSH 0,3
POP 3,0
PSH 0,3
POP 3,0
PSH 0,3
POP 3,0
PSH 0,3
POP 3,0
PSH 0,3
POP 3,0
PSH 0,3
POP 3,0
PSH 0,3
POP 3,0
PSH 0,3
POP 3,0
LDA 0,SP
LDA 1,0BEG
ADZ# 0,1,5ZR
ERROR
LOOP
JSR #ENTLO

PP5A:
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

;TEST "POP"
;INITIALIZE TEST....
;INITIALIZE STACK
;FAULT ADDRESS IS PP3A
;C(AC0)=RANDOM #
;PUSH A RANDOM NUMBER
;ONTO THE STACK, POP IT BACK.
;C(AC0)=CORRECT
;C(AC2)=POP RESULT
;ITERATE TEST ROUTINE....
;TEST "POP"
;INITIALIZE TEST....
;INITIALIZE STACK
;FAULT ADDRESS IS PP4A
;TEST DECFMNT OF
;STACK POINTER. SET
;C(SP) TO BEGIN. OF BUFFER
;+4 AFTER 4 POPS IT
;SHOULD BE BACK TO THE
;BEGINNING
;C(AC0)=FINAL STACK
;POINTER. C(AC1)=CORRECT
;C(CARRY) SHOULD NOT
;CHANGE FROM (0).
;ITERATE TEST ROUTINE....

10046 ECL21
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

;TEST "POP"
;INITIALIZE TEST....
;INITIALIZE STACK
;FAULT ADDRESS IS PP5A
;DO 0.
;PUSH 4 ITEMS
;POP THEM BACK.
;PUSH 4 ITEMS
;POP THEM BACK.
;PUSH 4 ITEMS
;POP THEM BACK.
;PUSH 4 ITEMS
;POP THEM BACK.
;PUSH 4 ITEMS
;POP THEM BACK.
;PUSH 4 ITEMS
;POP THEM BACK.
;PUSH 4 ITEMS
;POP THEM BACK.
;PUSH 4 ITEMS
;POP THEM BACK.
;PUSH 4 ITEMS
;POP THEM BACK.
;PUSH 4 ITEMS
;POP THEM BACK.
;PUSH 4 ITEMS
;POP THEM BACK.
;C(AC0)=FINAL STACK POINTER.
;C(AC1)=CORRECT STACK
;ITERATE TEST ROUTINE....

```



```

10049 ECL21
01 03105 020226 PP01 LDA 0,BBEG
02 03106 040275 STA 0,TEMP
03 03107 040275 STACK PP08
04 03108 040275 JSR 0,ISTK
05 03109 040275 PPR 0,1
06 03110 040275 SETUP 4
07 03111 040275 JSR 0,ENTIN
08 03112 040275 4
09 03113 040275 LDA 0,TEMP
10 03114 040275 STA 0,SP
11 03115 040275 RAND
12 03116 040275 JSR 0,CENTRA
13 03117 040275 MOV 0,2
14 03118 040275 MOV 0,3
15 03119 040275 PSH 0,3
16 03120 040275 POP 1,0
17 03121 040275 PSH 0,1
18 03122 040275 SUB# 0,2,SNR
19 03123 040275 SUB# 0,2,SZR
20 03124 040275 ERROR
21 03125 040275 LOOP
22 03126 040275 JSR 0,ENTLO
23 03127 040275 ISZ TEMP
24 03128 040275 LDA 1,TEMP
25 03129 040275 LDA 2,5L
26 03130 040275 ADC# 1,2,SZR
27 03131 040275 JMP PP0A

10050 ECL21
01 03176 020226 PP10: LDA 0,BBEG
02 03177 040275 STA 0,TEMP
03 03178 040275 STACK PP10B
04 03179 040275 JSR 0,ISTK
05 03180 040275 PPR 0,1
06 03181 040275 SETUP 10
07 03182 040275 JSR 0,ENTIN
08 03183 040275 10
09 03184 040275 LDA 0,TEMP
10 03185 040275 STA 0,SP
11 03186 040275 RAND
12 03187 040275 JSR 0,CENTRA
13 03188 040275 MOV 0,3
14 03189 040275 MOV 0,3
15 03190 040275 PSH 0,3
16 03191 040275 POP 2,1
17 03192 040275 SUB# 0,2,SNR
18 03193 040275 SUB# 0,2,SZR
19 03194 040275 ERROR
20 03195 040275 LOOP
21 03196 040275 JSR 0,ENTLO
22 03197 040275 ISZ TEMP
23 03198 040275 LDA 1,TEMP
24 03199 040275 LDA 2,5L
25 03200 040275 ADC# 1,2,SZR
26 03201 040275 JMP PP10A

ITERATE TEST ROUTINE.....
NEXT ADDRESS
TEST FOR LAST

10051 ECL21
01 03220 040275 PP11: SETUP 40
02 03221 040275 JSR 0,ENTIN
03 03222 040275 40
04 03223 040275 STACK PP11A
05 03224 040275 JSR 0,ISTK
06 03225 040275 PPR 0,1
07 03226 040275 SUB 0,0
08 03227 040275 SUB 1,1
09 03228 040275 SUB 2,2
10 03229 040275 SUB 3,3
11 03230 040275 PSH 0,2
12 03231 040275 PSH 1,2
13 03232 040275 ADD# 0,1,SNR
14 03233 040275 ADD# 0,1,SZR
15 03234 040275 ERROR
16 03235 040275 PP11A: LOOP
17 03236 040275 JSR 0,ENTLO
18 03237 040275 44
19 03238 040275 006231

INITIALIZE STACK
FAULT ADDRESS IS PP10B
INITIALIZE TEST.....
JSEE PREVIOUS TEST
IC(AC0)=RANDOM #
IPUSH ACS AND #
JPOP IT BACK TO 2-1
ICHECK RESULT
ITERATE TEST ROUTINE.....
NEXT ADDRESS
TEST FOR LAST

10052 ECL21
01 03240 040275 PP12: SETUP 40
02 03241 040275 JSR 0,ENTIN
03 03242 040275 40
04 03243 040275 STACK PP12A
05 03244 040275 JSR 0,ISTK
06 03245 040275 PPR 0,0
07 03246 040275 SUB 0,0
08 03247 040275 SUB 1,1
09 03248 040275 SUB 2,2
10 03249 040275 SUB 3,3
11 03250 040275 PSH 0,2
12 03251 040275 PSH 1,2
13 03252 040275 ADD# 0,1,SNR
14 03253 040275 ADD# 0,1,SZR
15 03254 040275 ERROR
16 03255 040275 PP12A: LOOP
17 03256 040275 JSR 0,ENTLO
18 03257 040275 44
19 03258 040275 006231

INITIALIZE STACK
FAULT ADDRESS IS PP11A
ITERATE TEST ROUTINE.....
NEXT ADDRESS
TEST FOR LAST

10053 ECL21
01 03260 040275 PP13: SETUP 40
02 03261 040275 JSR 0,ENTIN
03 03262 040275 40
04 03263 040275 STACK PP13A
05 03264 040275 JSR 0,ISTK
06 03265 040275 PPR 0,0
07 03266 040275 SUB 0,0
08 03267 040275 SUB 1,1
09 03268 040275 SUB 2,2
10 03269 040275 SUB 3,3
11 03270 040275 PSH 0,2
12 03271 040275 PSH 1,2
13 03272 040275 ADD# 0,1,SNR
14 03273 040275 ADD# 0,1,SZR
15 03274 040275 ERROR
16 03275 040275 PP13A: LOOP
17 03276 040275 JSR 0,ENTLO
18 03277 040275 44
19 03278 040275 006231

INITIALIZE STACK
FAULT ADDRESS IS PP11A
ITERATE TEST ROUTINE.....
NEXT ADDRESS
TEST FOR LAST

10054 ECL21
01 03280 040275 PP14: SETUP 40
02 03281 040275 JSR 0,ENTIN
03 03282 040275 40
04 03283 040275 STACK PP14A
05 03284 040275 JSR 0,ISTK
06 03285 040275 PPR 0,0
07 03286 040275 SUB 0,0
08 03287 040275 SUB 1,1
09 03288 040275 SUB 2,2
10 03289 040275 SUB 3,3
11 03290 040275 PSH 0,2
12 03291 040275 PSH 1,2
13 03292 040275 ADD# 0,1,SNR
14 03293 040275 ADD# 0,1,SZR
15 03294 040275 ERROR
16 03295 040275 PP14A: LOOP
17 03296 040275 JSR 0,ENTLO
18 03297 040275 44
19 03298 040275 006231

INITIALIZE STACK
FAULT ADDRESS IS PP11A
ITERATE TEST ROUTINE.....
NEXT ADDRESS
TEST FOR LAST

10055 ECL21
01 03300 040275 PP15: SETUP 40
02 03301 040275 JSR 0,ENTIN
03 03302 040275 40
04 03303 040275 STACK PP15A
05 03304 040275 JSR 0,ISTK
06 03305 040275 PPR 0,0
07 03306 040275 SUB 0,0
08 03307 040275 SUB 1,1
09 03308 040275 SUB 2,2
10 03309 040275 SUB 3,3
11 03310 040275 PSH 0,2
12 03311 040275 PSH 1,2
13 03312 040275 ADD# 0,1,SNR
14 03313 040275 ADD# 0,1,SZR
15 03314 040275 ERROR
16 03315 040275 PP15A: LOOP
17 03316 040275 JSR 0,ENTLO
18 03317 040275 44
19 03318 040275 006231

INITIALIZE STACK
FAULT ADDRESS IS PP11A
ITERATE TEST ROUTINE.....
NEXT ADDRESS
TEST FOR LAST

10056 ECL21
01 03320 040275 PP16: SETUP 40
02 03321 040275 JSR 0,ENTIN
03 03322 040275 40
04 03323 040275 STACK PP16A
05 03324 040275 JSR 0,ISTK
06 03325 040275 PPR 0,0
07 03326 040275 SUB 0,0
08 03327 040275 SUB 1,1
09 03328 040275 SUB 2,2
10 03329 040275 SUB 3,3
11 03330 040275 PSH 0,2
12 03331 040275 PSH 1,2
13 03332 040275 ADD# 0,1,SNR
14 03333 040275 ADD# 0,1,SZR
15 03334 040275 ERROR
16 03335 040275 PP16A: LOOP
17 03336 040275 JSR 0,ENTLO
18 03337 040275 44
19 03338 040275 006231

INITIALIZE STACK
FAULT ADDRESS IS PP11A
ITERATE TEST ROUTINE.....
NEXT ADDRESS
TEST FOR LAST

10057 ECL21
01 03340 040275 PP17: SETUP 40
02 03341 040275 JSR 0,ENTIN
03 03342 040275 40
04 03343 040275 STACK PP17A
05 03344 040275 JSR 0,ISTK
06 03345 040275 PPR 0,0
07 03346 040275 SUB 0,0
08 03347 040275 SUB 1,1
09 03348 040275 SUB 2,2
10 03349 040275 SUB 3,3
11 03350 040275 PSH 0,2
12 03351 040275 PSH 1,2
13 03352 040275 ADD# 0,1,SNR
14 03353 040275 ADD# 0,1,SZR
15 03354 040275 ERROR
16 03355 040275 PP17A: LOOP
17 03356 040275 JSR 0,ENTLO
18 03357 040275 44
19 03358 040275 006231

INITIALIZE STACK
FAULT ADDRESS IS PP11A
ITERATE TEST ROUTINE.....
NEXT ADDRESS
TEST FOR LAST

10058 ECL21
01 03360 040275 PP18: SETUP 40
02 03361 040275 JSR 0,ENTIN
03 03362 040275 40
04 03363 040275 STACK PP18A
05 03364 040275 JSR 0,ISTK
06 03365 040275 PPR 0,0
07 03366 040275 SUB 0,0
08 03367 040275 SUB 1,1
09 03368 040275 SUB 2,2
10 03369 040275 SUB 3,3
11 03370 040275 PSH 0,2
12 03371 040275 PSH 1,2
13 03372 040275 ADD# 0,1,SNR
14 03373 040275 ADD# 0,1,SZR
15 03374 040275 ERROR
16 03375 040275 PP18A: LOOP
17 03376 040275 JSR 0,ENTLO
18 03377 040275 44
19 03378 040275 006231

INITIALIZE STACK
FAULT ADDRESS IS PP11A
ITERATE TEST ROUTINE.....
NEXT ADDRESS
TEST FOR LAST

10059 ECL21
01 03380 040275 PP19: SETUP 40
02 03381 040275 JSR 0,ENTIN
03 03382 040275 40
04 03383 040275 STACK PP19A
05 03384 040275 JSR 0,ISTK
06 03385 040275 PPR 0,0
07 03386 040275 SUB 0,0
08 03387 040275 SUB 1,1
09 03388 040275 SUB 2,2
10 03389 040275 SUB 3,3
11 03390 040275 PSH 0,2
12 03391 040275 PSH 1,2
13 03392 040275 ADD# 0,1,SNR
14 03393 040275 ADD# 0,1,SZR
15 03394 040275 ERROR
16 03395 040275 PP19A: LOOP
17 03396 040275 JSR 0,ENTLO
18 03397 040275 44
19 03398 040275 006231

INITIALIZE STACK
FAULT ADDRESS IS PP11A
ITERATE TEST ROUTINE.....
NEXT ADDRESS
TEST FOR LAST

```

ITERATE TEST ROUTINE.....  
 ADVANCE TO NEXT BUFFER  
 LOCATION.

ITERATE TEST ROUTINE.....  
 ADVANCE TO NEXT BUFFER  
 LOCATION.  
 CHECK FOR END.

ITERATE TEST ROUTINE.....  
 ADVANCE TO NEXT BUFFER  
 LOCATION.  
 CHECK FOR END.

```

10051 ECL31
01
02
03
04 03247 006230
05 03250 000040
06
07 03251 006221
08 03252 003263
09 03253 102000
10 03254 105000
11 03255 131000
12 03256 155000
13 03257 147110
14 03258 157210
15 03261 106410
16 03262 156414
17
18
19 03267 006231
20
21 03270 006230
22 03271 000040
23
24 03272 006221
25 03273 003303
26 03274 102000
27
28 03277 006234
29 03278 115400
30 03279 171400
31 03282 143110
32 03300 167210
33 03301 116415
34 03302 112014
35
36
37 03307 006231
38
39
40 03310 006230
41 03311 000040
42
43 03312 006221
44 03313 003321
45
46 03314 006234
47 03315 115400
48 03316 163110
49 03317 117210
50 03320 116014
51
52 03325 006231
53
54
10052 ECL21
01
02
03
04 03326 006230
05 03327 000040
06
07 03330 006221
08 03331 003341
09 03332 102400
10 03333 105000
11 03334 131000
12 03335 155020
13 03336 103710
14 03337 107215
15 03340 157014
16
17
18 03345 006231
19
20
21 03346 006230
22 03347 000040
23
24 03350 006221
25 03351 003361
26 03352 102040
27 03353 120000
28 03354 155000
29 03355 170000
30 03356 103710
31 03357 106615
32 03360 156414
33
34
35 03365 006231
36
37
38 03366 006230
39 03367 000040
40
41 03370 006221
42 03371 003400
43 03372 101040
44 03373 020220
45 03374 103710
46 03375 024040
47 03376 106615
48 03377 101003
49
50
51 03404 006231
52

```

```

PP12:
SETUP 40
JSR #ENTIN
40
STACK PP12A
JSR #ISTK
PP12A
PRGA
MOV 0,0
NOV 0,1
NOV 1,2
NOV 2,3
PSH 2,3
POP 2,3
SUB# 0,1,SNR
ADD# 2,3,SZR
ERROR
LOOP
JSR #ENTLO

ITERATE TEST ROUTINE.....

PP13:
SETUP 40
JSR #ENTIN
40
STACK PP13A
JSR #ISTK
PP13A
RAND
INC 0,3
INC 3,2
PSH 2,0
POP 3,1
SUB# 0,3,SNR
ADD# 0,2,SZR
ERROR
LOOP
JSR #ENTLO

ITERATE TEST ROUTINE.....

PP14:
SETUP 40
JSR #ENTIN
40
STACK PP14A
JSR #ISTK
PP14A
RAND
INC 0,3
EXCHANGE AC0 AND ACS
PSH 3,0
POP 0,3
ADD# 0,3,SZR
ERROR
LOOP
JSR #ENTLO

ITERATE TEST ROUTINE.....

```

```

PR0:
SETUP 40
JSR #ENTIN
40
STACK PR0A
JSR #ISTK
PR0A
SUB 0,0
MOV 0,1
MOV 0,2
MOVZ 2,3
PSHR #0,1,SNR
ADD# 0,1,SNR
ADD# 2,3,SZR
ERROR
LOOP
JSR #ENTLO

ITERATE TEST ROUTINE.....

PR1:
SETUP 40
JSR #ENTIN
40
STACK PR1A
JSR #ISTK
PR1A
ADCO 0,0
ADC 1,1
ADC 2,2
ADC 3,3
PSHR #0,1,SNR
SUB# 2,3,SZR
ERROR
LOOP
JSR #ENTLO

ITERATE TEST ROUTINE.....

PR2:
SETUP 40
JSR #ENTIN
40
STACK PR2A
JSR #ISTK
PR2A
NOV0 0,0
LDA #0,0
PSHR
LDA 1,SP
SUB# 0,1,SNR
MOV 0,0,SNR
ERROR
LOOP
JSR #ENTLO

ITERATE TEST ROUTINE.....

PR2A:
SETUP 40
JSR #ENTIN
40
STACK PR2A
JSR #ISTK
PR2A
NOV0 0,0
LDA #0,0
PSHR
LDA 1,SP
SUB# 0,1,SNR
MOV 0,0,SNR
ERROR
LOOP
JSR #ENTLO

ITERATE TEST ROUTINE.....

```

```

TEST #PSH/POP#
INITIALIZE TEST.....

INITIALIZE STACK
FAULT ADDRESS IS PP12A

PUSH 4 AC'S OF ALL ONES
POP THEM BACK.

ITERATE TEST ROUTINE.....

TEST #PSH/POP#
INITIALIZE TEST.....

INITIALIZE STACK
FAULT ADDRESS IS PP13A

JC(AC0)#RANDOM #

STACK GETS AC.2,3,0
POP TO AC 3,21

ITERATE TEST ROUTINE.....

TEST #PSM/POP#
INITIALIZE TEST.....

INITIALIZE STACK
FAULT ADDRESS IS PP14A

JC(AC0)#RANDOM #
EXCHANGE AC0 AND ACS
VIA PSH/POP
STACK GETS ACS,AC0
POP BACK IN REVERSE.

ITERATE TEST ROUTINE.....

```

```

PUSH PC #PSHR#
INITIALIZE TEST.....

INITIALIZE STACK
FAULT ADDRESS IS PR0A
PSHR SHOULD NOT CHANGE
ANY AC OR C(CARRY)

SETUP 40
JSR #ENTIN
40
STACK PR0A
JSR #ISTK
PR0A
SUB 0,0
MOV 0,1
MOV 0,2
MOVZ 2,3
PSHR #0,1,SNR
ADD# 0,1,SNR
ADD# 2,3,SZR
ERROR
LOOP
JSR #ENTLO

ITERATE TEST ROUTINE.....

TEST #PSHR#
INITIALIZE TEST.....

INITIALIZE STACK
FAULT ADDRESS IS PR1A
SET ALL AC'S AND
C(CARRY) TO (1).

SHOULD NOT CHANGE ACS
AC OR C(CARRY) OR
STACK FAULT.

ITERATE TEST ROUTINE.....

TEST #PSHR#
INITIALIZE TEST.....

INITIALIZE STACK
FAULT ADDRESS IS PR2A
TEST THE INCREMENT
OF STACK POINTER (SP)

JC(AC0)#CORRECT SP
JC(AC1)#CURRENT SP
JC(CARRY) SHOULD NOT CHANGE
STACK POINTER (SP)

ITERATE TEST ROUTINE.....

```

18053 ECL21

```

01
02
03 03405 006221      STACK PRSB
04 03406 003420      JSR #1STK
05 03407 028040      INITIALIZE STACK
06 03410 049240      JFAULT ADDRESS IS PRSB
07
08 03411 006230      PR3:
09 03412 009020      PR3B
10 03413 028240      LDA #1SP
11 03414 040940      STA #1TEM
12 03415 103710      SETUP #20
13 03416 024040      PR3A:
14 03417 106014      JSR #ENTN
15
16 03424 006231      LDA #1TEM
17 03425 019240      STA #1SP
18 03426 029240      PSHR
19 03427 024042      LDA #1SP
20 03430 106414      ADW #0.1,SZR
21 03431 006760      SUBW #0.1,SZR
22
23
24
25 03432 006230      PR4:
26 03433 006400      SETUP 400
27
28 03434 006221      STACK PR4A
29 03435 003442      JSR #1STK
30 03436 103710      PRAA
31 03437 004401      PSHR
32 03440 022040      LDA #1SP
33 03441 116414      SUBW #0.3,SZR
34
35 03446 006231      ERROR
36

```

18054 ECL21

```

01
02
03 03447 006221      STACK PR5B
04 03450 003465      JSR #1STK
05 03451 020040      INITIALIZE STACK
06 03452 040240      JFAULT ADDRESS IS PR5B
07
08 03453 006230      PR5:
09 03454 009020      PR5B
10 03455 020240      LDA #1TEM
11 03456 040940      STA #1SP
12 03457 103710      PSHR
13 03460 004401      JSR #1
14 03461 024040      LDA #1SP
15 03462 032040      SUBW #2.3,SMR
16 03463 156415      ADW #0.1,SZR
17 03464 106014      ERROR
18
19
20 03471 006231      LOOP
21 03472 019240      JSR #ENTLO
22 03473 020240      ISZ TEM
23 03474 024042      LDA #1TEM
24 03475 106414      SUBW #0.1,SZR
25 03476 006755      JMP PR5A
26

```

18055 ECL21

```

01
02
03 03477 006221      STACK PR6B
04 03480 003465      JSR #1STK
05 03481 020040      INITIALIZE STACK
06 03482 040240      JFAULT ADDRESS IS PR6B
07
08 03483 006230      PR6:
09 03484 009020      PR6B
10 03485 020240      LDA #1TEM
11 03486 040940      STA #1SP
12 03487 103710      PSHR
13 03490 004401      JSR #1
14 03491 024040      LDA #1SP
15 03492 032040      SUBW #2.3,SMR
16 03493 156415      ADW #0.1,SZR
17 03494 106014      ERROR
18
19
20 03495 006231      LOOP
21 03496 019240      JSR #ENTLO
22 03497 020240      ISZ TEM
23 03498 024042      LDA #1TEM
24 03499 106414      SUBW #0.1,SZR
25 03500 006755      JMP PR6A
26

```

```

10855 ECL21
01
02
03
04 03477 006230 PRG:
05 03500 000100 JSR #ENTIN
06
07 03501 006221 STACK PRGB
08 03502 003332 JSR #ISTK
09 03503 024842 PRGB
10 03504 030040 LDA 1,5L
11 03505 151400 LDA 2,5P
12 03506 140400 INC 2,2
13 03507 155400 SUB 2,1
14 03508 054840 STA 3,7EH
15 03509 054840 LDA 0,8PSHR
16 03510 020026* SUB 0,0
17 03511 041000 BAH
18 03512 102400 LDA 2,5L
19 03513 113710 STA 0,0,3
20 03514 030042 LDA 0,0,JMP 0,3
21 03515 041000 LDA 3,5P
22 03516 034840 LDA 3,5P
23 03517 034840 JSR 1,5
24 03518 034840 JSR 1,5
25 03519 021777 PRG:
26 03520 102814 ADCM 3,0,SZR
27 03521 000405 JMP PRGB
28 03522 175400 INC 3,3
29 03523 156414 SUBM 2,3,SZR
30 03524 156414 JMP PRGA
31 03525 000773 JMP PRCC
32
33
34 03536 006231 PRGB:
35
36
37
38
39
40
41
42
43
44 03613 006230 PRG:
45 03614 000200 JSR #ENTLO
46
47 03615 006221 STACK PJ2A
48 03616 003052 JSR #ISTK
49 03617 020024* PJ2A
50 03618 103110 LDA 0,#PJ2R
51 03619 117710 PSH 0,0
52
53
54 03626 006231 PJ2A:
55
56
57
58
59
60
61
62
63
64 03537 006230 PJ01:
65 03538 000040 JSR #ENTIN
66
67 03541 006221 STACK PJ0A
68 03542 003352 JSR #ISTK
69 03543 102420 PRGB
70 03544 105000 LDA 1,5L
71 03545 131000 LDA 2,5P
72 03546 155000 INC 2,2
73 03547 103710 SUB 2,1
74 03548 117710 STA 3,7EH
75 03549 117710 LDA 0,8PSHR
76 03550 101003 PJ0A:
77 03551 107015 ERROR 0,1,SNR
78 03552 107015 ADDM 2,3,SZR
79 03553 157014 SUB 0,0
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```



10057 ECL21

```

01
02
03
04 03627 006230
05 03630 000820
06
07 03631 008221
08 03632 003644
09 03633 101020
10 03634 020020-
11 03635 103110
12 03636 117710
13
14 03643 101002 PJ3A1
15 03644 101002 PJ3B1
16
17 03650 006231
18
19
20 03651 006221 PJ301
21 03652 003665 PJ308
22 03653 020040 PJ308A1
23 03654 048240 PJ308A1
24
25 03655 006230 PJ308
26 03656 000820 PJ308
27 03657 020240 PJ308
28
29 03661 103710 PJ308
30 03662 117710 PJ308
31 03663 024040 PJ308
32 03664 106414 PJ308
33
34
35 03671 000231 PJ308
36 03672 010240 PJ308
37 03673 020240 PJ308
38 03674 024042 PJ308
39 03675 106414 PJ308
40 03676 000757 PJ308
41

```

10058 ECL21

```

01
02
03
04 03677 006221 PJ41
05 03700 003742 PJ41
06 03701 030000 PJ41
07 03702 024002 PJ41
08 03703 101400 PJ41
09 03704 140400 PJ41
10 03705 154000 PJ41
11 03706 020020- PJ41
12 03707 041000 PJ41
13 03710 102400 PJ41
14 03711 113710 PJ41
15 03712 020040 PJ41
16 03713 040240 PJ41
17
18 03714 006230 PJ4A1
19 03715 000000 PJ4A1
20 03716 030240 PJ4A1
21 03717 050000 PJ4A1
22 03720 020021 PJ4A1
23 03721 103110 PJ4A1
24 03722 004402 PJ4A1
25 03723 000400 PJ4A1
26 03724 103710 PJ4A1
27 03725 001001 PJ4A1
28 03726 020022 PJ4A1
29 03727 042040 PJ4A1
30 03730 000410 PJ4A1
31 03731 030240 PJ4A1
32 03732 025002 PJ4A1
33 03733 020040 PJ4A1
34
35 03740 000401 PJ401
36 03741 101001 PJ401
37
38
39 03746 006231 PJ4E1
40 03747 010240 PJ4E1
41 03750 020240 PJ4E1
42 03751 024042 PJ4E1
43 03752 106014 PJ4E1
44 03753 000741 PJ4E1
45

```

```

)TEST "POPJ"
)INITIALIZE TEST.....
)INITIALIZE STACK
)FAULT ADDRESS IS PJ4E
)DO A POPJ FROM EACH
)STACK/BUFFER LOCATION
)BACK TO THE PROGRAM.

)FILL THE BUFFER WITH
)JSR #S) INSTRUCTIONS
)FOR A ERROR RETURN.

)INITIALIZE TEST.....
)ON EACH ITERATION, 2 ENTRIES
)INTO STACK ARE
)PSHR
)PC RETURN
)POP BACK TO WRONG SPOT.
)STORE PC FOR RETURN.
)RETURN HERE
)RESTORE BUFFER.
)IC(AC2)+1=ADDRESS OF POPJ
)IC(AC1)=THE PC IT SHOULD USE
)IC(AC0)=CURRENT STACK POINTER
)STACK FAULT
)ITERATE TEST ROUTINE.....
)NEXT BUFFER LOCATION
)TEST FOR LAST.

```

```

)STACK PJ4E
)JSR #ISTK
)LDA 2,SP
)LDA 1,SL
)INC 2,2
)SUB 2,1
)INC 2,3
)LDA 0,#JSR 0,3
)STA 0,#J2
)SUB 0,0
)BAM
)LDA 0,SP
)STA 0,TEM
)SETUP 5
)JSR #ENTLO
)S
)LDA 2,TEM
)STA 2,SP
)LDA 0,#POPJ
)PSH 0,0
)JSR #2
)JMP PJ4C
)PSHR 1,2
)JMP 1,2
)LDA 0,#JSR 0,3
)STA 0,SP
)JMP PJ4D
)LDA 2,TEM
)LDA 1,2,2
)LDA 0,#SP
)ERROR
)JMP #1
)MOV 0,#R,SKP
)ERROR
)LOOP
)JSR #ENTLO
)ISZ TEM
)LDA 0,TEM
)LDA 1,SL
)ADC# 0,1,5ZR
)JMP PJ4A

```

```

)TEST "POPJ"
)INITIALIZE TEST.....
)INITIALIZE STACK
)FAULT ADDRESS IS PJ3B
)TEST C(CARRY)=0) WHEN
)BI 0 0 MEMORY=(1).
)RETURN TO PJ3A
)POPJ
)FAIL TO PC TRANSFER-
)C(CARRY) WAS SET OR
)STACK OVERFLOW.
)ITERATE TEST ROUTINE.....

)TEST "POPJ"
)INITIALIZE TEST.....
)INITIALIZE STACK
)FAULT ADDRESS IS PJ308
)CHECK STACK POINTER
)INITIALIZE TEST.....
)DECREMENT VIA POPJ.
)C(SP)+1
)C(SP)-1
)SHOULD BE SAME AS ORIGINAL.
)C(AC0)=ORIGINAL/CORRECT
)C(AC1)=FINAL/ERROR
)ITERATE TEST ROUTINE.....
)ADVANCE STACK POINTER
)TEST FOR END OF BUFFER.

```

```

)SETUP 20
)JSR #ENTLO
)STACK PJ308
)JSR #ISTK
)MOVZ 0,0
)LDA 0,#PJ3A
)PSH 0,0 )PUISH A
)POPJ
)ERROR
)MOV 0,0,5ZC
)ERROR
)LOOP
)JSR #ENTLO

)STACK PJ308
)JSR #ISTK
)LDA 0,SP
)STA 0,TEM
)SETUP 20
)JSR #ENTLO
)LDA 0,TEM
)STA 0,SP
)PSHR
)POPJ
)LDA 1,SP
)SUB# 0,1,5ZR
)ERROR
)LOOP
)JSR #ENTLO
)ISZ TEM
)LDA 0,TEM
)LDA 1,SL
)SUB# 0,1,5ZR
)JMP PJ30A

```

```

10059 ECL21
01
02
03
04 03754 006221 PJ51
05 03755 004022 JSC
06 03756 030040 JSR 0,1STK
07 03757 024042 LDA 2,SP
08 03758 151400 LDA 1,SL
09 03761 146000 INC 2,2
10 03762 155400 INC 2,3
11 03763 020022- LDA 0,1,JSR 0,3
12 03764 041000 STA 0,0,2
13 03765 102400 SUB 0,0
14 03766 113710 BAH
15 03767 020040 PJ5A: LDA 0,SP
16 03770 101400 INC 0,0
17 03771 040404 STA 0,PJ55
18 03772 006230 SETUP 10
19 03773 000010 JSR 0,ENTIN
20 03774 004412 10
21 03775 000000 PJ55: 0
22 03776 000010 .BLK 10
23 03777 000010 STA 3,SP
24 04006 054040 PJ56: LDA 0,PJ55
25 04007 020766 INC 0,1
26 04010 105400 JSR 0,2
27 04011 004402 MOV 0,0,SKP
28 04012 103001 POPJ
29 04013 117710 SUR# 1,3,5ZR
30 04014 130414 ERROR
31 04021 101001 MOV 0,0,SKP
32 04022 101001 ERROR
33
34
35 04026 006231 JSR 0,ENTLO
36 04027 010740 ISZ PJ55
37 04030 024042 LDA 1,SL
38 04031 020744 LDA 0,PJ55
39 04032 100414 SUR# 0,1,5ZR
40 04033 000737 JMP PJ50

TEST "POPJ"
INITIALIZE STACK
IF AULT ADDRESS IS PJ5C
IFILL A BUFFER WITH (JSR 0,3)
FINSTRUCTIONS. DO A
FPOPJ FROM THE PROGRAM
FTO THE BUFFER....
FBUFFER EXECUTES A "JSR"
FBACK TO THE PROGRAM....
FTHE PC STORED IN C(ACS)
FISY THIS JSR IS USED
FTO CHECK THE ADDRESS
FPOPPED TO.

INITIALIZE TEST....

NEXT TO BUFFER....
C(ACS)=CORRECT ADDRESS+1
C(ACS)=ACTUAL ADDRESS*1

STACK FAULT....

ITERATE TEST ROUTINE....
NEXT ADDRESS
TEST FOR END
TOP BUFFER.

STACK PJ5C
JSR 0,1STK
LDA 2,SP
LDA 1,SL
INC 2,2
INC 2,3
LDA 0,1,JSR 0,3
STA 0,0,2
SUB 0,0
BAH
LDA 0,SP
INC 0,0
STA 0,PJ55
SETUP 10
JSR 0,ENTIN
10
PJ55: 0
.BLK 10
STA 3,SP
LDA 0,PJ55
INC 0,1
JSR 0,2
MOV 0,0,SKP
POPJ
SUR# 1,3,5ZR
ERROR
MOV 0,0,SKP
ERROR
LOOP
JSR 0,ENTLO
ISZ PJ55
LDA 1,SL
LDA 0,PJ55
SUR# 0,1,5ZR
JMP PJ50

10060 ECL21
01
02
03
04 04034 006230 PJ6:
05 04035 001000 PJ6A:
06
07 04036 006221
08 04037 004072
09
10 04040 006234 JSR 0,ENTRA
11 04041 110300 MOVS 0,3
12 04042 024042 LDA 1,SL
13 04043 030040 LDA 2,SP
14 04044 146000 ADC 2,1
15 04045 124022 SUBZ 1,0,5ZC
16 04046 000777 JMP 0,1
17 04047 123000 SUBZ 1,3,5ZC
18 04050 130422 JMP 0,1
19 04051 000777
20 04052 137000 SUR# 0,3,5NR
21 04053 116415 JMP PJ6A
22 04054 000762 LDA 1,0P
23 04055 024040 ADD 1,0P
24 04056 123000 ADD 1,0P
25 04057 137000 LDA 1,0P
26 04060 024020- LDA 2,0P
27 04061 030020- STA 1,0P,3
28 04062 040000 JSR 0,3
29 04063 051401 STA 2,1,3
30 04064 040040 JSR 0,3
31 04065 005400 JGO TO DO THE "PSHJ"
32 04066 030040 PC(AC2)=NEW STACK POINTER
33 04067 025000 PC(AC1)=PC STORED VIA PSHR
34 04070 130415 SUR# 1,3,5ZR
35 04071 110014 ADC# 0,2,5ZR
36
37 04076 006231 ERROR
38 04076 006231 LOOP
39
40
ITERATE TEST ROUTINE....

```

10061 ECL21

```

01
02
03
04
05
06
07 04077 00623R
08 04100 000010
09
10 04101 006221
11 04102 004112
12 04103 102420
13 04104 105000
14 04105 131000
15 04106 163710
16 04107 000000
17 04108 107915
18 04109 151074
19
20
21 04110 006231
22
23
24 04117 00623R
25 04120 000010
26
27 04121 006221
28 04122 004117
29 04123 102020
30 04124 105000
31 04125 131000
32 04126 163710
33
34 04130 106415
35 04131 150214
36
37
38 04136 006231
39
40
41 04137 00623R
42 04140 000100
43
44 04141 006221
45 04142 004151
46 04143 176840
47 04144 163710
48
49 04146 020040
50 04147 116415
51 04150 101003
52
53
54 04155 006231
55

```

```

*** TEST SAVE ***
SETUP 10
JSR #ENTIN
10
STACK SV0A
JSR #1STK
SV0A 0,0
SUBZ 0,0
MOV 0,1
MOV 1,2
SAVE 0
ADD# 0,1,SNR
INCR# 2,2,SZR
ERROR
LOOP
JSR #ENTLO

```

```

ITEST "SAVE"
IINITIALIZE TEST....

IINITIALIZE STACK
IFault ADDRESS IS SV0A
IINITIALIZE STACK
IFault ADDRESS IS SV0A
ITEST TO INSURE THE STACK
IPointer (SP) IS INCREMENTED
IBY 5 ITEMS. ALSO THAT

IC(AC0) IS EQUAL NEW C(SP).
IC(AC1)=ORIGINAL C(SP)
IC(AC1)=C(SP) AFTER "SAVE"
IC(AC2)=CORRECT C(SP)
IC(AC3)=C(SP) VIA "SAVE"
ITERATE TEST ROUTINE....

```

```

ITERATE TEST ROUTINE....
ITEST "SAVE"
IINITIALIZE TEST....

IINITIALIZE STACK
IFault ADDRESS IS SV1
ISET C(AC0-2) TO 17777
ISET C(CARRY)=0
ITHE SAVE INSTRUCTION
ISHOULD NOT CHANGE C(AC0-2)
FOR C(CARRY)

```

```

ITERATE TEST ROUTINE....
FOR C(AC0-2)
SUB# 0,1,SNR
CDRR# 2,2,SZR
ERROR
LOOP
JSR #ENTLO

```

```

ITERATE TEST ROUTINE....
ITEST "SAVE"
IINITIALIZE TEST....

IINITIALIZE STACK
IFault ADDRESS IS SV2A
ADCO 3,3
SAVE 0
LDA 0,SP
SUB# 0,3,SNR
MOV 0,0,SNR
ERROR
LOOP
JSR #ENTLO

```

```

ITERATE TEST ROUTINE....
ITEST "SAVE"
IINITIALIZE TEST....

IINITIALIZE STACK
IFault ADDRESS IS SV2A
ADCO 3,3
SAVE 0
LDA 0,SP
SUB# 0,3,SNR
MOV 0,0,SNR
ERROR
LOOP
JSR #ENTLO

```

10062 ECL21

```

01
02
03
04 04156 00623R
05 04157 000100
06
07 04160 006221
08 04161 004174
09 04162 006221
10 04163 004174
11 04164 020040
12 04165 030020-
13 04166 163710
14
15 04170 024040
16 04171 113000
17 04172 156415
18 04173 146414
19
20
21 04200 006231
22
23
24 04201 006221
25 04202 004224
26 04203 020040
27 04204 040040
28 04205 020040
29 04206 024020-
30 04207 122400
31 04210 000273
32
33 04211 00623R
34 04212 000320
35 04213 020240
36 04214 040040
37 04215 163710
38
39 04217 030020-
40 04220 024040
41 04221 113000
42 04222 156415
43 04223 146414
44
45
46 04230 006231
47 04231 010020
48 04232 020040
49 04233 024273
50 04234 166414
51 04235 000754

```

```

SETUP 10R
JSR #ENTIN
100
STACK SV3A
JSR #1STK
SV3A SV3A
JSR #1STK
SV3A SV3A
LDA 0,SP
LDA 2,5
SAVE 0
LDA 1,SP
ADD 0,2
SUB# 2,3,SNR
SUB# 2,1,SZR
ERROR
LOOP
JSR #ENTLO

```

```

ITEST "SAVE"
IINITIALIZE TEST....

IINITIALIZE STACK
IFault ADDRESS IS SV3A
IINITIALIZE STACK
IFault ADDRESS IS SV3A
ITEST TO INSURE THE STACK
IPointer (SP) IS INCREMENTED
IBY 5 ITEMS. ALSO THAT

IC(AC0) IS EQUAL NEW C(SP).
IC(AC0)=ORIGINAL C(SP)
IC(AC1)=C(SP) AFTER "SAVE"
IC(AC2)=CORRECT C(SP)
IC(AC3)=C(SP) VIA "SAVE"
ITERATE TEST ROUTINE....

```

```

ITERATE TEST ROUTINE....
ITEST "SAVE"
IINITIALIZE TEST....

IINITIALIZE STACK
IFault ADDRESS IS SV4B
LDA 0,SP
LDA 0,TEM
LDA 0,SL
LDA 1,5
SUB 1,0
STA 0,TEM1
SETUP 20
JSR #ENTIN
20
LDA 0,TEM
LDA 0,SP
SAVE 0
LDA 2,5
LDA 1,SP
ADD 0,2
SUB# 2,3,SNR
SUB# 2,1,SZR
ERROR
LOOP
JSR #ENTLO

```

```

ITERATE TEST ROUTINE....
ITEST "SAVE"
IINITIALIZE TEST....

IINITIALIZE STACK
IFault ADDRESS IS SV4B
LDA 0,SP
LDA 0,TEM
LDA 0,SL
LDA 1,5
SUB 1,0
STA 0,TEM1
SETUP 20
JSR #ENTIN
20
LDA 0,TEM
LDA 0,SP
SAVE 0
LDA 2,5
LDA 1,SP
ADD 0,2
SUB# 2,3,SNR
SUB# 2,1,SZR
ERROR
LOOP
JSR #ENTLO

```

```

ITERATE TEST ROUTINE....
ITEST "SAVE"
IINITIALIZE TEST....

IINITIALIZE STACK
IFault ADDRESS IS SV4B
LDA 0,SP
LDA 0,TEM
LDA 0,SL
LDA 1,5
SUB 1,0
STA 0,TEM1
SETUP 20
JSR #ENTIN
20
LDA 0,TEM
LDA 0,SP
SAVE 0
LDA 2,5
LDA 1,SP
ADD 0,2
SUB# 2,3,SNR
SUB# 2,1,SZR
ERROR
LOOP
JSR #ENTLO

```

```

ITERATE TEST ROUTINE....
ITEST "SAVE"
IINITIALIZE TEST....

IINITIALIZE STACK
IFault ADDRESS IS SV4B
LDA 0,SP
LDA 0,TEM
LDA 0,SL
LDA 1,5
SUB 1,0
STA 0,TEM1
SETUP 20
JSR #ENTIN
20
LDA 0,TEM
LDA 0,SP
SAVE 0
LDA 2,5
LDA 1,SP
ADD 0,2
SUB# 2,3,SNR
SUB# 2,1,SZR
ERROR
LOOP
JSR #ENTLO

```

10063 ECL21

```

01
02
03
04 04236 006231
05 04237 000640
06
07 04240 006221
08 04241 004250
09 04242 020040
10 04243 024020
11 04244 123000
12 04245 163710
13 000000
14 04247 024041
15 04250 030040
16 04251 106415
17 04252 106414
18
19
20 04257 006231
21
22 04260 006221
23 04261 004302
24 04262 020040
25 04263 040240
26 04264 020040
27 04265 024020
28 04266 122400
29 04267 040273
30
31 04270 006230
32 04271 000010
33 04272 020240
34 04273 040040
35 04274 163710
36 04275 000000
37
38 04276 024041
39 04277 030040
40 04278 136415
41 04281 132414
42
43
44 04306 006231
45 04307 010240
46 04310 020240
47 04311 024273
48 04312 106414
49 04313 000752

```

10064 ECL21

```

01
02
03 04314 006221
04 04315 004341
05 04316 020040
06 04317 040240
07 04320 020040
08 04321 024020
09 04322 122400
10 04323 040273
11
12 04324 006230
13 04325 000010
14
15 04326 006234
16 04327 113300
17 04330 040041
18 04331 020240
19 04332 040040
20 04333 163710
21
22 04335 024041
23 04336 030040
24 04337 136415
25 04340 132414
26
27 04345 006231
28 04346 010240
29 04347 020240
30 04350 024273
31 04351 106414
32 04352 000752
33
34
35

```

```

STACK SV7B
JSR 01STK
SV7B
LDA 0,SP
STA 0,TEM
LDA 0,SL
LDA 1,=5
SUB 1,0
STA 0,TEM1
SETUP 10
JSR 0ENTIN
RAND
JSR 0ENTRA
MOVS 0,D
STA 0,FP
LDA 0,TEM
SAVE 0
LDA 1,FP
LDA 2,SP
SUB# 1,3,SNR
SUB# 1,2,SZR
ERRR
LOOP
JSR 0ENTLO
ISZ TEM
LDA 0,TEM
LDA 1,TEM1
SUB# 0,1,SZR
JMP SV7A

```

```

TEST "SAVE"
INITIALIZE TEST....
INITIALIZE STACK
IF AULT ADDRESS IS SV7B
ISANE AS PREVIOUS TEST
PECT BOTH C(AC3) AND
C(FP) ARE SET RANDOM
BEFORE THE SAME INSTRUCTION.
INITIALIZE TEST....
C(AC0)=RANDOM #
C(AC0)=ORIGINAL C(ASP)
C(AC1)=FRAME POINTER
C(AC2)=STACK POINTER
C(AC3)=STACK POINTER VIA "SAVE"
ITERATE TEST ROUTINE....

```

```

TEST "SAVE"
INITIALIZE TEST....
INITIALIZE STACK
IF AULT ADDRESS IS SV5A
TEST TO INSURE THE
FRAME POINTER IS UPDATED
AND MADE EQUAL TO STACK
POINTER
C(AC0)=CORRECT C(FP),C(ASP),C(AC3)
C(AC1)=FRAME POINTER
C(AC2)=FINAL STACK POINTER
C(AC3)=STACK POINTER VIA "SAVE"
ITERATE TEST ROUTINE....
TEST "SAVE"
INITIALIZE STACK
IF AULT ADDRESS IS SV6B
SAVE MACHINE STATE ON
THE STACK. ADVANCE THE
C(ASP) UNTL STACK FAULT
OULD OCCUR. TEST FOR
PROPER VALUE IN FRAME
POINTER.
INITIALIZE TEST....
C(AC0)=ORIGINAL C(ASP)
C(AC1)=FRAME POINTER (FP)
C(AC2)=STACK POINTER (SP)
C(AC3)=STACK POINTER VIA "SAVE"
ITERATE TEST ROUTINE....
INCREMENT TO NEXT
STACK ENTRY.
TEST FOR END OF STACK.

```

```

SETUP 40
JSR 0ENTIN
40
STACK SV5A
JSR 01STK
SV5A
LDA 0,SP
LDA 0,SL
LDA 1,=5
ADD 1,0
SAVE 0
LDA 1,FP
LDA 2,SP
SUB# 0,1,SNR
SUB# 3,1,SZR
ERRR
LOOP
JSR 0ENTLO
STACK SV6B
JSR 01STK
SV6B
LDA 0,SP
STA 0,TEM
LDA 0,SL
LDA 1,=5
SUB 1,0
STA 0,TEM1
SETUP 10
JSR 0ENTIN
LDA 0,TEM
STA 0,SP
SAVE 0
LDA 1,FP
LDA 2,SP
SUB# 1,3,SNR
SUB# 1,2,SZR
ERRR
LOOP
JSR 0ENTLO
ISZ TEM
LDA 0,TEM
LDA 1,TEM1
SUB# 0,1,SZR
JMP SV6A

```

10065 ECL21

01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44

04 04353 006231 SV8: STACK SV8B  
05 04354 004412 JSR #1STK  
06 04355 020840 SV8B #1STK  
07 04356 040840 LDA #,SP  
08 04357 020842 STA #,TEM  
09 04358 024817 = LDA 1,-24,  
10 04359 122400 SUB 1,#  
11 04362 040273 STA #,TEH1  
12 04363 006230 JSR #ENTIN  
13 04364 000810 LDA #,SP  
14 04365 006234 RAND  
15 04366 115060 JSR #ENTRA  
16 04367 040841 MOV #0,#J  
17 04368 020840 STA #,FP  
18 04369 020840 LDA #,TEM  
19 04370 040840 STA #,SP  
20 04371 040840 SAVE 1  
21 04372 103710 SAVE 1  
22 04373 000001 SAVE 1  
23 04374 103710 SAVE 1  
24 04375 103710 SAVE 1  
25 04376 000001 SAVE 1  
26 04377 000001 SAVE 1  
27 04400 103710 LDA 1,-24,  
28 04402 024817 = ADD #,1  
29 04403 107000 LDA 2,-FP  
30 04404 030041 LDA #,SP  
31 04405 020040 SUB #0,1,#NR  
32 04406 020040 MOV #2,1,#NR  
33 04407 146014 MOV #0,#,SVR  
34 04410 101001 MOV #3,1,#SVR  
35 04411 100014 ERROR  
36 04412 000740 LOOP  
37 04413 006231 JSR #ENTLO  
38 04414 010240 ISZ TEM  
39 04415 010240 LDA #,TEM  
40 04416 020240 LDA 1,TEM1  
41 04417 024273 SUB #0,1,#SZR  
42 04418 106414 JMP SV8A  
43 04419 000740 SV8A: SV8B: SV8C:  
44 04420 000740

10066 ECL21

01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52

04 04424 006230 SV9: SETUP 100  
05 04425 000100 JSR #ENTIN  
06 04426 006221 STACK SV9A  
07 04427 004440 JSR #1STK  
08 04428 004440 SV9A #1STK  
09 04429 006234 RAND  
10 04430 006234 JSR #ENTRA  
11 04431 104400 NEG #,1  
12 04432 103710 SAVE #  
13 04434 025775 LDA 1,-3,J  
14 04435 031774 LDA 2,-4,J  
15 04436 142415 SUB #0,#,SNR  
16 04437 123014 ADD #1,#,SZR  
17 04438 123014 SV9A: ERROR  
18 04439 000000 LOOP  
19 04440 006231 JSR #ENTLO  
20 04441 006231 STACK SV100  
21 04442 006221 JSR #1STK  
22 04443 004471 SV100: LDA #,SP  
23 04444 020040 STA #,TEM  
24 04445 040240 LDA #,SL  
25 04446 020042 LDA 1,-5  
26 04447 020042 STA #,TEM1  
27 04448 122400 SUB 1,#  
28 04449 040273 STA #,TEH1  
29 04450 040273 SV100A: SETUP #  
30 04451 006230 JSR #ENTIN  
31 04452 006230 STACK SV108  
32 04453 006221 JSR #1STK  
33 04454 004471 SV108: LDA #,SP  
34 04455 020040 STA #,TEM  
35 04456 040040 LDA #,SL  
36 04457 020042 LDA 1,-5  
37 04458 020042 STA #,TEM1  
38 04459 104400 NEG #,1  
39 04460 103710 SAVE #  
40 04461 006234 LDA 1,-3,J  
41 04462 025775 LDA 2,-4,J  
42 04463 031774 SUB #0,#,SNR  
43 04464 142415 ADD #1,#,SZR  
44 04465 123014 SV108A: ERROR  
45 04466 000000 LOOP  
46 04467 006231 JSR #ENTLO  
47 04468 010240 ISZ TEM  
48 04469 010240 LDA #,TEM  
49 04470 020240 LDA 1,TEM1  
50 04471 024273 SUB #0,1,#SZR  
51 04472 106414 JMP SV10A  
52 04473 000740

TEST "SAVE"  
INITIALIZE TEST.....  
INITIALIZE STACK  
FAULT ADDRESS IS SV9A  
TEST STORE OF AC0-1  
C(AC0)=RANDOM #  
SET C(AC1) TO NEGATE OF C(AC0).  
SAVE THE MACHINE STATE AND  
LOAD AC0-1 FROM STACK.  
C(AC0)=ORIGINAL C(AC0)/CORRECT  
C(AC1)=STACK AC1  
C(AC2)=C(AC0) FROM STACK  
C(AC3)=STACK POINTER  
ITERATE TEST ROUTINE.....  
TEST "SAVE"  
INITIALIZE STACK  
FAULT ADDRESS IS SV108  
TEST THE STORE OF  
AC0-1 ON THE STACK  
INITIALIZE TEST.....  
C(AC0)=RANDOM #  
C(AC1)=C(AC0) NEGATED.....  
SAVE MACHINE STATE AND  
LOAD BACK.  
C(AC0)=ORIGINAL C(AC0)/CORRECT  
C(AC1)=ORIGINAL C(AC1)/FROM STACK  
C(AC2)=ORIGINAL C(AC2)/FROM STACK  
C(AC3)=STACK POINTER  
ITERATE TEST ROUTINE.....

10066 ECL21

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

SV11:

SV12:

SV12A:

SV12B:

SETUP 400

JSR #ENTN

400

STACK SV12

JSR #ISTK

SV12

RAND

JSR #ENTRA

MOV5 0,2

STA 0,FP

SAVE 0

LDA 1,-1,3

LDA 3,-2,3

SUB# 0,1,SZR

SUB# 2,3,SZR

ERROR

LOOP

JSR #ENTLO

STACK SV12C

JSR #ISTK

SV12C

LDA 0,SP

LDA 0,SL

LDA 1,5

SUB 1,0

STA 0,TEM1

SETUP 20

JSR #ENTN

20

LDA 0,TEM

STA 0,SP

RAND

JSR #ENTRA

MOV5 0,2

STA 0,FP

SAVE 0

LDA 1,-1,3

LDA 2,-2,3

SUB# 0,1,SZR

JMP SV12C

MOV5 0,0

SUB# 0,2,SZR

ERROR

MOV 0,0,SKP

ERROR

LOOP

JSR #ENTLO

ISZ TEM

LDA 0,TEM

LDA 1,TEM1

SUB# 0,1,SZR

JMP SV12B

10066 ECL21

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

SV13:

SV14:

SV14A:

SETUP 40

JSR #ENTN

40

STACK SV13A

JSR #ISTK

SV13A

SUB0 3,3

SAVE 0

LDA 0,0,3

MOV 0,0,SZR

ERROR

LOOP

JSR #ENTLO

SETUP 40

JSR #ENTN

40

STACK SV14A

JSR #ISTK

SV14A

SUB2 3,3

SAVE 0

LDA 0,0,3

SUB2R 1,1

SUB# 0,1,SZR

ERROR

LOOP

JSR #ENTLO

SETUP 40

JSR #ENTN

40

STACK SV15A

JSR #ISTK

SV15A

ADCR 1,1

ADCR 3,3

SAVE 0

LDA 0,0,3

SUB# 0,1,SZR

ERROR

LOOP

JSR #ENTLO

!TEST "SAVE"

!INITIALIZE TEST.....

!INITIALIZE STACK

!FAULT ADDRESS IS SV13A

!CLEAR C(AC3)\*C(CARRY)

!GET ORIG AC3\*ARRY

!FROM THE STACK.

!SHOULD BOTH BE 0.

!ITERATE TEST ROUTINE.....

!TEST "SAVE"

!INITIALIZE TEST.....

!INITIALIZE STACK

!FAULT ADDRESS IS SV14A

!CLEAR C(AC3). SET C(CARRY)

!C(AC0)=ORIG AC3\*ARRY FROM

!STACK. C(AC1)=CORRECT

!C(3)=CURRENT STACK POINTER.

!ITERATE TEST ROUTINE.....

!TEST "SAVE"

!INITIALIZE TEST.....

!INITIALIZE STACK

!FAULT ADDRESS IS SV15A

!SET C(AC3)=1. C(CARRY)=0

!C(AC0)=ORIGINAL AC3 FROM STACK

!C(AC1)=CORRECT

!C(AC3)=CURRENT STACK POINTER

!ITERATE TEST ROUTINE.....



```

10071 ECL21
01 05042 024273
02 05043 106414
03 05044 006744
04
05
06 05045 00623P
07 04750 00623P
08 04751 008100
09
10 04752 006221
11 04753 004773
12
13 04754 006234
14 04755 105400
15 04756 131400
16 04757 004402
17 04758 006404
18 04759 163710
19 04760 006000
20 04763 127710
21 04764 106015
22 04765 132914
23
24 04772 101001
25
26
27 04777 006231
28
29
30 05060 006221
31 05061 005033
32 05062 020040
33 05063 049240
34 05064 020042
35 05065 024020
36 05066 122400
37 05067 040273
38
39 05010 006230
40 05011 008020
41 05012 026240
42 05013 049040
43
44 05014 006234
45 05015 105400
46 05016 131400
47 05017 004402
48 05020 006404
49 05021 163710
50 05022 000000
51 05023 127710
52 05024 106015
53 05025 132914
54
55 05032 101001
56
57
58 05037 006231
59 05040 010240
60 05041 020240

RT0:
RT1:
RT1A:
RT1B:

*** TEST RTN ***

SETUP 100
JSR #ENTIN
100
STACK RT0A
JSR #1STK
RT0A
RAND
JSR #ENTRA
INC 0,1
INC 1,2
JSR .*2
JMP .*4
SAVE 0

RTN
ADC# 0,1,SNR
ADC# 1,2,SZR
ERR0R
MOV 0,0,SKP
ERR0R
LOOP
JSR #ENTLD

STACK RT10
JSR #1STK
RT10
LDA 0,SP
STA 0,TEM
LDA 0,SL
SUB 1,0
STA 0,TEM1
SETUP 20
JSR #ENTIN
20
LDA 0,TEM
STA 0,SP
RAND
JSR #ENTRA
INC 0,1
INC 1,2
JSR .*2
JMP .*4
SAVE 0

RTN
ADC# 0,1,SNR
ADC# 1,2,SZR
ERR0R
MOV 0,0,SKP
ERR0R
LOOP
JSR #ENTLD

PC(AC0)#RANDOM #
ISET C(AC1) I GREATER THAN
PC(AC0), C(AC2) I GREATER THAN
PC(AC1). SET PROGRAM COUNTER
IN C(AC3) FOR A RETURN.
I SAVE MACHINE....
PRESTORE...
ICHECK FOR ORIGINAL
IASSENDING ORDER.

I STAC FAULT
LOOP
ITERATE TEST ROUTINE....
I INCREMENT TO NEXT
I STACK ENTRY.

I STACK FAULT
ITERATE TEST ROUTINE....

I INITIALIZE STACK
IF AULT ADDRESS IS RT10
I SAVE MACH STATE.
PRESTORE AND CHECK
I AC0-2. ADVANCE TO
NEXT STACK LOCATION,
I UNTIL OVERFLOW WOULD OCCUR

I INITIALIZE TEST....

PC(AC0)#RANDOM #
ISET C(AC1) I GREATER THAN
PC(AC0), C(AC2) I GREATER THAN
PC(AC1). SET PROGRAM COUNTER
IN C(AC3) FOR A RETURN.
I SAVE MACHINE....
PRESTORE...
ICHECK FOR ORIGINAL
IASSENDING ORDER.

I STAC FAULT
LOOP
ITERATE TEST ROUTINE....
I INCREMENT TO NEXT
I STACK ENTRY.

I STACK FAULT
ITERATE TEST ROUTINE....

I SETUP MEMORY
IPRESTORE WORLD.
ICHECK C(AC0-2) FOR THE
IASSENDING ORDER.

I STACK FAULT
ITERATE TEST ROUTINE....

I INITIALIZE TEST....
I STACK RT2A
I INITIALIZE STACK
IF AULT ADDRESS IS RT2A
I INITIALIZE STACK
IF AULT ADDRESS IS RT2A

I THE "RTN" INSTRUCTION DID
INOT CAUSE A PC TRANSFER.
ITERATE TEST ROUTINE....

RTN
ERR0R
LOOP
JSR #ENTLD
19

RT2:
RT2A:

0072 ECL21
01 05042 024273
02 05043 106414
03 05044 006744
04
05
06 05045 00623P
07 05046 008100
08
09 05047 006221
10 05050 005056
11 05051 004402
12 05052 006405
13 05053 163710
14 05054 006000
15 05055 127710
16
17 05062 006231
18
19

```

TEST FOR END OF STACK.

LDA 1,TEM1  
SUB# 0,1,SNR  
JMP RT1A

TEST "RTN"  
INITIALIZE TEST....

SETUP 100  
JSR #ENTIN  
100  
STACK RT2A  
JSR #1STK  
RT2A  
JMP .\*2  
JMP .\*5  
SAVE 0

RTN  
ERR0R  
LOOP  
JSR #ENTLD  
19

PC(AC0)#RANDOM #  
ISET C(AC1) IS I GREATER THEN C(AC0)  
PC(AC2) IS I GREATER, THAN C(AC1)  
IFUT AN ADDRESS, IN C(AC3)  
I SETUP MEMORY  
IPRESTORE WORLD.  
ICHECK C(AC0-2) FOR THE  
IASSENDING ORDER.

RTN  
ERR0R  
LOOP  
JSR #ENTLD  
19

PC(AC0)#RANDOM #  
ISET C(AC1) I GREATER THAN  
PC(AC0), C(AC2) I GREATER THAN  
PC(AC1). SET PROGRAM COUNTER  
IN C(AC3) FOR A RETURN.  
I SAVE MACHINE....  
PRESTORE...  
ICHECK FOR ORIGINAL  
IASSENDING ORDER.

I STAC FAULT  
LOOP  
ITERATE TEST ROUTINE....  
I INCREMENT TO NEXT  
I STACK ENTRY.



```

10074 ECL21
01
02
03
04 05151 006230
05 05152 000100
06 05153 006221
07 05154 005171
08 05155 006234
09 05156 004462
10 05157 000462
11 05158 000462
12 05159 000462
13 05160 163710
14 05161 000000
15 05162 041777
16 05163 105300
17 05164 044040
18 05165 127710
19 05166 030041
20 05167 112415
21 05170 116414
22
23
24 05175 006231
25

```

```

RT3:
SETUP 100
JSR @ENTIN
100
STACK RT3A
JSR @ISTK
RT3A +2
JMP RT3A-1
MOVZ 0,0
SAVE 0
MOV 0,0
RTN
MOV 0,0,SZC
ERROR
LOOP
JSR @ENTLO
17
18 05103 006231
19
20
21 05104 006230
22 05105 000100
23
24 05106 006221
25 05107 005120
26 05110 004462
27 05111 000462
28 05112 101040
29 05113 163710
30 05114 000000
31 05115 101020
32 05116 127710
33 05117 101003
34
RT4A:
JSR @ENTLO
35
36 05124 006231
37
38
39 05125 006230
40 05126 000462
41
42 05127 006221
43 05130 005144
44
45 05131 006234
46 05132 004462
47 05133 000462
48 05134 163710
49 05135 000000
50 05136 041777
51 05137 127710
52 05140 024040
53 05141 030041
54 05142 116415
55 05143 112414
56
57
58 05150 006231
59

```

```

)TEST "RTN"
)INITIALIZE TEST.....

)INITIALIZE STACK
)FAULT ADDRESS IS RT3A

)CLEAR C(CARRY) AND
)SAVE THE FACT.

)SET C(CARRY) TO (1).
)AND RESTORE TO (0).

)ITERATE TEST ROUTINE.....

)TEST "RTN"
)INITIALIZE TEST.....

)INITIALIZE STACK
)FAULT ADDRESS IS RT4A

)SET C(CARRY) AND
)SAVE THE FACT.

)RESET C(CARRY) AND
)PRESTORE TO (1)

)ITERATE TEST ROUTINE.....

)TEST "RTN"
)INITIALIZE TEST.....

)INITIALIZE STACK
)FAULT ADDRESS IS RT5A

)C(CARRY)=RANDOM #
)STORE A RANDOM NUMBER
)IN TH STACK FRAME ENTRY
)OF THE STATE BLOCK. TEST

)RTN" RESTORE OF FP

)C(CARRY)=ORIG RANDOM
)C(CARRY)=STACK POINTER
)C(CARRY)=FRAME POINTER
)C(CARRY)=FP VIA RTN
)ITERATE TEST ROUTINE.....

```

```

)TEST "RTN"
)INITIALIZE TEST.....

)INITIALIZE STACK
)FAULT ADDRESS IS RT6B

)C(CARRY)=RANDOM #
)SET THE FP ENTRY,
)ON THE STACK, TO
)RANDOM. ALSO RANDOM
)TD C(SP).

)C(CARRY)=FP ENTRY FROM STACK
)C(CARRY)=NEW FP(FRAME POINTER)
)C(CARRY)=ORIG C(SP)
)C(CARRY)=CORRECT

)ITERATE TEST ROUTINE.....

```

10075 ECL21

```

01
02
03
04 05176 006231
05 05177 005240
06 05206 010040
07 05201 030040
08 05202 030040
09 05203 155400
10 05204 024842
11 05205 020020
12 05206 106400
13 05207 044273
14 05210 146400
15 05211 020015
16 05212 041000
17 05213 102400
18 05214 113710
19
20 05215 006230
21 05216 000020
22 05217 004402
23 05220 000411
24 05221 171000
25 05222 054240
26 05223 054040
27 05224 103710
28
29 05225 020015
30 05227 041773
31 05230 127710
32 05231 020240
33 05232 116014
34
35 05237 101001
36
37
38 05244 006231
39 05245 010240
40 05246 020240
41 05247 024273
42 05250 106414
43 05251 000744
44

```

```

RT8:
STACK RT8C
JSR #1STK
RT8C
LSD SP
STA 2,SP
INC 2,3
LDA 1,SL
LDA 0,*5
SUB 0,1
STA 1,TEM1
SUB 2,1
LDA 0,*JSR 0,2
STA 0,0,2
DAN
RT8A:
SETUP 20
JSR #ENTIN
20
JSR #2
JMP RT8B
LDA 3,SP
SAVE 0
LDA 0,*JSR 0,2
RTN
LDA 0,TEM
ADCM 0,3,SZR
ERROR
MOV 0,0,SKP
ERROR
LOOP
JSR #ENTLO
LSD TEM
LDA 0,TEM
LDA 1,TEM1
SUBM 0,1,SZR
JMP RT8A

```

```

RT9:
03 05252 006231
04 05253 005324
06 05254 010040
07 05255 030040
08 05256 052400
09 05257 155400
10 05260 024842
11 05261 020020
12 05262 106400
13 05263 044273
14 05264 146400
15 05265 020015
16 05266 041000
17 05267 102400
18 05270 113710
19
20 05271 006230
21 05272 000020
22 05273 020240
23 05274 040040
24 05275 040041
25 05276 004403
26 05277 000410
27 05300 000417
28 05301 171400
29 05302 103710
30 000000
31 05304 024914
32 05305 045773
33 05306 001773
34 05307 024841
35 05310 100415
36 05311 116414
37
38 05316 101001
39
40 05323 101001
41
42
43 05330 006231
44 05331 010240
45 05332 020240
46 05333 024273
47 05334 106414
48 05335 000734
49

```

```

STACK RT9C
JSR #1STK
RT9C
LSD SP
STA 2,SP
INC 2,3
LDA 1,SL
LDA 0,*5
SUB 0,1
STA 1,TEM1
SUB 2,1
LDA 0,*JSR 0,2
STA 0,0,2
DAN
RT9A:
SETUP 20
JSR #ENTIN
20
LDA 0,TEM
STA 0,SP
STA 0,FP
JSR #3
JMP RT9B
JMP RT9D
INC 3,2
SAVE 0
LDA 1,*RTN
STA 1,*5,3
JMP -5,3
LDA 1,FP
SUBM 0,1,SNR
SUBM 0,3,SZR
ERROR
MOV 0,0,SKP
ERROR
MOV 0,0,SKP
ERROR
LOOP
JSR #ENTLO
LSD TEM
LDA 0,TEM
LDA 1,TEM1
SUBM 0,1,SZR
JMP RT9A

```

10076 ECL21

```

01
02
03
04 05252 006231
05 05253 005324
06 05254 010040
07 05255 030040
08 05256 052400
09 05257 155400
10 05260 024842
11 05261 020020
12 05262 106400
13 05263 044273
14 05264 146400
15 05265 020015
16 05266 041000
17 05267 102400
18 05270 113710
19
20 05271 006230
21 05272 000020
22 05273 020240
23 05274 040040
24 05275 040041
25 05276 004403
26 05277 000410
27 05300 000417
28 05301 171400
29 05302 103710
30 000000
31 05304 024914
32 05305 045773
33 05306 001773
34 05307 024841
35 05310 100415
36 05311 116414
37
38 05316 101001
39
40 05323 101001
41
42
43 05330 006231
44 05331 010240
45 05332 020240
46 05333 024273
47 05334 106414
48 05335 000734
49

```

```

STACK RT9E
JSR #1STK
RT9E
LSD SP
STA 2,TEM
INC 2,3
LDA 1,SL
LDA 0,*5
SUB 0,1
STA 1,TEM1
SUB 2,1
LDA 0,*JSR 0,2
STA 0,0,2
DAN
RT9A:
SETUP 20
JSR #ENTIN
20
LDA 0,TEM
STA 0,SP
STA 0,FP
JSR #3
JMP RT9B
JMP RT9D
INC 3,2
SAVE 0
LDA 1,*RTN
STA 1,*5,3
JMP -5,3
LDA 1,FP
SUBM 0,1,SNR
SUBM 0,3,SZR
ERROR
MOV 0,0,SKP
ERROR
MOV 0,0,SKP
ERROR
LOOP
JSR #ENTLO
LSD TEM
LDA 0,TEM
LDA 1,TEM1
SUBM 0,1,SZR
JMP RT9A

```

10077 ECL21

```
01 10077 ECL21
02 *** TEST POPS ***
03
04
05
06
07 XPS0: SETUP 40
08 JSR #ENTIN
09 STACK XPS0A
10 JSR #1STK
11 XPS0A #06221
12 XPS0B #05347
13 JSR #+2
14 JMP XPS0B
15 SAVE 0
16 XPS0A: XPS0A #06230
17 JSR #ENTLO
18 XPS0B: XPS0B #06231
19 XPS1: SETUP 100
20 JSR #ENTIN
21 XPS1A: XPS1A #05356
22 STACK XPS1A
23 JSR #1STK
24 XPS1A #05357
25 XPS1B: XPS1B #05357
26 JSR #ENTRA
27 INC 0.1
28 INC 1.2
29 JSR #+2
30 SAVE 0
31 XPS1A: XPS1A #06231
32 JSR #ENTLO
33 XPS1B: XPS1B #05356
34 INC 1.2/SZR
35 JSR #ENTLO
36 XPS1A: XPS1A #05371
37 JSR #ENTLO
38 XPS1B: XPS1B #05376
39
40
```

```
10078 ECL21
01
02
03 XPS2: XPS2 #05377
04 JSR #1STK
05 XPS2 #05400
06 LDA #,SP
07 STA #,TEM
08 LDA #,SL
09 LDA 1,*7
10 SUB 1,*8
11 STA #,TEM1
12 SETUP 20
13 XPS2A: XPS2A #05406
14 JSR #ENTIN
15 20
16 LDA #,TEM
17 STA #,SP
18 RAND
19 JSR #ENTRA
20 INC 0.1
21 INC 1.2
22 JSR #+2
23 JMP #+4
24 SAVE 0
25 XPS2B: XPS2B #05422
26 XPS2C: XPS2C #05424
27 XPS2D: XPS2D #05424
28 XPS2E: XPS2E #05431
29 JSR #ENTLO
30 ISZ TEM
31 LDA #,TEM
32 LDA 1,*EM1
33 SUB #0.1,SZR
34 JMP XPS2A
35
```

```
TEST "POPB"
INITIALIZE STACK
IF AULT ADDRESS IS XPS2B
```

```
INITIALIZE TEST....
```

```
INIT STACK POINTER
```

```
IC(AC0)=RANDOM #
ISET C(AC0-2) TO ASCENDING,
IRANDOM, C(CAC1)+C(AC0),ETC.
IC(AC3)=RETURN.
```

```
ISAVE+RESTORE MACH STATE
```

```
ICHECK THE AC VALUES RESTORED.
IC(AC3) POINTS TO STACK
```

```
ITERATE TEST ROUTINE....
ADVANCE TO NEXT BUFFER
LOCATION.
TEST FOR END.
```





10883 ECL21

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43

XPSA1:  
STACK XPSAB  
JSR #1STK  
XPSAB  
ISZ SP  
LDA 2,SP  
LDA 1,SL  
STA 2,TEM  
INC 2,S  
LDA 0,S  
SUB 0,1  
STA 1,TEM1  
SUB 2,1  
LDA 0,#JSR 0,2  
STA 0,0,2  
SUB 0,0  
RAM  
XPSAA1:  
SETUP 10  
JSR #ENTIN  
LDA 3,TEM  
STA 3,SP  
JSR #S  
JMP XPSAC  
JMP XPSAB  
INC 3,2  
SAVE 0  
LDA 0,#POPB  
STA 0,#S,3  
JMP -S,3  
XPSAB1:  
ERROR  
LDA 0,#JSR 0,2  
STA 0,STEM  
LOOP  
JSR #ENTLO  
ISZ TEM  
LDA 0,TEM  
LDA 1,TEM1  
SUB# 0,1,SZR  
JMP XPSAA

10884 ECL21  
01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43

XPSB:  
STACK XPSBB\*1  
JSR #1STK  
ISZ SP  
LDA 2,SP  
LDA 1,SL  
STA 2,TEM  
INC 2,S  
LDA 0,S  
SUB 0,1  
STA 1,TEM1  
SUB 2,1  
LDA 0,#JSR 0,2  
STA 0,0,2  
SUB 0,0  
RAM  
XPSBA1:  
SETUP 10  
JSR #ENTIN  
LDA 0,STEM  
STA 0,SP  
JSR #S  
JMP XPSBB  
MOV 3,2  
LDA 1,#JSR 0,2  
STA 1,ASP  
LDA 3,SP  
SAVE 0  
POPB  
ADCR 0,3,SZR  
ERROR  
LOOP  
JSR #ENTLO  
ISZ TEM  
LDA 0,TEM  
LDA 1,TEM1  
SUB# 0,1,SZR  
JMP XPSBA

TEST "POPB"  
INITIALIZE STACK  
FAULT ADDRESS IS XPSBB\*1  
TEST THAT "POPB" WILL  
LOAD THE PROGRAM COUNTER  
WITH THE ADDRESS OF  
EVERY LOCATION IN A  
BUFFER.  
FILL MEMORY BUFFER  
(STACK) WITH JSR 0,2  
INSTRUCTIONS. WHEN THE  
BUFFER IS ENTERED  
INITIALIZE TEST....  
INITIALIZE STACK POINTER.  
SET RETURN ADDRESS.  
POPB ADDRESS WHERE POPB SHOULD GO  
NORMAL RETURN ADDRESS  
-1 ADDRESS WHERE IT WENT  
ITERATE TEST ROUTINE....  
ADVANCE TO NEXT STACK  
LOCATION.  
ITERATE TEST ROUTINE....

```

10085 ECL21
01
02
03 06024 006221
04 06025 006947
05 06026 020040
06 06027 040240
07 06030 020042
08 06031 024820
09 06032 122400
10 06033 040273
11 06034 006230
12 06035 000010
13 06036 020240
14 06037 040040
15 06040 004402
16 06041 000404
17 06042 103710
18 06043 000000
19 06044 107710
20 06045 024040
21 06046 106414
22 06046 106414
23
24
25 06053 006231
26 06054 010240
27 06055 020240
28 06056 024273
29 06057 106414
30 06060 000754
31

10086 ECL21
01
02 06061 014204
03 06062 000437
04 06063 010203
05 06064 101001
06 06065 063077
07 06066 020205
08 06067 040204
09
10 06070 060477
11 06071 101112
12 06072 000403
13 06073 122470
14 06075 000037
15 06075 143770
16 06076 004000
17 06077 101004
18 06100 000406
19
20 06101 006220
21 06102 001333
22 06103 120020
23 06104 024203
24 06105 006217
25
26 06106 034010
27 06107 021400
28 06110 101005
29 06111 000410
30 06112 015403
31 06113 000406
32 06114 002077
33 06115 021403
34 06116 035404
35 06117 001776
36 00120 001400
37 06121 010407
38 06122 002403
39 06123 002401
40 06124 000562
41 06125 000563
42
43
44

; INTERNAL COUNT DONE?
; INDE, LOOP BACK
; IYEP, RUMP PASS COUNT
; PASS CNT > 65K
; RESTORE INTERNAL COUNT.
; CHECK SWITCH 4
; IF SET, DO NOT
; PRINT PASS COUNT
; PRINT PASS COUNT
; STANDARD DTOS RETURN
; AUTO MODE?
; INO, RE=00 TEST
; IYES, PASS COUNT ZERO?
; INO, BACK TO TEST
; IYES, BACK TO DTOS

; END OF TEST
; PRESTART TEST...
; FOR START CAT THEN TEST

; ***** END OF TEST ROUTINES *****

```

10087 ECL21

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
00

! \*\*\*\*\*EGGS & DIRT DATA BLOCKS\*\*\*\*\*

EGGS:  
05 00126 00000 AUTO: 0  
06 00127 00000 DEVI: 0  
07 00130 00000 CATSW: 0  
08 00131 00000 PCNT: 0  
09 00132 00000 RTN: 0  
10 00133 00000 SWREG: 0  
11 00134 00000 VCTAB: .BLK 100

14 00234 000234 PRGEND: PRGEND  
15  
16 00235 .TXT /COPYRIGHT (C) DEC.1974,76

17 04452  
18 04410  
19 020124  
20 041450  
21 020051  
22 043504  
23 026103  
24 034401  
25 032007  
26 033454  
27 00251 040466 ALL RIGHTS RESERVED/  
28 046114  
29 051040  
30 043511  
31 022110  
32 020123  
33 042322  
34 042523  
35 053122  
36 042105  
37 000000  
38  
39

40 00264 141705 DIRT: .TXTE IECLIPSE212!  
41 144714  
42 051320  
43 131305  
44 131201  
45 000000  
46 06272 000000  
47 00273 000200  
48 04274 175772  
49 04275 000000  
50 06276 000000  
51 06277 000000  
52 00300 000000  
53 00301 000000  
54  
55

56 00013-107710  
57 127710  
58 005000  
59 000000  
00

DTOSB  
175772  
DTOSB  
DTOSB

.END DTOSB

00808 ECL21

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

000330  
000005  
117710  
005400  
103643  
003626  
001400  
103710  
000024  
000012  
000004  
000240  
177770  
000212  
104400  
022000  
000000  
000011  
033031  
000144  
000100  
177700  
000010  
000140  
000377  
000200  
000400  
001777







8093 ECL21

|              |       |       |       |       |       |  |  |  |  |  |
|--------------|-------|-------|-------|-------|-------|--|--|--|--|--|
| MULH1 001570 | 31/39 |       |       |       |       |  |  |  |  |  |
| MULH2 001685 | 31/52 |       |       |       |       |  |  |  |  |  |
| MULH3 001611 | 31/56 |       |       |       |       |  |  |  |  |  |
| MGCAT 000225 | 15/19 | 15/28 |       |       |       |  |  |  |  |  |
| NSRT 000500  | 12/47 | 15/05 |       |       |       |  |  |  |  |  |
| OFF 000170   | 12/29 |       |       |       |       |  |  |  |  |  |
| OK0 000304   | 14/15 |       |       |       |       |  |  |  |  |  |
| OK1 000305   | 14/15 |       |       |       |       |  |  |  |  |  |
| OK2 000306   | 14/17 |       |       |       |       |  |  |  |  |  |
| OK3 000307   | 14/18 |       |       |       |       |  |  |  |  |  |
| DN 000171    | 12/30 |       |       |       |       |  |  |  |  |  |
| ORIG0 000300 | 14/11 | 24/29 |       |       |       |  |  |  |  |  |
| ORIG1 000301 | 14/12 | 24/30 |       |       |       |  |  |  |  |  |
| ORIG2 000302 | 14/13 | 24/31 |       |       |       |  |  |  |  |  |
| ORIG3 000303 | 14/14 | 24/23 | 24/32 |       |       |  |  |  |  |  |
| DVLM 000174  | 9/02  |       |       |       |       |  |  |  |  |  |
| CVT1 000227  | 9/14  |       |       |       |       |  |  |  |  |  |
| CVTRP 000301 | 9/34  |       |       |       |       |  |  |  |  |  |
| PASHE 001333 | 29/02 |       |       |       |       |  |  |  |  |  |
| PASS 000203  | 12/48 |       |       |       |       |  |  |  |  |  |
| PASS1 000204 | 12/49 | 28/18 | 22/32 | 06/04 | 08/23 |  |  |  |  |  |
| PASSV 000205 | 12/50 | 06/02 | 06/08 |       |       |  |  |  |  |  |
| PK 000327    | 14/37 | 06/07 |       |       |       |  |  |  |  |  |
| PCNT 000131  | 07/08 |       |       |       |       |  |  |  |  |  |
| PDEC 001100  | 13/10 |       |       |       |       |  |  |  |  |  |
| PDEC1 001205 | 25/26 |       |       |       |       |  |  |  |  |  |
| PDEC2 001217 | 25/32 |       |       |       |       |  |  |  |  |  |
| PDEC3 001224 | 25/07 |       |       |       |       |  |  |  |  |  |
| PDERE 000255 | 13/46 | 25/29 | 25/41 |       |       |  |  |  |  |  |
| POIVR 000312 | 14/21 | 25/15 | 25/45 |       |       |  |  |  |  |  |
| PERCE 001225 | 21/25 |       |       |       |       |  |  |  |  |  |
| PF 000003    | 11/12 | 33/12 |       |       |       |  |  |  |  |  |
| PJ0 003537   | 56/03 |       |       |       |       |  |  |  |  |  |
| PJ0A 003552  | 56/08 | 56/16 |       |       |       |  |  |  |  |  |
| PJ1 003555   | 56/23 |       |       |       |       |  |  |  |  |  |
| PJ1A 003600  | 56/28 | 56/30 |       |       |       |  |  |  |  |  |
| PJ2 003613   | 56/33 |       |       |       |       |  |  |  |  |  |
| PJ2A 003622  | 56/48 | 56/52 |       |       |       |  |  |  |  |  |
| PJ2B 003620  | 56/49 | 56/53 |       |       |       |  |  |  |  |  |
| PJ3 003627   | 57/03 |       |       |       |       |  |  |  |  |  |
| PJ30 003651  | 57/19 |       |       |       |       |  |  |  |  |  |
| PJ30A 003655 | 57/24 | 57/40 |       |       |       |  |  |  |  |  |
| PJ30B 003665 | 57/21 | 57/33 |       |       |       |  |  |  |  |  |
| PJ3A 003643  | 57/10 | 57/14 |       |       |       |  |  |  |  |  |
| PJ3B 003644  | 57/08 | 57/15 |       |       |       |  |  |  |  |  |
| PJ4 003677   | 58/03 |       |       |       |       |  |  |  |  |  |
| PJ4A 003714  | 58/17 | 58/44 |       |       |       |  |  |  |  |  |
| PJ4B 003725  | 58/27 |       |       |       |       |  |  |  |  |  |
| PJ4C 003731  | 58/25 | 58/31 |       |       |       |  |  |  |  |  |
| PJ4D 003748  | 58/30 | 58/35 |       |       |       |  |  |  |  |  |
| PJ4E 003742  | 58/05 | 58/05 |       |       |       |  |  |  |  |  |
| PJ5 003754   | 59/03 |       |       |       |       |  |  |  |  |  |
| PJ55 003775  | 59/17 | 59/22 | 59/25 | 59/36 | 59/38 |  |  |  |  |  |
| PJ56 004006  | 59/21 | 59/24 |       |       |       |  |  |  |  |  |
| PJ5A 003757  | 59/15 |       |       |       |       |  |  |  |  |  |
| PJ5B 003772  | 59/18 | 59/40 |       |       |       |  |  |  |  |  |
| PJ5C 004022  | 59/03 | 59/35 |       |       |       |  |  |  |  |  |
| PJ6 004034   | 60/03 |       |       |       |       |  |  |  |  |  |

8094 ECL21

|              |       |       |       |       |       |       |  |  |  |  |
|--------------|-------|-------|-------|-------|-------|-------|--|--|--|--|
| PJ6A 004036  | 50/06 | 50/22 |       |       |       |       |  |  |  |  |
| PJ6B 004072  | 50/06 | 50/36 |       |       |       |       |  |  |  |  |
| PMULR 000310 | 14/19 |       |       |       |       |       |  |  |  |  |
| POCT 001172  | 13/09 | 25/14 |       |       |       |       |  |  |  |  |
| PP0 002626   | 44/06 |       |       |       |       |       |  |  |  |  |
| PP0A 002644  | 44/11 | 44/19 |       |       |       |       |  |  |  |  |
| PP1 002651   | 44/23 |       |       |       |       |       |  |  |  |  |
| PP10 003176  | 50/02 | 50/26 |       |       |       |       |  |  |  |  |
| PP10A 003202 | 50/07 | 50/19 |       |       |       |       |  |  |  |  |
| PP10B 003214 | 50/06 |       |       |       |       |       |  |  |  |  |
| PP11 003226  | 50/28 |       |       |       |       |       |  |  |  |  |
| PP11A 003242 | 50/33 | 50/42 |       |       |       |       |  |  |  |  |
| PP12 003247  | 51/03 |       |       |       |       |       |  |  |  |  |
| PP12A 003263 | 51/08 | 51/17 |       |       |       |       |  |  |  |  |
| PP13 003270  | 51/21 |       |       |       |       |       |  |  |  |  |
| PP13A 003303 | 51/26 | 51/35 |       |       |       |       |  |  |  |  |
| PP14 003310  | 51/39 |       |       |       |       |       |  |  |  |  |
| PP14A 003321 | 51/44 | 51/51 |       |       |       |       |  |  |  |  |
| PP1A 002667  | 44/28 | 44/36 |       |       |       |       |  |  |  |  |
| PP2 002574   | 44/48 |       |       |       |       |       |  |  |  |  |
| PP2A 002704  | 44/45 | 44/51 |       |       |       |       |  |  |  |  |
| PP3 002711   | 45/02 |       |       |       |       |       |  |  |  |  |
| PP3A 002721  | 45/07 | 45/13 |       |       |       |       |  |  |  |  |
| PP4 002726   | 45/17 |       |       |       |       |       |  |  |  |  |
| PP4A 002746  | 45/22 | 45/35 |       |       |       |       |  |  |  |  |
| PP5 002753   | 46/03 |       |       |       |       |       |  |  |  |  |
| PP5A 003002  | 46/08 | 46/29 |       |       |       |       |  |  |  |  |
| PP6 003007   | 47/02 |       |       |       |       |       |  |  |  |  |
| PP6A 003025  | 47/18 | 47/23 |       |       |       |       |  |  |  |  |
| PP6B 003037  | 47/07 | 47/28 |       |       |       |       |  |  |  |  |
| PP7 003044   | 48/03 |       |       |       |       |       |  |  |  |  |
| PP7A 003062  | 48/19 | 48/28 |       |       |       |       |  |  |  |  |
| PP7B 003100  | 48/08 | 48/33 |       |       |       |       |  |  |  |  |
| PP8 003105   | 49/01 |       |       |       |       |       |  |  |  |  |
| PP8A 003111  | 49/06 | 49/26 |       |       |       |       |  |  |  |  |
| PP8B 003124  | 49/05 | 49/19 |       |       |       |       |  |  |  |  |
| PP9 003136   | 49/29 |       |       |       |       |       |  |  |  |  |
| PP9A 003142  | 49/34 | 49/58 |       |       |       |       |  |  |  |  |
| PP9B 003164  | 49/33 | 49/51 |       |       |       |       |  |  |  |  |
| PP8 003326   | 52/03 |       |       |       |       |       |  |  |  |  |
| PP8A 003341  | 52/08 | 52/16 |       |       |       |       |  |  |  |  |
| PP1 003346   | 52/20 |       |       |       |       |       |  |  |  |  |
| PP1A 003351  | 52/25 | 52/33 |       |       |       |       |  |  |  |  |
| PP2 003366   | 52/37 |       |       |       |       |       |  |  |  |  |
| PP2A 003400  | 52/42 | 52/49 |       |       |       |       |  |  |  |  |
| PP3 003405   | 53/02 |       |       |       |       |       |  |  |  |  |
| PP3A 003411  | 53/07 | 53/22 |       |       |       |       |  |  |  |  |
| PP3B 003420  | 53/04 | 53/15 |       |       |       |       |  |  |  |  |
| PP4 003432   | 53/24 |       |       |       |       |       |  |  |  |  |
| PP4A 003442  | 53/29 | 53/34 |       |       |       |       |  |  |  |  |
| PP5 003447   | 54/02 |       |       |       |       |       |  |  |  |  |
| PP5A 003453  | 54/07 | 54/25 |       |       |       |       |  |  |  |  |
| PP5B 003465  | 54/04 | 54/16 |       |       |       |       |  |  |  |  |
| PP6 003477   | 55/03 |       |       |       |       |       |  |  |  |  |
| PP6A 003523  | 55/25 | 55/30 |       |       |       |       |  |  |  |  |
| PP6B 003532  | 55/08 | 55/27 |       |       |       |       |  |  |  |  |
| PP6C 003536  | 55/31 | 55/33 |       |       |       |       |  |  |  |  |
| PP6EN 006234 | 13/04 | 13/05 | 13/18 | 13/19 | 15/14 | 07/14 |  |  |  |  |



0897 ECL121

0898 ECL21

|              |       |       |       |       |       |       |       |       |       |       |       |       |       |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| SRAN 080277  | 62/36 | 62/40 | 63/09 | 63/15 | 63/25 | 63/35 | 63/39 | 63/31 | 63/49 | 27/10 | 86/13 | 87/10 | 47/22 |
| STACK 080827 | 64/05 | 64/19 | 64/23 | 65/06 | 65/20 | 65/32 | 66/25 | 63/24 | 63/42 | 43/18 | 43/24 | 47/14 | 54/06 |
|              | 66/35 | 67/26 | 67/36 | 70/16 | 70/16 | 71/32 | 71/42 | 63/24 | 63/42 | 53/18 | 53/18 | 53/19 | 57/27 |
|              | 73/52 | 74/17 | 75/06 | 75/07 | 75/26 | 76/06 | 76/07 | 64/11 | 64/23 | 55/14 | 55/24 | 57/23 | 57/27 |
|              | 78/23 | 78/05 | 78/15 | 79/05 | 79/15 | 82/06 | 82/16 | 64/04 | 64/26 | 58/31 | 58/31 | 58/40 | 58/41 |
|              | 82/26 | 83/05 | 83/08 | 83/22 | 84/06 | 84/07 | 84/23 | 65/03 | 65/44 | 56/30 | 56/30 | 63/34 | 63/45 |
|              | 84/28 | 84/29 | 85/05 | 85/15 | 85/21 |       |       | 65/12 | 65/37 | 64/21 | 64/26 | 66/48 | 66/49 |
|              | 14/08 | 35/12 | 35/29 | 35/46 | 36/06 | 36/22 | 37/06 | 66/03 | 66/18 | 64/26 | 64/26 | 65/07 | 65/19 |
|              | 37/24 | 38/06 | 38/28 | 39/06 | 39/26 | 40/06 | 40/34 | 66/06 | 66/37 | 64/26 | 64/26 | 65/07 | 65/19 |
|              | 41/06 | 41/25 | 42/05 | 43/06 | 44/09 | 44/26 | 44/43 | 66/06 | 66/37 | 64/26 | 64/26 | 65/07 | 65/19 |
|              | 45/05 | 45/20 | 46/06 | 47/05 | 48/06 | 49/03 | 49/31 | 66/06 | 66/37 | 64/26 | 64/26 | 65/07 | 65/19 |
|              | 50/04 | 50/31 | 51/06 | 51/24 | 51/42 | 52/06 | 52/23 | 66/06 | 66/37 | 64/26 | 64/26 | 65/07 | 65/19 |
|              | 52/40 | 53/02 | 53/27 | 54/02 | 55/06 | 56/06 | 56/26 | 66/06 | 66/37 | 64/26 | 64/26 | 65/07 | 65/19 |
|              | 56/46 | 57/06 | 57/19 | 58/03 | 59/03 | 60/06 | 60/06 | 66/06 | 66/37 | 64/26 | 64/26 | 65/07 | 65/19 |
|              | 61/26 | 61/43 | 62/06 | 62/23 | 63/06 | 63/22 | 64/02 | 66/06 | 66/37 | 64/26 | 64/26 | 65/07 | 65/19 |
|              | 65/03 | 66/06 | 66/22 | 67/06 | 67/23 | 68/06 | 68/21 | 66/06 | 66/37 | 64/26 | 64/26 | 65/07 | 65/19 |
|              | 68/37 | 69/05 | 69/24 | 70/03 | 71/09 | 71/29 | 72/06 | 66/06 | 66/37 | 64/26 | 64/26 | 65/07 | 65/19 |
|              | 73/05 | 73/23 | 73/41 | 74/06 | 75/03 | 76/03 | 77/05 | 66/06 | 66/37 | 64/26 | 64/26 | 65/07 | 65/19 |
|              | 77/24 | 78/02 | 79/02 | 79/43 | 80/05 | 80/28 | 81/05 | 66/06 | 66/37 | 64/26 | 64/26 | 65/07 | 65/19 |
|              | 81/27 | 82/03 | 83/02 | 84/03 | 85/02 |       |       | 66/06 | 66/37 | 64/26 | 64/26 | 65/07 | 65/19 |
|              | 18/16 | 16/27 | 17/03 | 86/40 | 86/41 |       |       | 66/06 | 66/37 | 64/26 | 64/26 | 65/07 | 65/19 |
|              | 13/12 | 18/32 |       |       |       |       |       | 66/06 | 66/37 | 64/26 | 64/26 | 65/07 | 65/19 |
| START 080563 | 61/06 | 61/19 | 61/28 | 66/52 | 66/54 | 67/19 | 67/23 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| STK 080646   | 61/11 | 61/23 | 66/52 | 66/54 | 67/19 | 67/23 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV0 080677   | 61/23 | 66/52 | 66/54 | 67/19 | 67/23 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV0A 080412  | 66/52 | 66/54 | 67/19 | 67/23 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV1 080417   | 66/54 | 67/19 | 67/23 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV10 080445  | 67/19 | 67/23 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV18A 080455 | 67/23 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV18B 080471 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV11 080483  | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV12 080520  | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV12A 080525 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV12B 080535 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV12C 080551 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV13 080573  | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV13A 080604 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV14 080611  | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV14A 080623 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV15 080630  | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV15A 080642 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV16 080647  | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV16A 080653 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV17 080670  | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV17A 080670 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV18 080711  | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV18A 080721 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV18B 080736 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV1A 080432  | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV2 080437   | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV2A 080451  | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV3 080456   | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV3A 080456  | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV4 080491   | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV4A 080491  | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV4B 080491  | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV5 080426   | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV5A 080426  | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |
| SV6 080480   | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/25 | 67/46 | 67/51 | 67/51 | 67/51 | 67/51 |

0099 ECL21

|               |       |       |
|---------------|-------|-------|
| XP84A 005521  | 79/45 | 79/57 |
| XP85 005526   | 80/82 |       |
| XP85A 005543  | 80/87 | 80/28 |
| XP86 005552   | 80/25 |       |
| XP86A 005570  | 80/38 | 80/42 |
| XP87 005575   | 81/02 |       |
| XP87A 005614  | 81/07 | 81/20 |
| XP88 005621   | 81/24 |       |
| XP88A 005636  | 81/29 | 81/40 |
| XP89 005643   | 82/03 |       |
| XP89A 005653  | 82/12 | 82/37 |
| XP89B 005671  | 82/28 |       |
| XP89C 005673  | 82/65 | 82/30 |
| XP8A 005785   | 83/02 |       |
| XP8AA 005724  | 83/18 | 83/41 |
| XP8AB 005741  | 83/04 | 83/25 |
| XP8AC 005747  | 83/24 | 83/35 |
| XP8B 005755   | 84/03 |       |
| XP8BA 005774  | 84/19 | 84/41 |
| XP8BB 005811  | 84/05 | 84/25 |
| XP8BC 005816  | 84/35 |       |
| XP8C 005824   | 85/02 |       |
| XP8CA 005834  | 85/11 | 85/30 |
| XP8CB 005847  | 85/04 | 85/23 |
| ZEROA 006151  | 8/13  |       |
| .BRAN 001113  | 13/26 | 24/85 |
| .EGGS 000019- | 12/22 | 25/31 |
| .RAND 001115  | 13/25 | 24/06 |
| .RAN\$ 001133 | 13/14 | 24/23 |







**DataGeneral**

---

---

**DIAGNOSTIC  
LISTING**

---

---

LISTING

096-000243-73

PROGRAM

EXERCISER FOR ECLIPSE  
PART 5

TAPE

095-000228-73

ABSTRACT

'ECLIPSE22' IS AN EXERCISER PROGRAM USED TO TEST THE RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF THE ECLIPSE COMPUTER. 'ECLIPSE22' EXERCISES THE EXTENDED INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES OF ITS RELIABLE OPERATION.

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



0001 ECL22 MACRO REV 04.00P  
14107152 04/22/77

10002 ECL22  
02  
03  
04  
05  
06  
07

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

```
*****  
/ NAME: ECLIPSE22.TX PART NUMBER: 097-000243  
/ DESCRIPTION: ECLIPSE EXERCISER, PART 5  
/ TEXT FILE  
/ REVISION HISTORY:  
/ REV. DATE  
/ 00 08/02/74  
/ 01 12/20/74  
/ 02 05/06/76  
/ 03 04/22/77  
/ COPYRIGHT (C) DATA GENERAL CORPORATION, 1974, 1976, 1977  
/ ALL RIGHTS RESERVED.  
*****
```

TITL ECL22  
ECLIPSE22  
ECLIPSE22 - CONTINUATION OF ECLIPSE21  
PART 5 OF EXERCISER FOR ECLIPSE

10003 ECL22

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
EXERCISER FOR ECLIPSE: PART 5
PROGRAM NAME
ECLIPSE22
GENERAL DESCRIPTION
ECLIPSE21 IS AN EXERCISER PROGRAM USED TO TEST THE
RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF
THE ECLIPSE COMPUTER. ECLIPSE22 EXERCISES THE EXTENDED
INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES
OF ITS RELIABLE OPERATION.
THE INSTRUCTIONS EXERCISED ARE AS FOLLOWS:
XOP, DIVX AND XCT, STACK OVERFLOW AND UNDERFLOW
LOCATIONS 200 TO 205 IN PAGE 0 ARE FIXED FOR ECLIPSE22
PROGRAM.
LOCATION 203 KEEPS TRACK OF NUMBER OF PASSES RUN
THROUGH ECLIPSE22 PROGRAM.
LOCATION 201 KEEPS TRACK OF THE TEST RUNNING AT
PRESENT AND IS USEFUL FOR DEBUG WHEN LOOPING
OCCURS IN THE PROGRAM.
LOCATION 202 CONTAINS THE STARTING ADDRESS OF
ECLIPSE22 PROGRAM.
LOCATION 200 IS USED BY DTOS.
LOCATION 204 KEEPS TRACK OF INTERNAL PASS COUNT
WHICH IS FIXED BY LOCATION 205.
FIRST PASS THROUGH ECLIPSE22 TEST WILL RUN SUPERFAST.
NEXT PASSES WILL RUN SLOWER AS EACH TEST IS RUN SEVERAL
TIMES TO RUN ALL RANDOM NUMBER COMBINATIONS.
MACHINE REQUIREMENTS
ECLIPSE PROCESSOR
4K READ-WRITE MEMORY
CONSOLE EQUIPMENT

```

10004 ECL22

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
SWITCH SETTINGS
THIS PROGRAM USES DATA SWITCHES AS FOLLOWS
SW"0" - USE CONTENTS OF "SWREG" IF 0
USE DATA SWITCHES IF 1
SW"1" - LOOP ON FAILING TEST IF 0
PROCEED TO NEXT TEST IF 1
SW"2" - OUTPUT TO TTY IF 0
INHIBIT PRINTING TO TTY IF 1
SW"3" - DO NOT PRINT X ERRORS IF 0
PRINT FAILURE RATE IF 1
SW"4" - PRINT PASS COUNT IF 0
INHIBIT PRINTING PASS COUNT IF 1
SW"5" - INHIBIT OUTPUT TO LINE PRINTER IF 0
OUTPUT TO LINE PRINTER IF 1
EDTOS PRESUMES THAT ALL SWITCHES ARE 0. RETURN IS
MADE TO THE OPER SYS UPON AN ERROR AFTER FIRST PRINTING
THE FAILURE REPORT. IN STAND ALONE MODE, THE OPERATOR
IS ASKED TO PRESET THESE SWITCHES (THE MACHINE HALTS
FOR HIM TO DO SO) AND PRESS CONTINUE. THIS PRESET VALUE
CAN BE CHANGED AT ANY TIME BY SETTING DATA SWITCH MAP
TO A ONE. THE PROGRAM WILL THEN FOLLOW ACTUAL SETTINGS.
STAND ALONE STARTING ADDRESS = 200
IF 'ICAT1' OR 'KITTEN1' WAS LOADED FROM DTOS AND RESTRY
WAS NEEDED, THEN USE AS FOLLOWS
STARTING ADDR = 170 (FOR START WITH NO 'ICAT1')
STARTING ADDR = 171 (FOR START WITH 'ICAT1')
MONITOR LOCATION 203 TO CHECK THE CURRENT PASS COUNT
MONITOR LOCATION X68700 TO MAKE SURE THAT 'ICAT1' OR
'KITTEN1' IS RUNNING. IN CASES WHERE PROGRAM IS
STARTED WITH 'ICAT1' OR 'KITTEN1' LOCATION X68000 WILL SHOW
A PATTERN CHANGING FROM ZEROS) TO ALL ONES
TO AN INC/SWAP PATTERN.
(CX= THE NUMBER OF THE HIGHEST MEMORY MODULE IN THE
SYSTEM AND MAY BE A VALUE 0 - 7)

```

10005 ECL22

```

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

```

SWITCH SETTINGS  
-----

THIS PROGRAM USES THE DATA SWITCHES AS FOLLOWS:

SW"0" - USE CONTENTS OF "SWREG" IF 0  
SW"1" - LOOP ON FAILING TEST IF 0  
PROCEED TO NEXT TEST IF 1  
SW"2" - PRINT ERROR REPORT IF 0  
DO NOT PRINT IF 1  
SW"3" - DO NOT PRINT X ERRORS IF 0  
PRINT FAILURE RATE IF 1  
SW"4" - DO NOT PRINT ERRORS IF 0  
PRINT AFTER 1ST ERROR IF 1

EDTOS PRESUMES THAT ALL SWITCHES ARE 0. RETURN IS  
MADE TO THE OPER SYS UPON AN ERROR AFTER FIRST PRINTING  
THE FAILURE REPORT. IN STAND ALONE MODE, THE OPERATOR  
IS ASKED TO PRESET THESE SWITCHES (THE MACHINE HALTS  
FOR HIM TO DO SO) AND PRESS CONTINUE. THIS PRESET VALUE  
CAN BE CHANGED AT ANY TIME BY SETTING DATA SWITCH "0"  
TO A "1". THE PROGRAM WILL THEN FOLLOW ACTUAL SETTINGS.

14.2 STARTING ADDRESS = 200  
14.3 MONITOR LOCATION 205 TO CHECK THE CURRENT PASS COUNT.

10006 ECL22

```

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

```

OPERATING PROCEDURE/OPERATOR INPUT  
-----

15.1 LOAD THE PROGRAM VIA THE BINARY LOADER OR INSERT A  
PRELOADED MEMORY MODULE.  
15.2 SET SWITCHES TO 200.  
15.3 PRESS START.  
PROGRAM WILL HALT AFTER PRINTING THE MESSAGE  
"SET DATA SWITCHES AND PRESS CONTINUE".  
15.4 SET REQUIRED SWITCH SETTINGS AND PRESS CONTINUE.  
THE PROGRAM WILL RUN UNTIL MANUALLY STOPPED. IN CASE  
OF MALFUNCTIONING, THE PROGRAM WILL PRINT ERROR  
MESSAGE AND TAKE APPROPRIATE ACTION AS PER THE SM  
SETTINGS.

PROGRAM OUTPUT/ERROR DESCRIPTION  
-----

FOR ANY ERRORS DETECTED, THE PROGRAM WILL PRINT ERROR  
REPORT OR X FAILURES DEPENDING UPON THE SM SETTINGS.  
ERROR REPORT CONSISTS OF ALL ACCUMULATORS, CARRY,  
RELOCATED PROGRAM COUNTER OF THE TEST THAT IS FAILING  
AND PC IN THE LISTING AT THE TIME OF FAILURE.  
16.3 THE PROGRAM WILL LOOP IN THE TEST THAT IS FAILING IF  
SM"1" IS 0.  
16.4 THE PRINTING OF ERROR REPORT CAN BE ABORTED BY SETTING  
SM"2" TO 1.  
16.5 IF LOOPING OCCURS IN THE PROGRAM, STOP THE COMPUTER  
AND CHECK LOCATION 201 TO FIND OUT THE TEST THAT WAS  
RUNNING BEFORE THE LOOPING OCCURRED.

10007 ECL22

10008 ECL22

01  
02

.EOT

PROGRAM DESCRIPTION/THEORY OF OPERATION  
-----

17. EACH TEST IS COMPLETE IN ITSELF, SO THE PROGRAM CAN  
BE STARTED FROM ANY TEST WITHOUT CAUSING ANY  
INITIALIZATION ERRORS.  
17.1 WHEN 'ECLIPSE22' IS STARTED AT LOCATION 2000 OR BY  
DTOS, IT WILL SIZE UP THE MEMORY AND WILL PRINT  
17.2 THE TOP OF THE MEMORY.  
11 AFTER SETTING UP THE SWITCHES AND PRESSING CONTINUE,  
12 THE EXERCISER WILL RUN THE FIRST PASS VERY FAST, IN  
13 THE FIRST PASS EACH TEST IS RUN ONLY ONCE. ALL OTHER  
14 PASSES WILL TAKE MORE TIME AS EACH TEST IS RUN  
15 ACCORDING TO THE TEST COUNT SPECIFIED IN EACH TEST.  
16 REFER TO THE LISTING TO FIND OUT THE INFORMATION  
17 ABOUT EACH TEST.

RESTRICTIONS/MISC  
-----

18. CERTAIN INSTRUCTIONS LIKE BLM, XCT, BAM, ETC.,  
DO ALLOW INTERRUPTS TO OCCUR DURING THEIR  
EXECUTION. THIS FEATURE OF THOSE INSTRUCTIONS IS  
NOT CHECKED IN THIS TEST.

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

0009 ECL22

00010 ECL22

```

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
*****
NAME: ECLIPSE22,SR
PART NUMBER: 094-000027
*****
DESCRIPTION: ECLIPSE EXERCISER, PART 5
REVISION HISTORY:
REV. DATE
00 08/02/74
01 12/20/74
02 05/05/76
03 04/22/77
*****
COPYRIGHT (C) DATA GENERAL CORPORATION, 1974, 1976, 1977
ALL RIGHTS RESERVED.
*****

```

```

1 *** MACRO DEFINITIONS ***
.MACRO LOOP
JSR @ENTLO
ITERATE TEST ROUTINE.....
X
.MACRO SETUP
JSR @ENTIN
INITIALIZE TEST.....
X
.MACRO RAND
JSR @ENTRA
IC(A0)=RANDOH #
X
.MACRO BRANU
JSR @ENTRR
X
.MACRO ERROR
JMP *+2
** JMP *+3
** STA @JAC3
** JSR @ENTER
X
.MACRO STACK
JSR @ISTK
INITIALIZE STACK
IF AULT ADDRESS IS #1
X
.MACRO PSHSP
SETUP @0
STACK PSHSA1
MOV @0,0
PSH A1,A1
LDA @0,SP
LDA 1,BBEG
MOV @0,@SZC
SUB @0,1,SZH
PSHSA1: ERROR
LOOP
X
.MACRO TRAPER
JSR @ITRP
IF STACK FAULT, GO TO A1
IF TRAP ORIGIN IN CONTENTS OF #2
IF THE CAP NUMBER IS #3
IF SUBROUTINE ADDRESS IS #4
IF STACK POINTERS SET TO C(BBEG)
X

```

10011 ECL22

```

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

```

.MACRO TRYST  
 LDA 0, RBEG  
 LDA 1, \*+1+1  
 ADD 0, 1  
 XOR 1, 1, A2  
 SUB# 2, 3, 8NR  
 SUB# 1, 2, 5ZR  
 ERROR  
 X

.MACRO ZERDAC  
 SUB 0, 0  
 SUB 1, 1  
 SUB 2, 2  
 SUB 3, 3  
 X

.MACRO BAKER  
 STA 3, ACS  
 \*\* JSR 0, ENTEB  
 \*\* JSR 0, ENTEB  
 X

.MACRO NDERR  
 STA 3, ACS  
 \*\* JSR 0, ENTEB  
 X

.MACRO RANST  
 JSR 0, I, RAN  
 X

.TEST FOR ACS/ACD ADDRESS  
 STORED IN C(AC2=3).C(AC1)  
 INCORRECT STACK LOCATION  
 ICAP # A2  
 X

.ZERO ALL AC1'S  
 X

.RANDOM # TO ACS=2  
 X

10012 ECL22

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

```

.MACRO OVLIM  
 LDA 0, \*+2  
 MOVZ 0, 0, SRP  
 A2  
 STA 0, SF  
 LDA 0, A1  
 STA 0, SP  
 STA 0, SL  
 STA 0, FP  
 X

.MACRO OVT1  
 SETUP 200  
 OVLIM RBEG, OVIRA1  
 LDA 1, \*+1  
 ADD 0, 1  
 STA 1, SL  
 SAVE 2  
 ERROR  
 OVI0A1: LOOP  
 X

.MACRO UFLIM  
 LDA 0, \*PRK  
 STA 0, SP  
 LDA 0, RBEG  
 STA 0, SL  
 LDA 0, \*+1  
 STA 0, SF  
 X

.MACRO OVTRP  
 LDA 0, RBEG  
 STA 0, SP  
 STA 0, SL  
 STA 0, TO  
 LDA 1, \*+5  
 STA 1, \*TO  
 LDA 1, \*+4  
 STA 1, SF  
 JMP \*+3  
 A2  
 A1  
 X

.THE ADDRESS TO GO TO  
 .ION OVERFLOW IS A2.  
 .SET C(AC0), STACK, AND  
 .FRAME POINTERS TO  
 .THE CONTENTS OF A1.  
 .TEST OF STACK OVERFLOW.  
 X

.STACK LIMIT IS A1  
 .GREATER THEN STACK POINTER.  
 .OVERFLOW SHOULD OCCURE.  
 X

.SETUP STACK IN PAGE 0  
 .TO UNDER FLOW ON ANY  
 .PDP, POPJ, RTN, OR PORB.  
 .INSTRUCTION, THE FAULT  
 .LOCATION IS A1.  
 X

.TEST OVERFLOW OF THE STACK  
 .ION THE CAP INSTRUCTION.  
 .IF NO OV GO TO A2.  
 .IF OV GO TO A1.  
 X



10013 ECL22

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

;*****LITERAL DEFINITIONS*****
SP=40
FP=41
SL=42
SF=43
TD=44
FF=45
SC=2
PF=3
ISP=4
MX=5
ISL=6
ISF=7

;*****LITERAL DEFINITIONS*****
;*****LITERAL DEFINITIONS*****

;DISPATCH TABLE AT LOCATION
;VCTAB... ENTRY 0 SET
;TO ADDRESS A1, OTHERS
;TO ADDRESS A2.

;FILL THE BUFFER WITH
;A1.

;ADVANCE TO NEXT BUFFER
;LOCATION AND TEST FOR
;END OF BUFFER=110, IF NOT
;END OF BUFFER GO TO A1.

;FIND THE FINAL ADDRESS
;AND VALUE IN THE INDIRECT
;CHAIN, C(CAC2)=FIRST
;ADDRESS AT WHICH TO
;START LOOKING...
;IF INDIRECT CHAIN IS
;110, EXIT TO A1.

;AUTO INDEX REGISTERS
;ARE USED AS VARIABLES.

;MAKE SURE THAT FINAL
;DATA WILL NOT POINT
;TO ANY SPOT IN THE CHAIN.
;IF IT DOES, EXIT TO A1.

;C(CAC2)=FINAL ADDRESS
;C(CAC3)=DATA AT THAT
;ADDRESS.

```

10014 ECL22

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16

```

```

10015 ECL22
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58

/ ***** LOCAL ZREL *****
/ ***** DIAGNOSTIC PROGRAM PREAMBLE *****
.LOC 0
DIRT 0
.LOC 40
.LOC 0
+1
HALT
.LOC 45
EGGS
.ZREL
.BLK 8.
EGGS: EGGS
MENTOP: 0
ICAT: 0
/ SPACE RESERVED FOR CAT/NO CAT RESTART ENTRIES
.LOC 170
SUBO 0/0,SKP
ADC 0/0
ESTA 0/CATSW
JMP 0/+1
SPMESS
/ SPACE RESERVED FOR SPECIAL RESTART ENTRIES
.LOC 176
.BLK 2
/ LOCATIONS 200 = 213 RESERVED FOR ECLIPSE TESTS
.LOC 200
JMP
DTOSB: 0
ITRETI: 0
SGNADR: 0
PASSI: 0
PASSVLI: 1
ITRI: 0
ITRETI: 0
ITRETI: 0
ERRETI: 0
LOPRETI: 0
/ TEMPORARY STORAGE FOR MACHINE STATE IN LOOP, ERROR AND INIT SUBROUTINES
/ TEMPORARYS FOR
/ PRINT ROUTINES
ICONSTANT 0
/ ***** LOCAL ZREL *****
/ ***** DIAGNOSTIC PROGRAM PREAMBLE *****
.LOC 0
DIRT 0
.LOC 40
.LOC 0
+1
HALT
.LOC 45
EGGS
.ZREL
.BLK 8.
EGGS: EGGS
MENTOP: 0
ICAT: 0
/ SPACE RESERVED FOR CAT/NO CAT RESTART ENTRIES
.LOC 170
SUBO 0/0,SKP
ADC 0/0
ESTA 0/CATSW
JMP 0/+1
SPMESS
/ SPACE RESERVED FOR SPECIAL RESTART ENTRIES
.LOC 176
.BLK 2
/ LOCATIONS 200 = 213 RESERVED FOR ECLIPSE TESTS
.LOC 200
JMP
DTOSB: 0
ITRETI: 0
SGNADR: 0
PASSI: 0
PASSVLI: 1
ITRI: 0
ITRETI: 0
ITRETI: 0
ERRETI: 0
LOPRETI: 0
/ TEMPORARY STORAGE FOR MACHINE STATE IN LOOP, ERROR AND INIT SUBROUTINES
/ TEMPORARYS FOR
/ PRINT ROUTINES
ICONSTANT 0

```

```

10016 ECL22
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58

/ ***** LOCAL ZREL *****
/ ***** DIAGNOSTIC PROGRAM PREAMBLE *****
.LOC 0
DIRT 0
.LOC 40
.LOC 0
+1
HALT
.LOC 45
EGGS
.ZREL
.BLK 8.
EGGS: EGGS
MENTOP: 0
ICAT: 0
/ SPACE RESERVED FOR CAT/NO CAT RESTART ENTRIES
.LOC 170
SUBO 0/0,SKP
ADC 0/0
ESTA 0/CATSW
JMP 0/+1
SPMESS
/ SPACE RESERVED FOR SPECIAL RESTART ENTRIES
.LOC 176
.BLK 2
/ LOCATIONS 200 = 213 RESERVED FOR ECLIPSE TESTS
.LOC 200
JMP
DTOSB: 0
ITRETI: 0
SGNADR: 0
PASSI: 0
PASSVLI: 1
ITRI: 0
ITRETI: 0
ITRETI: 0
ERRETI: 0
LOPRETI: 0
/ TEMPORARY STORAGE FOR MACHINE STATE IN LOOP, ERROR AND INIT SUBROUTINES
/ TEMPORARYS FOR
/ PRINT ROUTINES
ICONSTANT 0

```

10017 ECL22

```

01 00272 17777 M1: -1
02 00273 00000 TEM1: 0
03 00274 00000 TEM2: 0
04 00275 00000 TEM3: 0
05 00276 00000 RAN1: 0
06 00277 00000 SHAN1: 0
07 00300 00000 ORIGN1: 0
08 00301 00000 ORIG1: 0
09 00302 00000 ORIG2: 0
10 00303 00000 ORIG3: 0
11 00304 00000 OK1: 0
12 00305 00000 OK2: 0
13 00306 00000 OK3: 0
14 00307 00000 PHULR1: 0
15 00310 00000 PHULR2: 0
16 00311 00000 MURET1: 0
17 00312 00000 FOIVR1: 0
18 00313 00000 BAMSIZ1: 0
19 00314 00000 BAMS11: 0
20 00315 00000 BAMS12: 0
21 00316 00000 FRDN1: 0
22 00317 00000 TDD1: 0
23 00320 00000 BACC1: 0
24 00321 00000 BACC2: 0
25 00322 00000 BACC3: 0
26 00323 00000 BACC3: 0
27 00324 00102 UPI: VCTAB
28 00325 00540 JSR3: JSR 0,3
29 00326 00500 JSR2: JSR 0,2
30 00327 00010 P8K1: P8K 0,

```

10018 ECL22

```

01 00500 00500
02 00501 00524
03 00502 150400
04 00503 150000
05 00504 050011-
06 00505 122470
07 00506 005370
08 00507 101005
09 00510 000415
10 00511 122470
11 00513 005470
12 00514 005400
13 00515 112033
14 00516 000487
15 00517 132400
16 00520 024077-
17 00521 147000
18 00522 344012-
19 00523 000215
20 00524 000487
21 00525 004075-NOCAT1
22 00526 132400
23 00527 120400
24 00530 146470
25 00532 006345
26 00533 050215
27
28
29
30
31
32
33
34

```

IRANDOM #

ISAVED RANDOM #

IVARIABLES IN MUL/DIV TEST  
JORIGINAL #

ICORRECT NUMBER

IBAM TEST VARIABLES

```

LOC 500
*****SIZE SYSTEM & RESERVE MEMORY*****
I *****
NSTR1:
JSR
NEG
COM
STA
ELDA
MOV
JMP
ELDA
LDA
ADD
JMP
SUB
LDA
ADD
STA
JMP
LDA
SUB
SUR
SUB
EST
STA

```

```

#SIZE
2,2
2,2
2,MENTOP
0,CATSW
0,0-SNR
0,CAT
0,PREND
1,1777
1,0
0,2-SNC
NOCAT
1,2
1,400
2,1
1,CAT
2,MAXLOC
SWRESS
1,200
1,2
1,1
1,CATSW
2,MAXLOC

```

```

ICALL SIZE ROUTINE
ISTORE ADDRESS OF LAST...
ILOCATION IN MEMTOP.
I TEST IF OTOS SET CATSW
I IF 0 CAT WAS NOT LOADED,
I IF CAT WAS LOADED
I TEST FOR SUFFICIENT
IMEMORY TO RUN CAT
ITAKE 2K OFF THE TOP
IOF MEMORY FOR CAT
ISET UP CAT START ADDR.
IMAX LOC FOR TEST BUFFERS
ITAKE 200 OFF THE
ITOP OF MEMORY FOR
IMINI-MONITOR
I CLEAR CATSW.
IMAX LOC FOR TEST BUFFERS

```

```

10P19 ECL22
01
02
03 00533 006220 SMES1 JSR @IMSS
04 00534 001755 MESIZ LDA
05 00535 024011- MOVO @,0
06 00536 101040 JSR @IPOCY
07 00537 006216 JSR @IMSS
08 00538 006220 KCRLF
09 00541 001771 SUB
10 00542 120400 STA
11 00543 044203 ELDA
12 00544 120470
13 00547 006327
14 00548 125004
15 00547 000414
16
17 00550 006220 JSR @IMSS
18 00551 001773 BESH
19 00552 006220 JSR @IMSS
20 00553 001771 HALT
21 00554 003077 JMP
22 00555 000401 ESTA
23 00556 000477
24 00557 142470
25 00561 000402 JMP
26

10020 ECL22
01
02 00562 006012- START: JSR @ICAT
03
04
05
06
07
08
09
10
11 00563 102400 BEG1: SUB @,0
12 00564 040210 STA P,ITRER
13
14
15 00565 006234 RAND
16 00566 103300 JSR @ENTRA
17 00567 030215 LDA 2,MAXLOC
18 00570 142422 SUBZ 2,0,5ZC
19 00571 000777 JMP *+1
20 00572 143000 ADD 2,0
21 00573 146422 SUBZ 2,1,5ZC
22 00574 000777 JMP *+1
23 00575 147000 ADD 2,1
24
25 00576 030214 BEG2: LDA 2,MINLOC
26 00577 142433 SUBZ 2,0,SNC
27 00580 000763 JMP BEG1
28 00581 140433 SUBZ 2,1,SNC
29 00582 000761 JMP BEG1
30 00583 106433 SUBZ 0,1,SNC
31 00584 104710 XCH 0,1
32
33 00585 131000 BEG3: MOV 1,2
34 00586 112400 SUB 0,2
35 00587 034075- LDA 3,0377
36 00510 172433 SUBZ 3,2,SNC
37 00511 000752 JMP BEG1
38 00512 040226 STA 0,IBEG
39 00513 044227 STA 1,BEND
40 00514 002401 JMP @,+1
41 00515 002015 BEGIN
42

```

```

I *****OUTPUT STRT MESSAGE & READ SWITCHES*****
I PRINT SIZE OF MEMORY
I INIT PASS COUNT
I PRUNING IN AUTO MODE?
I YES START PROGRAM,
I NO, PRINT SET SWITCHES MESS,
I READ NEW STATE OF SWITCHES
I START

```

```

*****INITIALIZING ROUTINES*****
I THESE ROUTINES GENERATE APPROPRIATELY SIZED RANDOM BUFFERS
I FOR USE BY THE TEST PROGRAMS.
I START OF PROGRAM
I MAKE C(AC0=1) MODULO
I MEMORY SIZE
I BUFFER ADDRESS MUST BE
I GREATER THAN END OF
I PROGRAM
I C(AC0) < C(AC1)
I C(AC2) = SIZE OF BUFFER
I BUFFER SIZE MUST BE
I 400 OF GREATER
I BEGIN OF BUFFER
I END OF BUFFER

```

```

10021 ECL22
01
02
03
04 00616 023401 TRP: LDA 0,0,1,3
05 00617 040040 STA 0,0,SP
06 00620 040041 STA 0,0,FP
07 00621 020227 LDA 0,0,END
08 00622 020074- LDA 1,1,140
09 00623 122400 SUR 1,0
10 00624 040042 STA 0,SL
11 00625 021400 LDA 0,0,3
12 00626 040043 STA 0,0,3
13 00627 040045 STA 0,0,3
14 00630 033401 LDA 2,0,1,3
15 00631 024073- LDA 1,1,0
16 00632 133000 ADD 1,2
17 00633 050044 STA 2,1,0
18 00634 024072- LDA 1,1,32,
19 00635 041000 STA 0,0,2
20 00636 151400 INC 2,2
21 00637 125400 INC 1,1,1,5Z
22 00640 000775 JMP "3
23 00641 025402 LDA 1,2,3
24 00642 133000 ADD 1,2
25 00643 025403 LDA 1,3,3
26 00644 045340 STA 1,-32,,2
27 00645 001404 JMP 4,3
28
29
30
31
32 00646 020226 STK: LDA 0,0,BEG
33 00647 100110 SBI 1,0
34 00650 040040 STA 0,0,SP
35 00651 040041 STA 0,0,FP
36 00652 040004 STA 0,0,ISP
37 00653 020227 LDA 0,0,END
38 00654 024071- LDA 1,1,100
39 00655 122400 SUR 1,0
40 00656 040042 STA 0,SL
41 00657 040006 STA 0,1,SL
42 00658 021400 LDA 0,0,3
43 00661 040045 STA 0,0,FP
44 00662 040043 STA 0,0,3F
45 00663 040007 STA 0,0,3F
46 00664 001401 JMP 1,3

```

```

10022 ECL22
01
02
03
04 00665 054420 VEC: STA 3,VRET
05 00666 033400 LDA 2,0,3
06 00667 021421 LDA 0,1,3
07 00670 041000 STA 0,0,2
08 00671 155400 INC 2,3
09 00672 024071- LDA 1,1,100
10 00673 133710 SLM
11 00674 034411 LDA 3,VRET
12 00675 021402 LDA 0,2,3
13 00676 025403 LDA 1,3,3
14 00677 133000 ADD 1,2
15 00700 041300 STA 0,-100,2
16 00701 062677 IDRSY
17 00702 102000 ORC 0,0
18 00703 062077 MSKO 0
19 00704 001404 JMP 4,3
20 00705 000000 VRET: 0

```

```

/ TRAP# FAULT, ORIGIN, TRAP#, SUBROUTINE ADDRESS
/ SET STACK POINTER
/ AND FRAME POINTER
/ END OF BUFFER=100
/ IS THE STACK LIMIT
/ STACK FAULT AND ADDRESS
/ INIT FAULT AND ADDRESS
/ FLOATING FAULT
/ TRAP ORIGIN ADDRESS
/ SET ORIGIN
/ PUT FAULT RETURN
/ IN THE DISPATCH TABLE
/ TRAP NUMBER.....
/ SUBROUTINE ADDRESS
/ SET A SINGLE SUBROUTINE ADDRESS AND EXIT

```

```

/ VECTOR TABLE INITIALIZER
/ SETUP A VECTOR TABLE.....
/ CALL+1=ORIGIN POINTER
/ CALL+2=ERROR ADDRESS
/ CALL+3=CORRECT ENTRY
/ CALL+4=ENTRY NUMBER
/ TABLE IS FILLED WITH ERR
/ RETURNS...
/ GOOD ENTRY
/ POSITION (DEVICE CODE).
/ THE GOOD ENTRY.
/ MASK OTHER DEVICES.

```

```

/ STACK INITIALIZATION ROUTINE
/ INITIALIZE A STACK.....
/ MAKE STACK POINTER
/ AND FRAME POINTER
/ POINT TO THE DATA
/ BUFFER =1.
/ BUFFER END -100 IS
/ THE STACK LIMIT
/ SETUP STACK FAULT
/ AND FLOATING FAULT

```

```

/ VECTOR TABLE INITIALIZER
/ SETUP A VECTOR TABLE.....
/ CALL+1=ORIGIN POINTER
/ CALL+2=ERROR ADDRESS
/ CALL+3=CORRECT ENTRY
/ CALL+4=ENTRY NUMBER
/ TABLE IS FILLED WITH ERR
/ RETURNS...
/ GOOD ENTRY
/ POSITION (DEVICE CODE).
/ THE GOOD ENTRY.
/ MASK OTHER DEVICES.

```

10023 ECL22

01 /  
02 / \*\*\*\*\*TEST UTILITY SUBROUTINES\*\*\*\*\*  
03 /

04 / SUBROUTINE TO INITIALIZE A TEST LOOP

05 INC 3,3 /TEST LOOP INITIALIZER  
06 STA 3,ITRER /SAVE RETURN LOCATION  
07 STA 0,AC0 /SAVE CONTENTS OF ACC  
08 LDA 0,-1,3 /SET # OF ITERATIONS  
09 STA 0,ITR /SET ITER. VALUE  
10 STA 0,ITRCT /SET ITER. COUNT

11 SUB 3,3 /CLEAR ERROR SWITCH  
12 STA 3,ITRER /CLEAR ERROR COUNT  
13 LDA 3,PASS /TEST FOR FIRST PASS  
14 MOV 3,3,SRZ  
15 JMP INI11

16 SUBZL 3,3 /THIS IS 1ST PASS  
17 STA 3,ITR /SET ITERATIONS FOR  
18 STA 3,ITRCT /1 LOOP ONLY.

19 LDA 0,AC0 /RESTORE AC'S AND  
20 JMP 0,ITRET /EXIT TO TEST

10024 ECL22

01 /  
02 / SUBROUTINE TO TERMINATE A TEST LOOP  
03 /

04 STA 3,LOPRET /END OF TEST ROUTINE  
05 ITRCT LOP3  
06 JMP LOP3 /ITERATION COMPLETE  
07 LDA 3,ITRER  
08 MOV 3,3,SRR  
09 JMP 0,LOPRET /WHEN NO ERROR, EXIT TO NEXT  
10 STA 3,ITR /PRESET ITERATION COUNTER  
11 STA 3,ITRCT  
12  
13 READS 3 /LOOK AT SWITCH #0  
14 MOVLM 3,3,STC /TO SEE WHETHER #0  
15 JMP +3 /FOR DATA SWITCHES USED,  
16 STA 3,SRREG /USE SRREG  
17  
18 ADDL 3,3 /IF SWITCH #3 PRINT X  
19 ADDL 3,3,SNL /NO X PRINT OUT REQUIRED  
20 JMP LOP2  
21 STA 0,AC0  
22 STA 1,AC1  
23 STA 2,AC2  
24 JBR 0,INSS /SAVE AC'S  
25 PERCENT /MESSAGE \*12\*15\* X FAIL\*  
26 SUB 0,0  
27 LDA 1,ITREC /GET ERROR COUNT  
28 STA 0,ITREC /CLEAR ERROR COUNT.  
29 LDA 2,-100,  
30 MUL  
31 LDA 2,ITR /COUNT X 100 / ITERATIONS=  
32 DIV /PERCENTAGE OF FAILURE  
33 JSR 0,IPDEC /DECIMAL PRINTER  
34 LDA 0,AC0  
35 LDA 1,AC1  
36 LDA 2,AC2  
37 SUB 3,3  
38 STA 3,ITREC

39  
40 LDA 3,ITRER /IF NO ERROR, ITERATE  
41 MOV 3,3,SRZ /OTHERWISE LOOK AT DATA  
42 READS 3 /"1" SWITCH FOR PROCEED,  
43 MOVLM 3,3,STC /FOR NOT.  
44 JMP +3  
45 ELDA 3,SRREG  
46 ADDLM 3,3,SNL  
47 JMP 0,ITRET  
48 STA 0,ITRET  
49 JMP 0,LOPRET

10025 ECL22

```

01 / LONG AND SHORT FORM ERROR ROUTINES
02
03
04 01003 054252  ERR: STA 3,EPC
05 01004 040245  STA 0,AC0
06 01005 102560  SUBCL 0,0
07 01006 040251  STA 0,CRY
08
09 01007 021400  LDA 0,0,3
10 01010 175400  INC 3,3
11 01011 101004  MOV 0,0,5ZR
12 01012 000775  JMP 0,3
13 01013 200405  JMP ERR
14
15 01014 054252  ERR: STA 3,EPC
16 01015 040245  STA 0,AC0
17 01016 100560  SUBCL 0,0
18 01017 040251  STA 0,CRY
19
20 01020 054212  ERR: STA 3,ERRET
21 01021 010211  ISZ ITREC
22
23 01022 020210  LDA 0,ITRER
24 01023 110033  ADGZ# 0,3,SNC
25 01024 000464  JMP ERR4
26
27 01025 054210  STA 3,ITRER
28 01026 044246  STA 1,AC1
29 01027 050247  STA 2,AC2
30 01030 000220  JSR #INSS
31 01031 001337  ERMSG
32 01032 024203  LDA 1,PASS
33 01033 125420  INCZ 1,1
34 01034 000217  JSR #IPDEC
35 01035 000220  JSR #INSS
36 01036 001733  HEADER
37 01037 101020  MOVZ 0,0
38 01040 024251  LDA #IPDEC
39 01041 000217  JSR #IPDEC
40 01042 101040  MOV0 0,0
41 01043 024245  LDA 1,AC0
42 01044 000210  JSR #IPOCT
43 01045 024246  LDA 1,AC1
44 01046 000210  JSR #IPOCT
45 01047 024247  LDA 1,AC2
46 01050 000210  JSR #IPOCT
47 01051 024250  LDA 1,AC3
48 01052 000210  JSR #IPOCT
49 01053 024252  LDA 1,EPC
50 01054 000210  JSR #IPOCT
51

```

10026 ECL22

```

01 / ERROR ROUTINES (CONTINUED)
02
03 01055 030252  ERR:1 LDA 2,EPC
04 01056 010252  ISZ EPC
05 01057 020212  LDA 0,ERRET
06 01060 140227  ADGZ 2,0,ISBN
07 01061 000413  JMP ERR3
08
09 01062 033000  LDA 3,0,2
10 01063 175113  MOVLM 3,3,SNC
11 01064 000405  JMP ERR2
12 01065 054402  STA 3,0,2
13 01066 000220  JSR #INSS
14 01067 000000  0
15 01070 000705  JMP ERH1
16
17 01071 025400  ERR:2 LDA 1,0,3
18 01072 000216  JSR #IPOCT
19 01073 000702  JMP ERR1
20
21 01074 000220  ERR:3 JSR #INSS
22 01075 001771  KCRLF
23 01076 024246  LDA 1,AC1
24 01077 030247  LDA 2,AC2
25 01100 136470  ELDA 3,AUTO
26
27 01102 175005  MOV 3,3,SNR
28 01103 000405  JMP ERR4
29
30 01104 002077  IORST
31 01105 034010  LDA 3,EGDS
32 01106 035404  LDA 3,0,3
33 01107 001400  JMP 0,3
34
35 01110 020245  ERR:4 LDA 0,ACR
36 01111 034250  LDA 3,AC3
37 01112 002212  JMP #ERRET

```

FARE WE IN LONG OR SHORT MODE FOR LIST IS DONE.

ILCAD ARGUMENT FROM LIST MESSAGE PRINT? INO, USE OCTAL PRINT

ISSETUP FOR MESSAGE PRINT

ILCAD VALUE TO PRINT

RESTORE AC'S TEST, AUTO MODE?

INNOPE, BACK TO TEST

RETURN TO OTDS

RETURN TO TEST

10027 ECL22

01 / RANDOM NUMBER GENERATOR SUBROUTINES

```

02 /
03 /
04 /
05 01113 000276 ,BRAN: LDA 0,RAN
06 01114 000405 JAP ,RAND*4
07 01115 000276 ,RAND: LDA 0,RAN
08 01116 024210 LDA 1,ITRER
09 01117 100004 MOV 1,1,ISZR
10 01120 001400 JMP 0,3
11 01121 100000 MOV 0,1
12 01122 100410 HXL 2,1
13 01123 100000 ADD 0,1
14 01124 100120 MOVZL 1,1
15 01125 100120 MOVZL 1,1
16 01126 100000 ADD 1,0
17 01127 024067 ,LDA 1,0,03031
18 01130 100000 ADD 1,0
19 01131 040276 STA 0,RAN
20 01132 001400 JMP 0,3
21 /
22 /
23 01133 054303 ,RANS: STA 3,ORIG3
24 RAN: RAN: STA 3,ORIG3
25 JSR 0,ENTRA
26 INCS 0,1
27 NEG 0,2
28 ADD 1,2
29 STA 0,ORIG0
30 STA 1,ORIG1
31 STA 2,ORIG2
32 JMP 0,ORIG3
33 /
34 /
35 /
36 /
37 01144 000214 SIZE: LDA 2,HEINLOC
38 01145 101400 INC 2,2
39 01146 101112 MOVL# 2,2,ISZC
40 01147 000406 JHP ,*6
41 01150 021000 LDA 0,0,2
42 01151 051000 STA 2,0,2
43 01152 025000 LDA 1,0,2
44 01153 041000 STA 0,0,2
45 01154 100414 SUB# 1,2,ISZR
46 01155 001400 STA 0,3
47 01156 000111 ,JMP 2,HEMTP0
48 01157 000766 JMP SIZE*1

```

1 SIZEING SUBROUTINE1 RETURNS SIZE OF LOGICAL MEM IN ACC.

```

39 /
40 /
41 /
42 /
43 /
44 /
45 /
46 /
47 /
48 /
49 /
50 /
51 /
52 /
53 /
54 /
55 /
56 /
57 /
58 /
59 /
60 /

```

10028 ECL22

01 /\*\*\*\*\*PRINT ROUTINES\*\*\*\*\*/

```

02 /
03 /
04 01160 175100 PDEC: MOVL 3,3
05 01161 054255 STA 3,PODERET /DECIMAL PRINT C(AC1).
06 01162 175200 MOVR 3,3
07 01163 004441 JSR PDEC3 /PRESET C(CARRY) FOR ZERO SUPPRESSION
08 01164 003420 10000. /RESET C(CARRY) IF NOT
09 01165 001750 1000.
10 01166 000144 10.
11 01167 000012 1.
12 01170 000001 0.
13 01171 000000 MOVL 3,3
14 01172 175100 POCT: STA 3,PODERET /OCTAL PRINT C(AC1)
15 01173 054255 MOVR 3,3
16 01174 175200 JSR PDEC3 /PRESET C(CARRY) FOR ZERO SUPPRESSION
17 01175 004427 100000. /RESET C(CARRY) IF NOT
18 01176 100000 10000.
19 01177 010000 1000.
20 01200 001000 100.
21 01201 000100 10.
22 01202 000010 1.
23 01203 000001 0.
24 01204 000000 /
25 /
26 01205 000065 ,PDEC1: MOVL 0,0,11
27 01206 051377 LDA 2,1,2
28 01207 101015 MOVR 2,2,SNR
29 01210 000415 JAP PDEC3*1
30 01211 102450 SUBC 0,0
31 01212 146452 SUBO# 2,1,ISZC
32 01213 000404 JHP PDEC2
33 01214 146420 SUBZ 2,1
34 01215 101400 INC 0,0
35 01216 000774 JHP ,*4
36 01217 101234 PDEC2: MOVR# 2,2,ISZ
37 01220 102452 SUBC 2,2,ISZ
38 01221 000005 ,LDA 2,0,0
39 01222 143004 INC 2,0,ISZR
40 01223 171481 INC 3,2,SKP
41 01224 171481 PDEC3: JSR CHAR
42 01225 004423 MOVR 2,3,ISZR
43 01226 100004 JHP PDEC1 /PRINT,***
44 01227 000756 LDA 3,PODERET /SKIP IF TAB EXIT
45 01230 034255 MOVR 3,3 /NEXT DIGIT
46 01231 175200 JHP ,*3 /EXIT
47 01232 001400 JHP 0,3 /MESSAGE PRINTER
48 /
49 01233 175400 MESS: INC 3,3
50 01234 054256 STA 3,MESSRET /JSR (MESS)
51 01235 031777 LDA 2,1,3 /MESSAGE ADDRESS
52 01236 020075 ,MESS1: LDA 0,3,377
53 01237 025000 LDA 1,0,2
54 01240 101420 INCZ 2,2
55 01241 103400 AND 1,0
56 01242 100700 SUBS 0,1
57 01243 004405 JSR CHAR
58 01244 121005 MOVR 1,0,SNR
59 01245 002256 JHP MESSRET
60 01246 004402 JSR CHAR

```

IPRECESS WITH TAB

IFEXIT, ALL DIGITS PRINTED

IFORM THE DIGIT  
IAND SET C(CARRY)

IFSKIP IF LAST DIGIT  
IFSKIP IF ZERO SUPPRESS  
IFMAKE COUNT INTO ASCII

IFPRINT,\*\*\*  
IFSKIP IF TAB EXIT  
IFNEXT DIGIT  
IFEXIT

IFMESSAGE PRINTER  
IFJSR (MESS)  
IFMESSAGE ADDRESS



0029 ECL22  
 01 01247 000767  
 02

JMP MESS1

```

10030 ECL22
01
02
03 01250 175100 CHAR1 MOVIL 3,3
04 01251 054253 STA 3,CHARET
05 01252 050460 STA 2,CHRSV
06
07 01253 074477 READS 3
08 01254 175112 MOVILW 3,3,3ZC
09 01255 000403 JMP *3
10 01256 136470 ELDA 3,3,REG
11
12 01260 030064 LDA 2,*22000
13 01261 123400 AND 3,2
14 01262 159120 ADDZL 2,2
15 01263 153285 ADDCR 2,2,SNR
16 01264 000435 JMP REST
17
18 01265 034075 LDA 3,*377
19 01266 117725 ANDZS 0,3,SNR
20 01267 000432 JMP REST
21
22 01270 103004 ADD 0,0,SR
23 01271 000777 JMP *1
24 01272 172260 ADDCR 3,3
25 01273 020053 LDA 0,*211*400
26 01274 162445 SUBG 3,0,SNR
27 01275 000430 JMP CHAR4
28
29 01276 161340 MOVDS 3,0
30 01277 010254 CHAR1 ISZ CHORZ
31
32 01300 151133 MOVZLW 2,2,SNR
33 01301 000405 JMP CHAR2
34 01302 061117 DOAS 0,LPT
35 01303 065517 SKPBZ LPT
36 01304 000777 JMP *1
37 01305 060217 NI0C LPT
38
39 01306 151133 CHAR2 MOVZLW 2,2,SNR
40 01307 000405 JMP CHAR3
41 01310 061111 DOAS 0,TTO
42 01311 063511 SKPBZ TTD
43 01312 000777 JMP *1
44 01313 060211 NI0C TTD
45
46 01314 175403 CHAR3 INC 3,3,SNR
47 01315 000762 JMP CHAR1
48 01316 030062 LDA 2,*212
49 01317 142405 SUB 2,0,SNR
50 01320 040254 STA 0,CHORZ
51
52 01321 030411 REST LDA 2,CHRSV
53 01322 054253 LDA 3,CHARET
54 01323 175200 MOVR 3,3
55 01324 001400 JMP *3
56
57 01325 034254 CHAR4 LDA 3,CHORZ
58 01326 020061 LDA 0,*08
59 01327 114410 TOR 0,3
60 01330 020060 LDA 0,*240
  
```

1 LPT/TTO INTERFACE ROUTINE: CHARACTER PASSED IN ACB

```

!SAVE RETURN ADDR.
!SAVE AC2
!READ SWITCHES INTO ACS
!TEST SWITCH 0
! SW0 SET
! SW0 CLEAR, DEFAULT
!MASK SW2 & SW5 INTO
!FAC2 FROM ACS
!LEFT JUSTIFY SW2
!COMPLEMENT SW2
!NO OUTPUT, RETURN
!MASK CHARACTER INTO L-BYTE
!OF ACS, CLEAR CARRY.
!IF NULL CHAR. RETURN.
!DETERMINE REQUIRED
!STATE OF PARITY BIT &
!INSERT IT
!TEST FOR TAB
!TRUE: SETUP TAB SIMULATION
!RESTORE CHARACTER TO R-BYTE ACB
!SET CARRY, BUMP LINE COUNT.
!SEND TO LPT
!NOPE, MUST BE TTO
!O.K. FETCH CHARACTER
!WAIT FOR DONE
!CLEAR DEVICE
!SEND TO TTO
!O.K. SEND CHARACTER
!WAIT FOR DONE
!CLEAR DEVICE
!IF TABBING, AND NOT
!FINISHED, LOOP.
!TEST FOR CR/LF
!TRUE: ZERO LINE COUNT
!OTHERWISE RETURN
!SET UP TO TAB
!IC(ACS) IS TWOIS COMPLEMENT
!OF # OF SPACES NEEDED
  
```

0031 ECL22  
01 01331 000746  
02  
03 01332 000000 CHRSV: 0

CHAR1

JMP

ITEMP SAVE FOR AC2

0032 ECL22

01  
02 01333 005215 PASHES: .TXTE I<15><12>PASS I  
03 040320  
04 051523  
05 000240  
06 01337 005215 ERMSG: .TXTE I<15><12>ERROR IN PASS I  
07 151305  
08 147722  
09 120322  
10 047311  
11 050240  
12 051301  
13 035123  
14 000240  
15 01350 005215 BTOH1: .TXTE I<15><12>  
16 152102 BTO ADDR GOOD BAD WORD <15><12> I  
17 004717  
18 042101  
19 151104  
20 043411  
21 147717  
22 004504  
23 040502  
24 004504  
25 147727  
26 042322  
27 106640  
28 000012  
29 01366 005215 BITM1: .TXTE I<15><12>  
30 152102 BITZ ADDR GOOD BAD WORD <15><12> I  
31 004532  
32 042101  
33 151104  
34 043411  
35 147717  
36 004504  
37 040502  
38 004504  
39 147727  
40 042322  
41 106640  
42 000012  
43 01404 005215 BAHH1: .TXTE I<15><12>ACIS AFTER BAH INSTRUCTION WRONG  
44 141501  
45 051447  
46 040640  
47 152306  
48 151305  
49 041240  
50 040501  
51 140640  
52 051516  
53 151324  
54 141025  
55 144724  
56 047317  
57 153640  
58 147722  
59 043515  
60 01425 005215 <15><12>ORIG C/C -3) I

|  |  |
|--|--|
| 0033 ECL22   | 0034 ECL22   |
| 01 151317  | 01 040411  |
| 02 043711  | 02 150753  |
| 03 141640  | 03 040411  |
| 04 040450  | 04 151303  |
| 05 030303  | 05 005215  |
| 06 031455  | 06 01527 151317 ORIG I   |
| 07 120251  | 07 000000  |
| 08 000000  | 08 000011  |
| 09 01436 005215 BAMH2: .TXTE I<15><12>GOOD C(AC0=3) I                              | 09 01532 005215 BAMH7: .TXTE I<15><12>SHOULD BE ZERO, C(AC0=3) I |
| 10 147507  | 10 044123  |
| 11 045317  | 11 052717  |
| 12 141640  | 12 043314  |
| 13 040450  | 13 041240  |
| 14 030303  | 14 120305  |
| 15 031455  | 15 142532  |
| 16 120251  | 16 147722  |
| 17 000000  | 17 120254  |
| 18 01447 005215 BAMH3: .TXTE I<15><12>BAD C(AC0=3) I                               | 18 043303  |
| 19 040502  | 19 141501  |
| 20 120104  | 20 026400  |
| 21 024303  | 21 124463  |
| 22 141581  | 22 120275  |
| 23 026400  | 23 000000  |
| 24 124463  | 24 01551 005215 BAMH8: .TXTE I<15><12> BAD                       |
| 25 000240  | 25 01592 021101 ADDR GOOD  |
| 26 01457 005215 BAMH4: .TXTE I<15><12>BAM" SET C(CARRY) I                          | 26 151104  |
| 27 041042  | 27 043411  |
| 28 046501  | 28 147717  |
| 29 120042  | 29 004504  |
| 30 142523  | 30 040502  |
| 31 120324  | 31 004504  |
| 32 024303  | 32 040502  |
| 33 040703  | 33 004515  |
| 34 151322  | 34 040700  |
| 35 124531  | 35 146311  |
| 36 000000  | 36 042305  |
| 37 01472 005215 BAMH5: .TXTE I<15><12> FROM TO WORD ORIG C(AC0) GOOD BAD<15><12> I | 37 196640  |
| 38 01473 151306  | 38 000012  |
| 39 046717  | 39 01572 005215 MULH1: .TXTE I<15><12> ACP AC1 AC2<15><12>       |
| 40 152011  | 40 01571 052515 MUL ACP AC1                                      |
| 41 004717  | 41 004714  |
| 42 147727  | 42 141501  |
| 43 042322  | 43 004400  |
| 44 147411  | 44 141501  |
| 45 144722  | 45 004401  |
| 46 004507  | 46 141501  |
| 47 024303  | 47 106662  |
| 48 141501  | 48 01601 147412 ORIG I   |
| 49 124400  | 49 144722  |
| 50 043411  | 50 004507  |
| 51 147717  | 51 000000  |
| 52 004504  | 52 01605 005215 MULH2: .TXTE I<15><12>BAD I                      |
| 53 040502  | 53 040502  |
| 54 106504  | 54 004504  |
| 55 000012  | 55 000000  |
| 56 01515 005215 EDIVH: .TXTE I<15><12> AC2<15><12>                                 | 56 01611 005215 MULH3: .TXTE I<15><12>GOOD I                     |
| 57 01516 144504 DIVX AC0 AC1   | 57 147507  |
| 58 154126  | 58 042317  |
| 59 040411  | 59 000011  |
| 60 050303  | 60 01615 005215 DIVH1: .TXTE I<15><12>DIV ACP AC1 AC2<15><12>    |

```

0035 ECL22
01 144504
02 044526
03 141501
04 044460
05 141501
06 044601
07 141501
08 141501
09 01626 147412 ORIG I
10 144722
11 044507
12 044600
13 01632 045215 MOD4: .TXTE I<15><12>DIV/MUL AC0 AC1 AC2
14 144504
15 127826
16 045215
17 044714
18 141501
19 044460
20 141501
21 044661
22 141501
23 01644 046662 <15><12>ORIG I
24 147412
25 144722
26 044507
27 044600
28 01651 045215 SDVH1: .TXTE I<15><12>SDIV AC0 AC1 AC2
29 042123
30 053311
31 040411
32 038303
33 043411
34 130703
35 040411
36 131303
37 01662 045215 <15><12>ORIG I
38 151317
39 043711
40 040011
41 01666 045215 SHH1: .TXTE I<15><12>SHUL AC0 AC1 AC2
42 048523
43 148125
44 040411
45 030503
46 040411
47 130703
48 040411
49 131303
50 01677 045215 <15><12>ORIG I
51 151317
52 043711
53 040011
54 01703 045215 BLMH5: .TXTE I<15><12> BLM FAILED <15><12>I
55 01704 151306 FROM TO WORD 6000 BAD
56 048717
57 152011
58 040411
59 147707
60 042322

0036 ECL22
01 043411
02 147717
03 044524
04 044504
05 044504
06 146102
07 146115
08 040700
09 146311
10 042305
11 146840
12 040012
13 01726 045215 PERCENT: .TXTE I<15><12>X FAIL=1
14 146315
15 040700
16 146311
17 040275
18 01733 045215 HEADER: .TXTE I<15><12><15><12>
19 045215
20 01735 151303 CRY AC0 AC1 AC2 AC3 LISTING <15><12>I
21 044531
22 141501
23 044460
24 141501
25 044661
26 141501
27 044602
28 141501
29 044483
30 144714
31 152103
32 047311
33 044507
34 045215
35 040000
36 01755 045215 MESIZ: .TXTE I<15><12>LAST LOGICAL ADDRESS=1
37 040714
38 152123
39 146240
40 043717
41 141711
42 146101
43 040640
44 042104
45 142722
46 041523
47 040275
48 01771 045215 KCRLF: .TXTE I<15><12>I
49 040000
50 01773 142523 SETSN: .TXTE ISET DATA SWITCHS AND PRESS CONTINUE!
51 120324
52 040504
53 040724
54 041640
55 144727
56 141724
57 041510
58 040640
59 042116
60 040240

```

0037 ECL22  
 01 142722  
 02 051323  
 03 141649  
 04 047317  
 05 144724  
 06 052516  
 07 000305

10038 ECL22

```

01 BEGIN:
02 J *****FIRST TEST*****
03
04
05
06
07
08
09
10 TRAP: SETUP 100 ;TEST "CAP"
11 ;INITIALIZE TEST....
12 TRAPER TR0C,BBEG,0,TR0A
13 ;IF STACK FAULT, GO TO TR0C
14 ;TRAP ORIGIN IN CONTENTS OF BBEG
15 ;THE CAP NUMBER IS 0
16 ;SUBROUTINE ADDRESS IS TR0A
17 ;STACK POINTERS SET TO C(BBEG)
18 LDA 0,SP
19 XOP 3,3,0
20 ;WILL THE STACK POINTER
21 ;BE UPDATED BY 7?
22 LDA 2,15
23 ADD 0,2
24 SUB# 1,2,512
25 ERROR 1,2,SKP
26 MOV 0,0,SKP
27 ERROR
28 LOOP
29
30 TR11: JSR #ENTLO
31 ;ITERATE TEST ROUTINE....
32
33
34 TR11: SETUP 100 ;TEST CAP
35 ;INITIALIZE TEST....
36 TRAPER TR1C,BBEG,1,TR1A
37 ;IF STACK FAULT, GO TO TR1C
38 ;TRAP ORIGIN IN CONTENTS OF BBEG
39 ;THE CAP NUMBER IS 1
40 ;SUBROUTINE ADDRESS IS TR1A
41 ;STACK POINTERS SET TO C(BBEG)
42
43 JSP #ENTRA
44 INCZ 0,1,5ZC
45 ;MAKE C(AC1) ONE MORE
46 ;THAN C(AC0) CHECK FOR
47 ;PROPER ACP-1 VALUE
48 XOP 3,3,1
49 ;AFTER THE TRAP,
50 ;C(CARRY) SHOULD BE (0).
51
52 TR1A: ADC# 0,1,5NR
53 MOV 0,0,5ZC
54 ERROR
55 MOV 0,0,SKP
56
57 TR1B: MOV 0,0,SKP
58 ERROR
59 LOOP
60
61 TR1C: JSP #ENTLO
62 ;DISPATCH OR STACK FAULT.
63 ;ITERATE TEST ROUTINE....
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

10039 ECL22

```

01
02
03
04 02073 006230
05 02074 006100
06
07 02075 006222
08 02076 002115
09 02077 000226
10 02100 000003
11 02101 002106
12
13 02102 002026
14 02103 024056
15 02104 107000
16 02105 174330
17 02106 156415
18 02107 132414
19
20 02114 101001
21
22
23 02121 006231
24
25
26 02122 006230
27 02123 006100
28
29 02124 006222
30 02125 002144
31 02126 000226
32 02127 000004
33 02130 002135
34
35 02131 000226
36 02132 024055
37 02133 107000
38 02134 150430
39 02135 156415
40 02136 132414
41
42 02143 101001
43
44
45 02150 006231

```

```

TR21
SETUP 100
JSR #ENTIN
LOOP
TRAPER TR20,BBEG,3,TR2A+4
JSR #ITRP
TR2B
IF STACK FAULT, GO TO TR2B
TRAP ORIGIN IN CONTENTS OF BBEG
BBEG
THE CAP NUMBER IS 3
SUBROUTINE ADDRESS IS TR2A+4
STACK POINTERS SET TO C(BBEG)
TR2A+4
TRTST 3,3
LOA 0,BBEG
TEST FOR ACS/ACD ADDRESS
STORED IN C(AC2-3),C(AC1)
ADD 0,1
XOP 3,3,3
SUB# 2,3,SNR
SUB# 1,2,SRZ
ERROR
MOV 0,0,SKP
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE....

TR2B1
SETUP 100
JSR #ENTIN
LOOP
TRAPER TR3B,BBEG,4,TR3A+4
JSR #ITRP
TR3B
IF STACK FAULT, GO TO TR3B
TRAP ORIGIN IN CONTENTS OF BBEG
BBEG
THE CAP NUMBER IS 4
SUBROUTINE ADDRESS IS TR3A+4
STACK POINTERS SET TO C(BBEG)
TR3A+4
TRTST 0,4
LOA 0,BBEG
TEST FOR ACS/ACD ADDRESS
STORED IN C(AC2-3),C(AC1)
ADD 0,1
XOP 0,2,4
SUB# 2,3,SNR
SUB# 1,2,SRZ
ERROR
MOV 0,0,SKP
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE....

TR31
SETUP 100
JSR #ENTIN
LOOP
TRAPER TR3B,BBEG,4,TR3A+4
JSR #ITRP
TR3B
IF STACK FAULT, GO TO TR3B
TRAP ORIGIN IN CONTENTS OF BBEG
BBEG
THE CAP NUMBER IS 4
SUBROUTINE ADDRESS IS TR3A+4
STACK POINTERS SET TO C(BBEG)
TR3A+4
TRTST 0,4
LOA 0,BBEG
TEST FOR ACS/ACD ADDRESS
STORED IN C(AC2-3),C(AC1)
ADD 0,1
XOP 0,2,4
SUB# 2,3,SNR
SUB# 1,2,SRZ
ERROR
MOV 0,0,SKP
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE....

```

10040 ECL22

```

01
02
03 02151 006230
04 02152 006100
05
06 02153 006222
07 02154 002173
08 02155 000226
09 02156 000005
10 02157 002104
11
12 02158 000226
13 02159 024054
14 02162 107000
15 02163 124530
16 02164 156415
17 02165 132414
18
19 02172 101001
20
21
22 02177 006231
23
24
25 02200 006230
26 02201 006100
27
28 02202 006222
29 02203 002222
30 02204 000226
31 02205 000006
32 02206 002213
33
34 02207 000226
35 02210 024053
36 02211 107000
37 02212 100030
38 02213 156415
39 02214 132414
40
41
42 02221 101001
43
44
45 02226 006231

```

```

TR41
SETUP 100
JSR #ENTIN
LOOP
TRAPER TR4B,BBEG,5,TR4A+4
JSR #ITRP
TR4B
IF STACK FAULT, GO TO TR4B
TRAP ORIGIN IN CONTENTS OF BBEG
BBEG
THE CAP NUMBER IS 5
SUBROUTINE ADDRESS IS TR4A+4
STACK POINTERS SET TO C(BBEG)
TR4A+4
TRTST 1,5
LOA 0,BBEG
TEST FOR ACS/ACD ADDRESS
STORED IN C(AC2-3),C(AC1)
ADD 0,1
XOP 1,1,5
SUB# 2,3,SNR
SUB# 1,2,SRZ
ERROR
MOV 0,0,SKP
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE....

TR4B1
SETUP 100
JSR #ENTIN
LOOP
TRAPER TR5B,BBEG,6,TR5A+4
JSR #ITRP
TR5B
IF STACK FAULT, GO TO TR5B
TRAP ORIGIN IN CONTENTS OF BBEG
BBEG
THE CAP NUMBER IS 6
SUBROUTINE ADDRESS IS TR5A+4
STACK POINTERS SET TO C(BBEG)
TR5A+4
TRTST 0,6
LOA 0,BBEG
TEST FOR ACS/ACD ADDRESS
STORED IN C(AC2-3),C(AC1)
ADD 0,1
XOP 0,0,6
SUB# 2,3,SNR
SUB# 1,2,SRZ
ERROR
MOV 0,0,SKP
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE....

```

```

10041 ECL22
01
02
03 02227 00230 TR6:
04 02230 000100
05
06 02231 00222 TRAPR TR6A*,BBEG,7,TR6A
07 02232 00225 JSR #ITRP
08 02233 00226 TR6A+1
09 02234 00207 TR7A
10 02235 00224 JSR #ITRP
11 02236 101020 TR7A
12 02237 100730 MOV 0,0
13 ERROR 0,0,7
14 02244 101002 TR6A: MOV 0,0,SZC
15 ERROR
16 LOOP
17 JSR #ENTLO
18
19
20
21 02252 00230 TR7:
22 02253 000100
23
24 02254 00222 TRAPR TR7A,BBEG,10,TR7B
25 02255 00223 TR7A
26 02256 00226 JSR #ITRP
27 02257 00010 TR7A
28 02260 00227 TR7B
29 02261 101040 MOV 0,0
30 02262 100030 TR7A:
31 02263 101003 TR7B:
32 02267 101002 MOV 0,0,SZC
33 02270 150014 ADC# 2,3,SZR
34 ERROR
35 LOOP
36
37 02275 00231

```

```

TEST "CAP"
INITIALIZE TEST....
IF STACK FAULT, GO TO TR6A+1
JTRAP ORIGIN IN CONTENTS OF BBEG
JTHE CAP NUMBER IS 7
JSUBROUTINE ADDRESS IS TR6A
JSTACK POINTERS SET TO C(BBEG)
IFAIL TO DISPATCH AT ALL....
JIF C(CARRY) IS 1, TRAP CHANGED.
JC(CARRY), OTHERWISE DISPATCH
FOR STACK FAULT.
ITERATE TEST ROUTINE....

```

```

TR6:
01 02276 00230
02 02277 000100
03
04 02300 00222
05 02301 002307 TR8A
06 02302 00226 BBEG
07 02303 00011 TR8A
08 02304 002313 MOV 0,0
09 02305 101040 XOP 1,2,11
10 02306 131130 ERROR 0,0,SZC
11 02313 101002 TR8B:
12 02314 150014 ADC# 2,3,SZR
13 ERROR
14 LOOP
15 JSR #ENTLO
16
17 02321 00231
18
19
20
21 02322 00230 TR9:
22 02323 000100
23
24 02324 00222 TRAPR TR9A,BBEG,12,TR9B
25 02325 00233 TR9A
26 02326 00226 JSR #ITRP
27 02327 00012 TR9A
28 02330 00237 TR9B
29 02331 101020 MOV 0,0
30 02332 155230 XOP 2,3,12
31 ERROR
32 02337 101003 TR9A:
33 02340 150014 ADC# 2,3,SZR
34 ERROR
35 LOOP
36 02345 00231

```

```

SETUP 100
JSR #ENTIN
100
TRAPR TR8A,BBEG,11,TR8B
JSR #ITRP
TR8A
JTRAP ORIGIN IN CONTENTS OF BBEG
JTHE CAP NUMBER IS 11
JSUBROUTINE ADDRESS IS TR8B
JSTACK POINTERS SET TO C(BBEG)
MOV 0,0
XOP 1,2,11
ERROR 0,0,SZC
ADC# 2,3,SZR
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE....

```

```

TEST "CAP"
INITIALIZE TEST....
IF STACK FAULT, GO TO TR9A
JTRAP ORIGIN IN CONTENTS OF BBEG
JTHE CAP NUMBER IS 12
JSUBROUTINE ADDRESS IS TR9B
JSTACK POINTERS SET TO C(BBEG)
MOV 0,0
XOP 2,3,12
ERROR
JC(CARRY) SHOULD REMAIN (0).
JACS LOCATION SHOULD BE
JONE LESS THAN ACC LOCATION.
ITERATE TEST ROUTINE....

```

10043 ECL22

```

01
02
03 02346 006230
04 02347 000100
05
06 02350 006222
07 02351 002357
08 02352 000205
09 02353 000013
10 02354 002363
11 02355 101020
12 02356 121330
13
14 02363 101003
15 02364 172014
16
17
18 02371 006231

```

```

TR101
SETUP 100
JSR #ENTIN
100
TRAPER TR10A,BBEG,13,TR10B
JSR #ITRP
TR10A
BREG
13
TR10B
MOVZ 0,0
XOP 1,0,13
ERROR
TR10A:
TR10B:
MOV 0,0,SNC
ADC# 3,2,SZR
ERROR JONE GREATER THAN ACC
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE,....

```

10044 ECL22

```

01
02
03 02372 006230
04 02373 000100
05
06 02374 006222
07 02375 002403
08 02376 000226
09 02377 000014
10 02400 002407
11 02401 101040
12 02402 145430
13
14 02407 101002
15 02410 172014
16
17
18 02415 006231
19
20
21 02416 006230
22 02417 000100
23
24
25 02420 006222
26 02421 002427
27 02422 000226
28 02423 000015
29 02424 002433
30 02425 101040
31 02426 171330
32
33 02433 101002
34 02434 172014
35
36
37 02441 006231

```

```

TR111
SETUP 100
JSR #ENTIN
100
TRAPER TR11A,BBEG,14,TR11B
JSR #ITRP
TR11A
BREG
14
TR11B
MOVZ 0,0
XOP 2,1,14
ERROR
MOV 0,0,SZC
ADC# 3,2,SZR
ERROR JONE GREATER THAN ACC
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE,....

```

```

TR121
SETUP 100
JSR #ENTIN
100
TRAPER TR12A,BBEG,15,TR12B
JSR #ITRP
TR12A
BREG
15
TR12B
MOVZ 0,0
XOP 3,2,15
ERROR
MOV 0,0,SZC
ADC# 3,2,SZR
ERROR JONE GREATER THAN ACC
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE,....

```

```

TR12A:
TR12B:
MOV 0,0,SZC
ADC# 3,2,SZR
ERROR JONE GREATER THAN ACC
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE,....

```



10045 ECL22

```

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
    LDA 0,BREG
    STA 0,TEM
    LDA 1,SL
    LDA 0,7
    SUB 0,1
    STA 1,TEM1
    SETUP 20
    JSR 0,ENTIN
    TRAPR TRI30,TEM,16,TRI3C
    JSR 0,TRP
    TRI3B
    TEM
    16
    TRI3C
    LDA 0,TEM
    LDA 1,84
    ADD 0,1
    XOP 0,3,16
    ERROR
    ADCORW 0,2,SNR
    SUBN 1,3,SZR
    ERROR
    LOOP
    JSR 0,ENTLO
    ISZ TEM
    LDA 0,TEM
    SUBN 0,1,SZR
    JMP TRI3A
    JTEST "CAP"
    JAT ASCENDING LOCATIONS IN THE
    BUFFER; CHANGE TRAP ORIGIN
    JTEST FOR ACS/ACD ADDRESS
    ILOOK ALSO FOR DISPATCH
    IAND STACK FAULTS.
    JINITIALIZE TEST....
    TRAPR TRI30,TEM,16,TRI3C
    JIF STACK FAULT, GO TO TRI3B
    JIF TRAP ORIGIN IN CONTENTS OF TEM
    JTHE CAP NUMBER IS 16
    JSUBROUTINE ADDRESS IS TRI3C
    JSTACK POINTERS SET TO C(BREG)
    JFAIL TO DISPATCH PROPERLY,
    FOR STACK FAULT.
    JC(AC0)=CORRECT ACS ADDRESS
    JC(AC1)=CORRECT ACD ADDRESS
    JC(AC2)=ACS ADDRESS VIA TRAP
    JC(AC3)=ACD ADDRESS VIA TRAP
    JITERATE TEST ROUTINE....
    JNEXT BUFFER ADDRESS
    JTEST FOR LAST.

```

10046 ECL22

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
    LDA 0,BREG
    STA 0,TEM
    LDA 1,SL
    LDA 0,75
    SUB 0,1
    STA 1,TEM1
    SETUP 20
    JSR 0,ENTIN
    TRAPR TRI40,TEM,17,TRI4C
    JSR 0,TRP
    TRI4R
    TEM
    17
    TRI4C
    LDA 0,TEM
    STA 0,SP
    LDA 1,75
    ADD 0,1
    XOP 1,7,17
    ERROR
    LDA 2,SP
    MOV 0,0,SNR
    SUBN 1,2,SZR
    ERROR
    LOOP
    JSR 0,ENTLO
    ISZ TEM
    LDA 1,TEM1
    SUBN 0,1,SZR
    JMP TRI4A
    JTEST "CAP"
    JINITIALIZE TEST....
    TRAPR TRI40,TEM,17,TRI4C
    JIF STACK FAULT, GO TO TRI4B
    JIF TRAP ORIGIN IN CONTENTS OF TEM
    JTHE CAP NUMBER IS 17
    JSUBROUTINE ADDRESS IS TRI4C
    JSTACK POINTERS SET TO C(BREG)
    ISET C(5P) TO TEM,
    JCAP #17
    JC(AC2)=CURRENT STACK POINTER.
    JC(AC1)=CORRECT STACK POINTER.
    JC(CARRY) SHOULD BE (0).
    JITERATE TEST ROUTINE....
    JNEXT BUFFER LOCATION
    JTEST FOR END
    SETUP 200
    JSR 0,ENTIN
    TRAPR TRI5A,BREG,20,TRI5B
    JSR 0,TRP
    TRI5A
    BREG
    20
    TRI5B
    RAND
    JSR 0,ENTRA
    MOVN 0,2
    XOP 0,2,20
    ERROR
    LDA 3,0,3
    LDA 2,0,2
    MOVN 0,1
    SUBN 0,2,SNR
    SUBN 1,3,SZR
    ERROR
    LOOP
    JSR 0,ENTLO
    JITERATE TEST ROUTINE....

```

10045 ECL22

```

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
    LDA 0,BREG
    STA 0,TEM
    LDA 1,SL
    LDA 0,75
    SUB 0,1
    STA 1,TEM1
    SETUP 20
    JSR 0,ENTIN
    TRAPR TRI40,TEM,17,TRI4C
    JSR 0,TRP
    TRI4R
    TEM
    17
    TRI4C
    LDA 0,TEM
    STA 0,SP
    LDA 1,75
    ADD 0,1
    XOP 1,7,17
    ERROR
    LDA 2,SP
    MOV 0,0,SNR
    SUBN 1,2,SZR
    ERROR
    LOOP
    JSR 0,ENTLO
    ISZ TEM
    LDA 1,TEM1
    SUBN 0,1,SZR
    JMP TRI4A
    JTEST "CAP"
    JINITIALIZE TEST....
    TRAPR TRI40,TEM,17,TRI4C
    JIF STACK FAULT, GO TO TRI4B
    JIF TRAP ORIGIN IN CONTENTS OF TEM
    JTHE CAP NUMBER IS 17
    JSUBROUTINE ADDRESS IS TRI4C
    JSTACK POINTERS SET TO C(BREG)
    ISET C(5P) TO TEM,
    JCAP #17
    JC(AC2)=CURRENT STACK POINTER.
    JC(AC1)=CORRECT STACK POINTER.
    JC(CARRY) SHOULD BE (0).
    JITERATE TEST ROUTINE....
    JNEXT BUFFER LOCATION
    JTEST FOR END
    SETUP 200
    JSR 0,ENTIN
    TRAPR TRI5A,BREG,20,TRI5B
    JSR 0,TRP
    TRI5A
    BREG
    20
    TRI5B
    RAND
    JSR 0,ENTRA
    MOVN 0,2
    XOP 0,2,20
    ERROR
    LDA 3,0,3
    LDA 2,0,2
    MOVN 0,1
    SUBN 0,2,SNR
    SUBN 1,3,SZR
    ERROR
    LOOP
    JSR 0,ENTLO
    JITERATE TEST ROUTINE....

```

10045 ECL22

```

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
    LDA 0,BREG
    STA 0,TEM
    LDA 1,SL
    LDA 0,75
    SUB 0,1
    STA 1,TEM1
    SETUP 20
    JSR 0,ENTIN
    TRAPR TRI40,TEM,17,TRI4C
    JSR 0,TRP
    TRI4R
    TEM
    17
    TRI4C
    LDA 0,TEM
    STA 0,SP
    LDA 1,75
    ADD 0,1
    XOP 1,7,17
    ERROR
    LDA 2,SP
    MOV 0,0,SNR
    SUBN 1,2,SZR
    ERROR
    LOOP
    JSR 0,ENTLO
    ISZ TEM
    LDA 1,TEM1
    SUBN 0,1,SZR
    JMP TRI4A
    JTEST "CAP"
    JINITIALIZE TEST....
    TRAPR TRI40,TEM,17,TRI4C
    JIF STACK FAULT, GO TO TRI4B
    JIF TRAP ORIGIN IN CONTENTS OF TEM
    JTHE CAP NUMBER IS 17
    JSUBROUTINE ADDRESS IS TRI4C
    JSTACK POINTERS SET TO C(BREG)
    ISET C(5P) TO TEM,
    JCAP #17
    JC(AC2)=CURRENT STACK POINTER.
    JC(AC1)=CORRECT STACK POINTER.
    JC(CARRY) SHOULD BE (0).
    JITERATE TEST ROUTINE....
    JNEXT BUFFER LOCATION
    JTEST FOR END
    SETUP 200
    JSR 0,ENTIN
    TRAPR TRI5A,BREG,20,TRI5B
    JSR 0,TRP
    TRI5A
    BREG
    20
    TRI5B
    RAND
    JSR 0,ENTRA
    MOVN 0,2
    XOP 0,2,20
    ERROR
    LDA 3,0,3
    LDA 2,0,2
    MOVN 0,1
    SUBN 0,2,SNR
    SUBN 1,3,SZR
    ERROR
    LOOP
    JSR 0,ENTLO
    JITERATE TEST ROUTINE....

```

10045 ECL22

```

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
    LDA 0,BREG
    STA 0,TEM
    LDA 1,SL
    LDA 0,75
    SUB 0,1
    STA 1,TEM1
    SETUP 20
    JSR 0,ENTIN
    TRAPR TRI40,TEM,17,TRI4C
    JSR 0,TRP
    TRI4R
    TEM
    17
    TRI4C
    LDA 0,TEM
    STA 0,SP
    LDA 1,75
    ADD 0,1
    XOP 1,7,17
    ERROR
    LDA 2,SP
    MOV 0,0,SNR
    SUBN 1,2,SZR
    ERROR
    LOOP
    JSR 0,ENTLO
    ISZ TEM
    LDA 1,TEM1
    SUBN 0,1,SZR
    JMP TRI4A
    JTEST "CAP"
    JINITIALIZE TEST....
    TRAPR TRI40,TEM,17,TRI4C
    JIF STACK FAULT, GO TO TRI4B
    JIF TRAP ORIGIN IN CONTENTS OF TEM
    JTHE CAP NUMBER IS 17
    JSUBROUTINE ADDRESS IS TRI4C
    JSTACK POINTERS SET TO C(BREG)
    ISET C(5P) TO TEM,
    JCAP #17
    JC(AC2)=CURRENT STACK POINTER.
    JC(AC1)=CORRECT STACK POINTER.
    JC(CARRY) SHOULD BE (0).
    JITERATE TEST ROUTINE....
    JNEXT BUFFER LOCATION
    JTEST FOR END
    SETUP 200
    JSR 0,ENTIN
    TRAPR TRI5A,BREG,20,TRI5B
    JSR 0,TRP
    TRI5A
    BREG
    20
    TRI5B
    RAND
    JSR 0,ENTRA
    MOVN 0,2
    XOP 0,2,20
    ERROR
    LDA 3,0,3
    LDA 2,0,2
    MOVN 0,1
    SUBN 0,2,SNR
    SUBN 1,3,SZR
    ERROR
    LOOP
    JSR 0,ENTLO
    JITERATE TEST ROUTINE....

```

10047 ECL22

```
01
02 LDA R,BBEG
03 STA 0,TEM
04 LDA 1,S
05 LDA 1,SL
06 LDA 0,M5
07 SUR R,1
08 STA 1,TEM1
09 SETUP 20
10 JSP #ENTIN
11 JSP #ENTIN
12 TRAPR TR16B,TEM,21,TR16C
13 JSR #ENTR
14 TR16B
15 TEM
16 TR16C
17 LDA 0,TEM
18 STA 0,ISP
19 RAND
20 JSR #ENTRA
21 MOV 0,1
22 MOV 1,3
23 XDP 1,3,21
24 ERROR
25 MOV 0,0
26 LDA 2,0,2
27 LDA 3,0,3
28 SUB# 0,3,SNR
29 SUB# 1,2,1SZR
30 ERROR
31 LOOP
32 JSR #ENTLO
33 IS2 TEM
34 LDA 0,TEM
35 STA 1,TEM1
36 SUB# 0,1,1SZR
37 JMP TR16A
38

10048 ECL22
01
02 JTEST #CAP#
03 INITIALIZE TEST....
04 TRAPER TR17A,BBEG,22,TR17B
05 JSR #ENTR
06 TR17A
07 BREG
08 TR17B
09 RAND
10 JSR #ENTRA
11 MOV 0,3
12 XDP 3,0,22
13 ERROR
14 LDA 1,0,3
15 LDA 2,0,2
16 MOV 0,1
17 SUB# 0,1,SNR
18 SUB# 2,3,1SZR
19 ERROR
20 LOOP
21 JSR #ENTLO
22
24
```

10048 ECL22

```
01
02 TR17A
03 02644 006230
04 02645 000100
05 02646 006222
06 02647 002656
07 02650 000226
08 02651 000022
09 02652 002062
10
11 RAND
12 JSR #ENTRA
13 MOV 0,3
14 XDP 3,0,22
15 ERROR
16 TR17A
17 TR17B
18
19
20
21
22
23
24
```

10048 ECL22

```

01
02
03 02674 020226 TR181 LDA 0, BBEG
04 STA 0, TEM
05 02675 040240 LDA 1, BL
06 02676 024642 LDA 0, RS
07 02677 020577 -
08 02700 100400 STA 0, 1
09 02701 044273 TR181 STA 1, TEM1
10 SETUP 20
11 02702 000230 TR181 JSR #ENTIN
12 02703 000020
13 02704 000222 TRAPER TR18B, TEM, 23, TR18C
14 02705 002216 JIF STACK FAULT, GO TO TR18B
15 02706 000240 TR18B TRAP ORIGIN IN CONTENTS OF TEM
16 02707 000240 TEM
17 02708 000623 JTHE CAP NUMBER IS 23
18 02709 002222 JSUBROUTINE ADDRESS IS TR18C
19 02710 020240 TR18C LDA 0, TEM
20 02711 020240 STA 0, SP
21 02712 040040 JINITIALIZE STACK POINTER
22 02713 000234 JRAND
23 02714 111300 JSR #ENTRA
24 02715 142330 MOVZ 0, 2
25 02722 031000 TR181 XOP 2, 0, 2
26 02723 035400 TR181 LDA 3, 0, 3
27 02724 103300 MOVZ 0, 1
28 02725 110415 SUB# 0, 3, SMC
29 02726 132414 SUB# 1, 2, SZR
30 ERROR
31 LOOP
32 02733 000231 JSR #ENTLO
33 02734 010240 TR181 ISZ TEM
34 02735 020240 LDA 0, TEM
35 02736 024273 LDA 1, TEM1
36 02737 100414 SUB# 0, 1, SZR
37 02740 000742 JMP TR18A
38
39 SETUP 100
40 02741 000230 TR191 JSR #ENTIN
41 02742 000100
42 TRAPER TR19A, BBEG, 24, TR19B
43 02743 000222 JIF STACK FAULT, GO TO TR19A
44 02744 002752 TR19A TRAP ORIGIN IN CONTENTS OF BBEG
45 02745 000220 BBEG
46 02746 000024 JTHE CAP NUMBER IS 24
47 02747 002750 TR19B JSUBROUTINE ADDRESS IS TR19B
48 02750 101020 JSTACK POINTERS SET TO C(BBEG)
49 02751 160430 MOVZ 0, 0
50 XOP 3, 1, 24
51 ERROR
52 02756 021001 TR191 LDA 0, 1, 3
53 02757 101112 TR191 MOVL# 0, 0, SZC
54 ERROR
55 02764 000231 TR191 JSR #ENTLO

```

10050 ECL22

```

01
02 ITEST "CAP"
03 INITIALIZE TEST, ....
04 SETUP 100
05 JSR #ENTIN
06 LDA 0, BBEG
07 TRAPER TR20A, BBEG, 25, TR20B
08 JIF STACK FAULT, GO TO TR20A
09 TRAP ORIGIN IN CONTENTS OF BBEG
10 TR20A TRAP ORIGIN IN CONTENTS OF BBEG
11 JTHE CAP NUMBER IS 25
12 JSUBROUTINE ADDRESS IS TR20B
13 JSTACK POINTERS SET TO C(BBEG)
14 MOVZ 0, 0
15 XOP 3, 1, 25
16 ERROR
17 TR20A1 LDA 0, 3, 3
18 TR20B1 MOVL# 0, 0, SMC
19 ERROR
20 JSTORED BY TRAP, C(CARRY)
21 SHOULD BE (1).
22 ITERATE TEST ROUTINE, ....
23 SETUP 200
24 03011 000230 TR211 JSR #ENTIN
25 03012 000200
26 TRAPER TR21A, BBEG, 26, TR21B
27 03013 000222 JIF STACK FAULT, GO TO TR21A
28 03014 003023 TR21A TRAP ORIGIN IN CONTENTS OF BBEG
29 03015 000220 BBEG
30 03016 000026 JTHE CAP NUMBER IS 26
31 03017 003027 JSUBROUTINE ADDRESS IS TR21B
32 03018 101020 JSTACK POINTERS SET TO C(BBEG)
33 03019 004401 MOVZ 0, 0
34 03021 004401 JSR #I
35 XOP 0, 3, 26
36 ERROR
37 TR21A1 LDA 0, 1, 3
38 TR21B1 LDA 1, 0, 3
39 MOVL# 0, 0, SZR
40 ADCK 1, 0, SZR
41 ERROR
42 JSTACK POINTERS SET TO C(BBEG)
43 ITERATE TEST ROUTINE, ....
44 SETUP 100
45 03036 000231 TR211 JSR #ENTLO

```

```

ITEST "CAP"
INITIALIZE TEST, ....
SETUP 100
JSR #ENTIN
LDA 0, BBEG
TRAPER TR18B, TEM, 23, TR18C
JIF STACK FAULT, GO TO TR18B
TRAP ORIGIN IN CONTENTS OF TEM
JTHE CAP NUMBER IS 23
JSUBROUTINE ADDRESS IS TR18C
JSTACK POINTERS SET TO C(BBEG)
INITIALIZE STACK POINTER
JRAND
JSR #ENTRA
MOVZ 0, 2
XOP 2, 0, 2
LDA 3, 0, 3
MOVZ 0, 1
SUB# 0, 3, SMC
SUB# 1, 2, SZR
ERROR
LOOP
JSR #ENTLO
ISZ TEM
LDA 0, TEM
LDA 1, TEM1
SUB# 0, 1, SZR
JMP TR18A

SETUP 100
JSR #ENTIN
TRAPER TR19A, BBEG, 24, TR19B
JIF STACK FAULT, GO TO TR19A
TRAP ORIGIN IN CONTENTS OF BBEG
JTHE CAP NUMBER IS 24
JSUBROUTINE ADDRESS IS TR19B
JSTACK POINTERS SET TO C(BBEG)
MOVZ 0, 0
XOP 3, 1, 24
ERROR
LDA 0, 1, 2
MOVL# 0, 0, SZC
ERROR
JSR #ENTLO

```

```

ITEST "CAP"
INITIALIZE TEST, ....
SETUP 100
JSR #ENTIN
LDA 0, BBEG
TRAPER TR20A, BBEG, 25, TR20B
JIF STACK FAULT, GO TO TR20A
TRAP ORIGIN IN CONTENTS OF BBEG
JTHE CAP NUMBER IS 25
JSUBROUTINE ADDRESS IS TR20B
JSTACK POINTERS SET TO C(BBEG)
MOVZ 0, 0
XOP 3, 1, 25
ERROR
LDA 0, 3, 3
MOVL# 0, 0, SMC
ERROR
JSTORED BY TRAP, C(CARRY)
JSHOULD BE (1).
ITERATE TEST ROUTINE, ....
SETUP 200
JSR #ENTIN
TRAPER TR21A, BBEG, 26, TR21B
JIF STACK FAULT, GO TO TR21A
TRAP ORIGIN IN CONTENTS OF BBEG
JTHE CAP NUMBER IS 26
JSUBROUTINE ADDRESS IS TR21B
JSTACK POINTERS SET TO C(BBEG)
MOVZ 0, 0
JSR #I
XOP 0, 3, 26
ERROR
LDA 0, 1, 3
LDA 1, 0, 3
MOVL# 0, 0, SZR
ADCK 1, 0, SZR
ERROR
JSTACK POINTERS SET TO C(BBEG)
ITERATE TEST ROUTINE, ....
SETUP 100
JSR #ENTLO

```

```

10051 ECL22
01
02 03037 04226 TR22:
03 03040 03071-
04 03041 12000
05 03042 04226
06 03043 04226
07 03044 04226 TR22A1
08 03045 04226
09 03046 04226
10 03047 04226
11 03048 04226
12 03049 04226
13 03050 04226
14 03051 04226
15 03052 04226
16 03053 04226
17 03054 04226
18 03055 04226
19 03056 04226
20 03057 04226
21 03058 04226 TR22B1
22 03059 04226 TR22C1
23 03060 04226
24 03061 04226
25 03062 04226
26 03063 04226
27 03064 04226 TR22D1
28 03065 04226
29 03066 04226
30 03067 04226
31 03068 04226
32 03069 04226
33 03070 04226 TR22E1
34

TEST "CAP"
/TEST PC AND CARRY BY
/EXECUTING A TRAP INSTRUCTION
/IN THE BUFFER.
/INITIALIZE TEST....

TRAPR TR22B,08EG,27,TR22C
/IF STACK FAULT, GO TO TR22B
/IF TRAP ORIGIN IN CONTENTS OF 08EG
/IF THE CAP NUMBER IS 27
/IF SUBROUTINE ADDRESS IS TR22C
/IF STACK POINTERS SET TO C(08EG)

/STORE A TRAP INSTRUCTION
/GO TO THE BUFFER, AND DO IT.
/DISPATCH OR STACK FAULT.
/IC(AC3)=PC STORED VIA TRAP
/IC(AC1)+1=CORRECT PC

/ITERATE TEST ROUTINE....
/NEXT BUFFER LOCATION
/TEST FOR END.

LOA 1,08EG
LOA 0,100
ADD 1,0
STA 0,TEM
SETUP 5
JSR 0,ENTIN
S
TRAPR TR22B,08EG,27,TR22C
JSR 0,ITRP
TR22B
08EG
27
TR22C
LOA 0,TR22E
LOA 2,TEM
MOVZ 2,1
STA 0,0,2
JMP 0,2
ERROR
LOA 3,1,3
ADCM 1,3,SZR
ERRDR
LOOP
JSR 0,ENTLO
ISZ TEM
LOA 0,TEM
LOA 1,BEND
SUBM 0,1,SZR
JMP TR22A
JMP TR22
XOP 0,3,27

10052 ECL22
01
02 03101 02426 TR23:
03 03102 02071-
04 03103 12000
05 03104 04226 TR23A1
06 03105 04226
07 03106 04226
08 03107 04226
09 03108 04226
10 03109 04226
11 03110 04226
12 03111 04226
13 03112 04226
14 03113 04226
15 03114 04226
16 03115 04226
17 03116 04226
18 03117 04226
19 03118 04226
20 03119 04226
21 03120 04226 TR23B1
22 03121 04226 TR23C1
23 03122 04226
24 03123 04226
25 03124 04226
26 03125 04226
27 03126 04226
28 03127 04226
29 03128 04226
30 03129 04226
31 03130 04226
32 03131 04226
33 03132 04226 TR23D1
34

TEST "CAP"
/TEST PC AND CARRY BY
/EXECUTING A TRAP INSTRUCTION
/IN THE BUFFER.
/INITIALIZE TEST....

TRAPR TR23B,08EG,30,TR23C
/IF STACK FAULT, GO TO TR23B
/IF TRAP ORIGIN IN CONTENTS OF 08EG
/IF THE CAP NUMBER IS 30
/IF SUBROUTINE ADDRESS IS TR23C
/IF STACK POINTERS SET TO C(08EG)

/STORE A TRAP INSTRUCTION
/GO TO THE BUFFER, AND DO IT.
/DISPATCH OR STACK FAULT.
/IC(AC3)=PC STORED VIA TRAP
/IC(AC1)+1=CORRECT PC

/ITERATE TEST ROUTINE....

LOA 1,08EG
LOA 0,100
ADD 1,0
STA 0,TEM
SETUP 5
JSR 0,ENTIN
S
TRAPR TR23B,08EG,30,TR23C
JSR 0,ITRP
TR23B
08EG
30
TR23C
LOA 0,TR23D
LOA 2,TEM
MOVZ 2,1
STA 0,0,2
JMP 0,2
ERROR
LOA 3,1,3
ADCM 1,3,SZR
ERRDR
LOOP
JSR 0,ENTLO
ISZ TEM
LOA 0,TEM
LOA 1,BEND
SUBM 0,1,SZR
JMP TR23A
JMP TR23
XOP 1,3,30

```

10053 ECL22

01  
02  
03  
04 03144 006230  
05 03145 006190  
06  
07 03146 006222  
08 03147 006170  
09 03150 006226  
10 03151 006031  
11 03152 006157  
12  
13 03153 006234  
14 03154 106400  
15 03155 11300  
16 03156 156401  
17 03167 107710 TR24A:  
18 03160 133130  
19 03161 106015  
20 03162 156014  
21  
22 03167 101001  
23  
24  
25 03174 006231  
26  
27  
28 03175 006230  
29 03176 006100  
30  
31 03177 006222  
32 03200 003025  
33 03201 006226  
34 03202 006032  
35 03203 106212  
36  
37 03204 006234  
38 03205 106340  
39 03206 111000  
40 03207 115000  
41 03210 123230  
42 03211 006405  
43 03212 106215 TR25A:  
44 03213 106214  
45 03214 006215  
46 03215 107710  
47 03216 156415 TH260:  
48 03217 112414  
49  
50 03224 101001  
51  
52  
53 03231 006231  
54

10054 ECL22

01  
02 03232 036225 TR261  
03 03233 155400  
04 03234 024227  
05 03235 146400  
06 03236 006051-  
07 03237 041000  
08 03240 102400  
09 03241 113710  
10 03242 002020 TR26A:  
11 03243 002040  
12 03244 020042  
13 03245 024057-  
14 03246 102400  
15 03247 040273  
16  
17 03250 006230  
18 03251 002020  
19  
20 03252 006222  
21 03253 003300  
22 03254 006240  
23 03255 006033  
24 03256 106240  
25 03257 004402  
26 03260 000421  
27 03261 054274  
28 03262 030240  
29 03263 000040  
30 03264 000050-  
31 03265 041000  
32  
33 03266 006234  
34 03267 104400  
35 03270 101040  
36 03271 173330  
37 03272 107015 TR26C:  
38 03273 101003  
39  
40 03300 101001  
41  
42 03305 101001 TR26D:  
43  
44 03312 020051-  
45 03313 042240  
46  
47 03314 006231  
48 03315 010240 TR26F:  
49 03316 002040  
50 03317 024273  
51 03320 106414  
52 03321 000727

!TEST "CAP"  
!INITIALIZE TEST....  
!IF STACK FAULT, GO TO TR24B  
!TRAP ORIGIN IN CONTENTS OF BBEG  
!THE CAP NUMBER IS 31  
!SUBROUTINE ADDRESS IS TR24A  
!STACK POINTERS SET TO C(BBEG)  
!RANDOM  
!JSP #ENTRA  
!INC 0,1  
!MOV 0,2  
!POP  
!XDP 1,2,31  
!ADC 0,1,SNR  
!ERR 2,3,SNR  
!MOV 0,0,SKP  
!LOOP  
!JSP #ENTLO  
!RANDOM  
!MOVOS 0,1  
!MOV 0,2  
!XDP 1,0,32  
!JMP TR25B  
!\*,\*1 !DISPATCH TABLE IS  
!\*,\*1 !INDIRECT 3 LEVELS  
!\*,\*1  
!POP  
!SUB# 1,3,SNR  
!SUB# 0,2,SNR  
!ENRR  
!MOV 0,0,SKP  
!ERRR  
!LOOP  
!JSP #ENTLO  
!ITERATE TEST ROUTINE....  
!TEST "CAP"  
!INITIALIZE TEST....  
!IF STACK FAULT, GO TO TR24B  
!TRAP ORIGIN IN CONTENTS OF BBEG  
!THE CAP NUMBER IS 31  
!SUBROUTINE ADDRESS IS TR24A  
!STACK POINTERS SET TO C(BBEG)  
!RANDOM  
!JSP #ENTRA  
!INC 0,1  
!MOV 0,2  
!POP  
!XDP 1,2,31  
!ADC 0,1,SNR  
!ERR 2,3,SNR  
!MOV 0,0,SKP  
!LOOP  
!JSP #ENTLO  
!RANDOM  
!MOVOS 0,1  
!MOV 0,2  
!XDP 1,0,32  
!JMP TR25B  
!\*,\*1 !DISPATCH TABLE IS  
!\*,\*1 !INDIRECT 3 LEVELS  
!\*,\*1  
!POP  
!SUB# 1,3,SNR  
!SUB# 0,2,SNR  
!ENRR  
!MOV 0,0,SKP  
!ERRR  
!LOOP  
!JSP #ENTLO  
!ITERATE TEST ROUTINE....  
!TEST "CAP"  
!INITIALIZE TEST....  
!IF STACK FAULT, GO TO TR26E  
!TRAP ORIGIN IN CONTENTS OF TEP  
!THE CAP NUMBER IS 33  
!SUBROUTINE ADDRESS IS 0TEM  
!STACK POINTERS SET TO C(BBEG)  
!RANDOM  
!JSP #ENTRA  
!INC 0,1  
!MOV 0,0  
!XDP 3,2,33  
!ADD 0,1,SNR  
!MOV 0,0,SNR  
!ERRR  
!MOV 0,0,SKP  
!ERRR  
!MOV 0,0,SKP  
!ERRR  
!LOA 0,0,TEM  
!LOOP  
!JSP #ENTLO  
!ITERATE TEST ROUTINE....  
!TEST "CAP"  
!INITIALIZE TEST....  
!IF STACK FAULT, GO TO TR26E  
!TRAP ORIGIN IN CONTENTS OF TEP  
!THE CAP NUMBER IS 33  
!SUBROUTINE ADDRESS IS 0TEM  
!STACK POINTERS SET TO C(BBEG)  
!RANDOM  
!JSP #ENTRA  
!INC 0,1  
!MOV 0,0  
!XDP 3,2,33  
!ADD 0,1,SNR  
!MOV 0,0,SNR  
!ERRR  
!MOV 0,0,SKP  
!ERRR  
!LOA 0,0,TEM  
!LOOP  
!JSP #ENTLO  
!ITERATE TEST ROUTINE....

```

10055 ECL22
01
02
03 03322 006230 TR271 SETUP 40 ITEST 'CAP'
04 03323 000040 JSR #ENTIN INITIALIZE TEST,....
05 03324 000040 JSR #ENTIN INITIALIZE TEST
06 03325 000040 TRAPER TR270,BREG,34,#TR27A JSR #ITRP
07 03326 000222 JSR #ITRP IIF STACK FAULT, GO TO TR27B
08 03327 000335 TR27B IIF STACK FAULT, GO TO TR27B
09 03328 000226 BREG IIF STACK FAULT, GO TO TR27B
10 03329 000034 BREG IIF STACK FAULT, GO TO TR27B
11 03330 100332 #TR27A IIF STACK FAULT, GO TO TR27B
12 03331 133430 #TR27A IIF STACK FAULT, GO TO TR27B
13 03332 100333 TR27A1 IIF STACK FAULT, GO TO TR27B
14 03333 000334 #TR27A IIF STACK FAULT, GO TO TR27B
15 03334 101001 #TR27A IIF STACK FAULT, GO TO TR27B
16
17
18 03341 006231 TR2701 JSR #ENTLO IIF STACK FAULT, GO TO TR270B
19
20
21 03342 006230 TR281 SETUP 40 ITEST 'CAP'
22 03343 000040 JSR #ENTIN INITIALIZE TEST,....
23
24 03344 006222 TRAPER TR280,BREG,35,#TR28A JSR #ITRP
25 03345 003354 TR28B IIF STACK FAULT, GO TO TR28B
26 03346 000226 BREG IIF STACK FAULT, GO TO TR28B
27 03347 000035 BREG IIF STACK FAULT, GO TO TR28B
28 03350 100352 #TR28A IIF STACK FAULT, GO TO TR28B
29 03351 100350 #TR28A IIF STACK FAULT, GO TO TR28B
30 03352 000353 TR28A1 IIF STACK FAULT, GO TO TR28B
31 03353 101001 TR2801 IIF STACK FAULT, GO TO TR28B
32
33
34 03360 006231 TR291 JSR #ENTLO IITERATE TEST ROUTINE,....
35
36
37 03361 006230 TR291 SETUP 40 ITEST 'CAP'
38 03362 000040 JSR #ENTIN INITIALIZE TEST,....
39
40 03363 006222 TRAPER TR290,BREG,36,#TR29A JSR #ITRP
41 03364 003373 TR29B IIF STACK FAULT, GO TO TR29B
42 03365 000226 BREG IIF STACK FAULT, GO TO TR29B
43 03366 000036 BREG IIF STACK FAULT, GO TO TR29B
44 03367 100371 #TR29A IIF STACK FAULT, GO TO TR29B
45 03370 127630 #TR29A IIF STACK FAULT, GO TO TR29B
46 03371 003372 TR29A1 IIF STACK FAULT, GO TO TR29B
47 03372 101001 TR2901 IIF STACK FAULT, GO TO TR29B
48
49
50 03377 006231 TR2901 IITERATE TEST ROUTINE,....

```

|             |                  |       |                |                          |  |
|-------------|------------------|-------|----------------|--------------------------|--|
| 10057 ECL22 |                  |       |                |                          |  |
| 01          |                  |       |                |                          |  |
| 02          | 03 03422 006230  | OV11: | SETUP 200      | TEST STACK OVERFLOW      |  |
| 03          | 04 03423 006200  |       | JSR #ENTLN     | INITIALIZE TEST....      |  |
| 04          | 05 03424 020402  |       | OVLM BREG,OV1A |                          |  |
| 05          | 06 03425 101021  |       | LDA 0,+2       | THE ADDRESS TO GO TO     |  |
| 06          | 07 03426 003443  |       | MOVZ 0,0,SKP   |                          |  |
| 07          | 08 03427 040045  |       | STA 0,5F       | ION OVERFLOW IS OV1A.    |  |
| 08          | 09 03430 040045  |       | STA 0,5F       |                          |  |
| 09          | 10 03431 020220  |       | LDA 0,BREG     | SET C(AC0),STACK, AND    |  |
| 10          | 11 03432 040040  |       | STA 0,5P       | IFRAME POINTERS TO       |  |
| 11          | 12 03433 040042  |       | STA 0,5L       | THE CONTENTS OF BREG.    |  |
| 12          | 13 03434 040041  |       | STA 0,5P       | TEST OF STACK OVERFLOW.  |  |
| 13          | 14 03435 103110  |       | PSH 0          | PSHR FAILED TO OVERFLOW. |  |
| 14          | 15 03436 024040  |       | LDA 1,5P       |                          |  |
| 15          | 16 03443 006231  | OV1A1 | ERROR          |                          |  |
| 16          | 17 03444 006230  |       | LOOP           |                          |  |
| 17          | 18 03445 006200  |       | JSR #ENTLO     | ITERATE TEST ROUTINE.... |  |
| 18          | 19 03446 020402  | OV2:  | SETUP 200      | TEST STACK OVERFLOW      |  |
| 19          | 20 03447 101021  |       | JSR #ENTLN     | INITIALIZE TEST....      |  |
| 20          | 21 03448 003443  |       | OVLM BREG,OV2A |                          |  |
| 21          | 22 03449 040045  |       | LDA 0,+2       | THE ADDRESS TO GO TO     |  |
| 22          | 23 03450 040045  |       | MOVZ 0,0,SKP   |                          |  |
| 23          | 24 03451 040045  |       | STA 0,5F       | ION OVERFLOW IS OV2A.    |  |
| 24          | 25 03453 020220  |       | STA 0,5F       |                          |  |
| 25          | 26 03454 040040  |       | LDA 0,BREG     | SET C(AC0),STACK, AND    |  |
| 26          | 27 03455 040042  |       | STA 0,5P       | IFRAME POINTERS TO       |  |
| 27          | 28 03456 040041  |       | STA 0,5L       | THE CONTENTS OF BREG.    |  |
| 28          | 29 03457 103110  |       | STA 0,5P       | TEST OF STACK OVERFLOW.  |  |
| 29          | 30 03458 024040  |       | PSH 0          | PSHR FAILED TO OVERFLOW. |  |
| 30          | 31 03465 006231  | OV2A1 | LDA 1,5P       |                          |  |
| 31          | 32 03466 024040  |       | ERROR          |                          |  |
| 32          | 33 03467 040045  |       | LOOP           |                          |  |
| 33          | 34 03468 006230  |       | JSR #ENTLO     | ITERATE TEST ROUTINE.... |  |
| 34          | 35 03469 006230  |       |                |                          |  |
| 35          | 36 03470 006230  |       |                |                          |  |
| 36          | 37 03471 101021  |       |                |                          |  |
| 37          | 38 03472 003510  |       |                |                          |  |
| 38          | 39 03473 040045  |       |                |                          |  |
| 39          | 40 03474 040045  |       |                |                          |  |
| 40          | 41 03475 020220  |       |                |                          |  |
| 41          | 42 03476 040040  |       |                |                          |  |
| 42          | 43 03477 040042  |       |                |                          |  |
| 43          | 44 03478 040041  |       |                |                          |  |
| 44          | 45 03479 103110  |       |                |                          |  |
| 45          | 46 03480 024040  |       |                |                          |  |
| 46          | 47 03481 006230  |       |                |                          |  |
| 47          | 48 03482 024040  |       |                |                          |  |
| 48          | 49 03483 040045  |       |                |                          |  |
| 49          | 50 03484 040045  |       |                |                          |  |
| 50          | 51 03485 020402  |       |                |                          |  |
| 51          | 52 03486 101021  |       |                |                          |  |
| 52          | 53 03487 003510  |       |                |                          |  |
| 53          | 54 03488 040045  |       |                |                          |  |
| 54          | 55 03489 040045  |       |                |                          |  |
| 55          | 56 03490 020220  |       |                |                          |  |
| 56          | 57 03491 040040  |       |                |                          |  |
| 57          | 58 03492 040042  |       |                |                          |  |
| 58          | 59 03493 040041  |       |                |                          |  |
| 59          | 60 03494 103110  |       |                |                          |  |
| 60          | 61 03495 024040  |       |                |                          |  |
| 61          | 62 03496 006230  |       |                |                          |  |
| 62          | 63 03497 024040  |       |                |                          |  |
| 63          | 64 03498 040045  |       |                |                          |  |
| 64          | 65 03499 040045  |       |                |                          |  |
| 65          | 66 03500 020402  |       |                |                          |  |
| 66          | 67 03501 101021  |       |                |                          |  |
| 67          | 68 03502 003510  |       |                |                          |  |
| 68          | 69 03503 040045  |       |                |                          |  |
| 69          | 70 03504 040045  |       |                |                          |  |
| 70          | 71 03505 020220  |       |                |                          |  |
| 71          | 72 03506 040040  |       |                |                          |  |
| 72          | 73 03507 040042  |       |                |                          |  |
| 73          | 74 03508 040041  |       |                |                          |  |
| 74          | 75 03509 103110  |       |                |                          |  |
| 75          | 76 03510 024040  |       |                |                          |  |
| 76          | 77 03511 006230  | OV3:  | LDA 1,5P       |                          |  |
| 77          | 78 03512 006200  |       | ERROR          |                          |  |
| 78          | 79 03513 006200  |       | LOOP           |                          |  |
| 79          | 80 03514 040044  | OV3A1 | JSR #ENTLU     | ITERATE TEST ROUTINE.... |  |
| 80          | 81 03515 020402  |       |                |                          |  |
| 81          | 82 03516 040040  |       |                |                          |  |
| 82          | 83 03517 020402  | OV4:  | SETUP 200      | TEST STACK OVERFLOW      |  |
| 83          | 84 03518 101021  |       | JSR #ENTLN     | INITIALIZE TEST....      |  |
| 84          | 85 03519 040044  |       | OVLM BREG,OV4A |                          |  |
| 85          | 86 03520 040045  |       | LDA 0,+2       | THE ADDRESS TO GO TO     |  |
| 86          | 87 03521 003535  |       | MOVZ 0,0,SKP   |                          |  |
| 87          | 88 03522 040043  |       | STA 0,5F       | ION OVERFLOW IS OV4A.    |  |
| 88          | 89 03523 040045  |       | STA 0,5F       |                          |  |
| 89          | 90 03524 020220  |       | LDA 0,BREG     | SET C(AC0),STACK, AND    |  |
| 90          | 91 03525 040040  |       | STA 0,5P       | IFRAME POINTERS TO       |  |
| 91          | 92 03526 040042  |       | STA 0,5L       | THE CONTENTS OF BREG.    |  |
| 92          | 93 03527 040041  |       | STA 0,5P       | TEST OF STACK OVERFLOW.  |  |
| 93          | 94 03528 103110  |       | PSH 0          | PSHR FAILED TO OVERFLOW. |  |
| 94          | 95 03529 024040  |       | LDA 1,5P       |                          |  |
| 95          | 96 03530 006230  |       | ERROR          |                          |  |
| 96          | 97 03531 006230  |       | LOOP           |                          |  |
| 97          | 98 03532 006230  |       | JSR #ENTLO     | ITERATE TEST ROUTINE.... |  |
| 98          | 99 03533 006230  |       |                |                          |  |
| 99          | 100 03534 006230 |       |                |                          |  |

10059 ECL22

```

01
02 03536 020226 OV51
03 03537 040240
04 03540 024071
05 03541 020227
06 03542 122400
07 03543 040273 OV5A1
08
09 03544 006230
10 03545 000002
11
12 03546 020402
13 03547 101021
14 03550 003556
15 03551 040043
16 03552 040043
17 03553 020240
18 03554 040040
19 03555 040042
20 03556 040041
21 03557 103710
22 03560 024040
23 03561 030042
24
25 OV5B1
26 03566 006231
27 03567 010240
28 03570 020240
29 03571 024273
30 03572 106414
31 03573 000751
32
33

```

```

LOA 0,0BEG
STA 0,TEMP
LDA 1,0BEG
LDA 0,SEND
SUB 1,0
STA 0,TEMP
SETUP 2
JSR 0,ENTR
MOVZ 0,0,SKP
OVLIM TEMP,OV5B
LDA 0,0,2
MOVZ 0,0,SKP
OV5B
STA 0,SP
LDA 0,TEMP
STA 0,SP
LDA 0,TEMP
STA 0,SP
PUSH 0
LDA 1,SP
LDA 2,SL
ERROR
LOOP
JSR 0,ENTR
ISZ TEMP
LDA 0,TEMP
LDA 1,TEMP
SUB# 0,1,SPR
JMP OV5A1

```

```

10060 ECL22
01 03574 020226 OV61
02 03575 040240
03 03576 024071
04 03577 020227
05 03600 122400
06 03601 040273 OV6A1
07
08 03602 006230
09 03603 000002
10
11 03604 020402
12 03605 101021
13 03606 003625
14 03607 040043
15 03610 040043
16 03611 020240
17 03612 040040
18 03613 040042
19 03614 040041
20 03615 010042
21 03616 107110
22 03617 024040
23 03620 030042
24
25 OV6B1
26 03625 006231
27 03626 010240
28 03627 020240
29 03630 024273
30 03631 106414
31 03632 000750
32
33
34
35 03633 006230 OV71
36 03634 000200
37
38 03635 020402
39 03636 101021
40 03637 003657
41 03640 040043
42 03641 040045
43 03642 020226
44 03643 040040
45 03644 040042
46 03645 040041
47 03646 010042
48 03647 010042
49 03650 115110
50 03651 024040
51 03652 030042
52
53
54 03657 006231 OV7A1
55
56

```

```

LOA 0,0BEG
STA 0,TEMP
LDA 1,0BEG
LDA 0,SEND
SUB 1,0
STA 0,TEMP
SETUP 2
JSR 0,ENTR
OVLIM TEMP,OV6B
LDA 0,0,2
MOVZ 0,0,SKP
OV6B
STA 0,SP
LDA 0,TEMP
STA 0,SP
LDA 0,TEMP
STA 0,SP
LDA 1,SP
LDA 2,SL
ERROR
LOOP
JSR 0,ENTR
ISZ TEMP
LDA 0,TEMP
LDA 1,TEMP
SUB# 0,1,SPR
JMP OV6A1

```

```

TEST FOR STACK OVERFLOW
AT EACH LOCATION IN THE
BUFFER.
SHOULD OVERFLOW,
INITIALIZE TEST....
THE ADDRESS TO GO TO
ION OVERFLOW IS OV6B,
SET C(AC0), STACK, AND
FRAME POINTERS TO
THE CONTENTS OF TEMP.
TEST OF STACK OVERFLOW,
ALLOW 1 WORD AND
OVERFLOW ON SECOND.
C(AC1)=STACK POINTER
C(AC2)=STACK LIMIT
C(AC0)=OLD STACK POINTER
ITERATE TEST ROUTINE....
ADVANCE TO NEXT
BUFFER ADDRESS.
INITIALIZE TEST....
THE ADDRESS TO GO TO
ION OVERFLOW IS OV7A,
SET C(AC0), STACK, AND
FRAME POINTERS TO
THE CONTENTS OF 0BEG.
TEST OF STACK OVERFLOW,
OVERFLOW ON LAST OF
15 WORD PUSH.
C(AC0)=OLD STACK POINTER
C(AC1)=CURRENT STACK POINTER
C(AC2)=STACK LIMIT
ITERATE TEST ROUTINE....

```



```

10061 ECL22
01
02
03 03660 006230
04 03661 000200
05
06 03662 020402
07 03663 101021
08 03664 003755
09 03665 040043
10 03666 040043
11 03667 020226
12 03670 040040
13 03671 040042
14 03672 040041
15 03673 010042
16 03674 010042
17 03675 010042
18 03676 117110
19 03677 024040
20 03700 030042
21
22
23 03705 006231
24
25
26 03706 006230
27 03707 000200
28
29 03710 020402
30 03711 101021
31 03712 003724
32 03713 040043
33 03714 040045
34 03715 020226
35 03716 040040
36 03717 040042
37 03720 040041
38 03721 010042
39 03722 103110
40 03723 101001
41
42
43 03730 006231

INITIALIZE TEST....
)THE ADDRESS TO GO TO
)ON OVERFLOW IS OV8A.
)SET C(AC0),STACK, AND
)FRAME POINTERS TO
)THE CONTENTS OF BBEG.
)TEST OF STACK OVERFLOW.
)SHOULD OVERFLOW STACK.
)ON LAST WORD OF A
)4 WORD PUSH.
)C(AC0)OLD STACK POINTER
)C(AC1)CURRENT STACK POINTER
)C(AC2)STACK LIMIT
)ITERATE TEST ROUTINE....

OV8:
OV8A:
OV9:
OV9A:
OV10:
OV10A:
OV11:
OV12:
OV13:
OV14:
OV15:
OV16:
OV17:
OV18:
OV19:
OV20:
OV21:
OV22:
OV23:
OV24:
OV25:
OV26:
OV27:
OV28:
OV29:
OV30:
OV31:
OV32:
OV33:
OV34:
OV35:
OV36:
OV37:
OV38:
OV39:
OV40:
OV41:
OV42:
OV43:
OV44:
OV45:
OV46:
OV47:
OV48:

```

```

10062 ECL22
01
02
03
04 03731 006230
05 03732 000200
06
07 03733 020402
08 03734 101021
09 03735 003755
10 03736 040043
11 03737 040045
12 03740 020226
13 03741 040040
14 03742 040042
15 03743 040041
16 03744 024056
17 03745 107000
18 03746 044042
19 03747 103710
20 000000
21
22
23 03755 006231
24
25
26 03756 006230
27 03757 000200
28
29 03760 020402
30 03761 101021
31 03762 044002
32 03763 040043
33 03764 040045
34 03765 020226
35 03766 040040
36 03767 040042
37 03770 040041
38 03771 024053
39 03772 107000
40 03773 044042
41 03774 103710
42
43 000000
44
45
46 04002 006231
47
48

```

```

INITIALIZE TEST....
)THE ADDRESS TO GO TO
)ON OVERFLOW IS OV10A.
)SET C(AC0),STACK, AND
)FRAME POINTERS TO
)THE CONTENTS OF BBEG.
)TEST OF STACK OVERFLOW.
)STACK LIMIT IS 1
)GREATER THEN STACK POINTER.
)OVERFLOW SHOULD OCCURE.

OV10:
OV10A:
OV11:
OV12:
OV13:
OV14:
OV15:
OV16:
OV17:
OV18:
OV19:
OV20:
OV21:
OV22:
OV23:
OV24:
OV25:
OV26:
OV27:
OV28:
OV29:
OV30:
OV31:
OV32:
OV33:
OV34:
OV35:
OV36:
OV37:
OV38:
OV39:
OV40:
OV41:
OV42:
OV43:
OV44:
OV45:
OV46:
OV47:
OV48:

```



|                    |        |               |                          |                    |        |               |                          |
|--------------------|--------|---------------|--------------------------|--------------------|--------|---------------|--------------------------|
| 18065 ECL22        | OV16:  | SETUP 200     | TEST MULTI LEVEL         | 18065 ECL22        | OV19:  | SETUP 200     |                          |
| 01 02 04121 006230 |        | JSR 0ENTIN    | INITIALIZE TEST....      | 01 02 04216 006230 |        | JSR 0ENTIN    | INITIALIZE TEST....      |
| 03 04122 000200    |        | 000           |                          | 03 04217 000200    |        | 200           |                          |
| 04 04123 000220    |        | LDA 0,0BEG    | INDIRECT AT STACK        | 04 04220 020402    |        | LDA 0,0,+2    | THE ADDRESS TO GO TO     |
| 05 04124 040040    |        | STA 0,0P      | FAULT LOCATION.          | 05 04221 101021    |        | OV19A         |                          |
| 06 04125 040042    |        | STA 0,0SL     |                          | 06 04222 004234    |        | MOVZ 0,0,SKP  |                          |
| 07 04126 004402    |        | JMP 0,+2      |                          | 07 04223 040043    |        | STA 0,0,0P    | ION OVERFLOW IS OV19A.   |
| 08 04127 000414    |        | JMP OV16A     |                          | 08 04224 040045    |        | LDA 0,0,0P    | SET C(AC0),STACK, AND    |
| 09 04130 020045-   |        | LDA 0,0,0,+3  |                          | 09 04225 020220    |        | LDA 0,0,0P    | FRAME POINTERS TO        |
| 10 04131 040043    |        | STA 0,0,0P    |                          | 10 04226 040040    |        | STA 0,0,0P    | THE CONTENTS OF BBEG.    |
| 11 04132 127110    |        | PSH 1,1       | 0,+1                     | 11 04227 040042    |        | STA 0,0,0P    | TEST OF STACK OVERFLOW.  |
| 12 04133 104134    |        | 0,+1          | 0,+1                     | 12 04230 040041    |        | MOVZ 0,0      | OVERFLOW ON STACK        |
| 13 04134 104135    |        | 0,+1          | 0,+1                     | 13 04231 103120    |        | SAVE 0        | ICHANGED THE STATE       |
| 14 04135 004137    |        | ERROR         | ERROR                    | 14 04232 000000    |        | MOV 0,0,0,SKP |                          |
| 15 04142 101001    | OV16A: | MOV 0,0,0,SKP |                          | 15 04233 103710    |        | ERROR         | OF CARRY.                |
| 16 04147 000231    |        | ERROR         |                          | 16 04234 101002    | OV19A: | LOOP          |                          |
| 17 04152 020402    |        | JSR 0ENTLO    | ITERATE TEST ROUTINE.... | 17 04241 000231    |        | JSR 0ENTLO    | ITERATE TEST ROUTINE.... |
| 18 04153 101021    |        | 000           |                          | 18 04242 000230    |        | 200           |                          |
| 19 04154 040043    |        | SETUP 200     | INITIALIZE TEST....      | 19 04243 000200    |        | OV20:         | INITIALIZE TEST....      |
| 20 04155 040045    |        | JSR 0ENTIN    |                          | 20 04244 020402    |        | LDA 0,0,+2    | THE ADDRESS TO GO TO     |
| 21 04156 040043    |        | 000           |                          | 21 04245 101021    |        | OV20A         |                          |
| 22 04157 020225    |        | LDA 0,0,0BEG  | ION OVERFLOW IS OV17A.   | 22 04246 004200    |        | MOVZ 0,0,SKP  |                          |
| 23 04158 040045    |        | STA 0,0,0P    |                          | 23 04247 040043    |        | STA 0,0,0P    | ION OVERFLOW IS OV20A.   |
| 24 04159 040045    |        | LDA 0,0,0BEG  | SET C(AC0),STACK, AND    | 24 04250 040045    |        | LDA 0,0,0P    | SET C(AC0),STACK, AND    |
| 25 04160 040040    |        | STA 0,0,0P    | FRAME POINTERS TO        | 25 04251 020225    |        | LDA 0,0,0P    | FRAME POINTERS TO        |
| 26 04161 040041    |        | STA 0,0,0P    | THE CONTENTS OF BBEG.    | 26 04252 040040    |        | STA 0,0,0P    | THE CONTENTS OF BBEG.    |
| 27 04162 040041    |        | STA 0,0,0P    | TEST OF STACK OVERFLOW.  | 27 04253 040041    |        | STA 0,0,0P    | TEST OF STACK OVERFLOW.  |
| 28 04163 101020    |        | PSH 0,0       | STACK OVERFLOW SHOULD    | 28 04254 101040    |        | MOVZ 0,0      | OVERFLOW ON STACK        |
| 29 04164 103110    |        | MOV 0,0,0,SKP | NOT CHANGE THE STATE OF  | 29 04255 103710    |        | SAVE 0        | ICHANGED THE STATE       |
| 30 04165 101002    | OV17A: | MOV 0,0,0,SKP | ICARRY.                  | 30 04260 000000    |        | ERROR         | OF CARRY.                |
| 31 04172 006231    |        | ERROR         |                          | 31 04261 101003    | OV20A: | LOOP          |                          |
| 32 04173 006230    |        | JSR 0ENTLO    | ITERATE TEST ROUTINE.... | 32 04265 006231    |        | JSR 0ENTLO    | ITERATE TEST ROUTINE.... |
| 33 04174 000200    |        | 000           |                          | 33 04266 000200    |        | 200           |                          |
| 34 04175 020402    |        | SETUP 200     | INITIALIZE TEST....      | 34 04267 040045    |        | LDA 0,0,+2    | THE ADDRESS TO GO TO     |
| 35 04176 101021    |        | JSR 0ENTIN    |                          | 35 04268 040043    |        | OV18A         |                          |
| 36 04177 004210    |        | 000           |                          | 36 04269 040045    |        | STA 0,0,0P    | ION OVERFLOW IS OV18A.   |
| 37 04178 040043    |        | LDA 0,0,0BEG  | ION OVERFLOW IS OV18A.   | 37 04270 020225    |        | LDA 0,0,0P    | SET C(AC0),STACK, AND    |
| 38 04179 040045    |        | STA 0,0,0P    |                          | 38 04271 040040    |        | LDA 0,0,0P    | FRAME POINTERS TO        |
| 39 04180 040040    |        | LDA 0,0,0BEG  | THE CONTENTS OF BBEG.    | 39 04272 040042    |        | STA 0,0,0P    | THE CONTENTS OF BBEG.    |
| 40 04181 040042    |        | STA 0,0,0P    | TEST OF STACK OVERFLOW.  | 40 04273 040041    |        | STA 0,0,0P    | TEST OF STACK OVERFLOW.  |
| 41 04182 040041    |        | STA 0,0,0P    | STACK OVERFLOW SHOULD    | 41 04274 103110    |        | MOVZ 0,0      | OVERFLOW ON STACK        |
| 42 04183 101020    |        | PSH 0,0       | NOT CHANGE THE STATE OF  | 42 04275 103003    | OV18A: | MOV 0,0,0,SKP |                          |
| 43 04184 103110    |        | MOV 0,0,0,SKP | ICARRY.                  | 43 04280 000000    |        | ERROR         | OF CARRY.                |
| 44 04185 101002    | OV18A: | MOV 0,0,0,SKP |                          | 44 04281 101003    | OV20A: | LOOP          |                          |
| 45 04175 020402    |        | JSR 0ENTLO    | ITERATE TEST ROUTINE.... | 45 04285 006231    |        | JSR 0ENTLO    | ITERATE TEST ROUTINE.... |
| 46 04176 101021    |        | 000           |                          | 46 04286 000200    |        | 200           |                          |
| 47 04177 004210    |        | SETUP 200     | INITIALIZE TEST....      | 47 04287 040045    |        | LDA 0,0,+2    | THE ADDRESS TO GO TO     |
| 48 04200 040043    |        | JSR 0ENTIN    |                          | 48 04288 040043    |        | OV18A         |                          |
| 49 04201 040045    |        | 000           |                          | 49 04289 040045    |        | STA 0,0,0P    | ION OVERFLOW IS OV18A.   |
| 50 04202 020225    |        | LDA 0,0,0BEG  | ION OVERFLOW IS OV18A.   | 50 04290 020225    |        | LDA 0,0,0P    | SET C(AC0),STACK, AND    |
| 51 04203 040040    |        | STA 0,0,0P    |                          | 51 04291 040040    |        | LDA 0,0,0P    | FRAME POINTERS TO        |
| 52 04204 040042    |        | LDA 0,0,0BEG  | THE CONTENTS OF BBEG.    | 52 04292 040042    |        | STA 0,0,0P    | THE CONTENTS OF BBEG.    |
| 53 04205 040041    |        | STA 0,0,0P    | TEST OF STACK OVERFLOW.  | 53 04293 040041    |        | STA 0,0,0P    | TEST OF STACK OVERFLOW.  |
| 54 04206 101040    |        | MOVZ 0,0      | OVERFLOW ON STACK        | 54 04294 101040    |        | MOVZ 0,0      | OVERFLOW ON STACK        |
| 55 04207 103110    |        | SAVE 0        | ICHANGED THE STATE OF    | 55 04295 103110    |        | ERROR         | OF CARRY.                |
| 56 04210 101003    | OV18A: | MOV 0,0,0,SKP |                          | 56 04296 103710    |        | ERROR         | OF CARRY.                |
| 57 04213 006231    |        | ERROR         |                          | 57 04297 000000    |        | LOOP          |                          |
| 58 04214 000200    |        | JSR 0ENTLO    | ITERATE TEST ROUTINE.... | 58 04298 000200    |        | JSR 0ENTLO    | ITERATE TEST ROUTINE.... |
| 59 04215 020402    |        | 000           |                          | 59 04299 040045    |        | LDA 0,0,+2    | THE ADDRESS TO GO TO     |

```

10067 ECL22
01
02 SETUP 200
03 JSR #ENTIN
04 OVLIM BBEG,OV21A
05 LDA 0,0,2
06 MOVZ 0,0,SKP
07 OV21A
08 STA 0,FF
09 LDA 0,BBEG
10 STA 0,SP
11 JSR #ENTLO
12 JSR #ENTLO
13 JSR #ENTLO
14 JSR #ENTLO
15 JSR #ENTLO
16 JSR #ENTLO
17 JSR #ENTLO
18 JSR #ENTLO
19 JSR #ENTLO
20 JSR #ENTLO
21 JSR #ENTLO
22 JSR #ENTLO
23 JSR #ENTLO
24 JSR #ENTLO
25 JSR #ENTLO
26 JSR #ENTLO
27 JSR #ENTLO
28 JSR #ENTLO
29 JSR #ENTLO
30 JSR #ENTLO
31 JSR #ENTLO
32 JSR #ENTLO
33 JSR #ENTLO
34 JSR #ENTLO
35 JSR #ENTLO
36 JSR #ENTLO
37 JSR #ENTLO
38 JSR #ENTLO
39 JSR #ENTLO
40 JSR #ENTLO
41 JSR #ENTLO
42 JSR #ENTLO
43 JSR #ENTLO
44 JSR #ENTLO
45 JSR #ENTLO
46 JSR #ENTLO
47 JSR #ENTLO
48 JSR #ENTLO
49 JSR #ENTLO
50 JSR #ENTLO
51 JSR #ENTLO
52 JSR #ENTLO

10068 ECL22
01
02 SETUP 200
03 JSR #ENTIN
04 OVLIM BBEG,OV23A
05 LDA 0,0,2
06 MOVZ 0,0,SKP
07 OV23A
08 STA 0,FF
09 LDA 0,BBEG
10 STA 0,SP
11 JSR #ENTLO
12 JSR #ENTLO
13 JSR #ENTLO
14 JSR #ENTLO
15 JSR #ENTLO
16 JSR #ENTLO
17 JSR #ENTLO
18 JSR #ENTLO
19 JSR #ENTLO
20 JSR #ENTLO
21 JSR #ENTLO
22 JSR #ENTLO
23 JSR #ENTLO
24 JSR #ENTLO
25 JSR #ENTLO
26 JSR #ENTLO
27 JSR #ENTLO
28 JSR #ENTLO
29 JSR #ENTLO
30 JSR #ENTLO
31 JSR #ENTLO
32 JSR #ENTLO
33 JSR #ENTLO
34 JSR #ENTLO
35 JSR #ENTLO
36 JSR #ENTLO
37 JSR #ENTLO
38 JSR #ENTLO
39 JSR #ENTLO
40 JSR #ENTLO
41 JSR #ENTLO
42 JSR #ENTLO
43 JSR #ENTLO
44 JSR #ENTLO
45 JSR #ENTLO
46 JSR #ENTLO
47 JSR #ENTLO
48 JSR #ENTLO
49 JSR #ENTLO
50 JSR #ENTLO
51 JSR #ENTLO
52 JSR #ENTLO

INITIALIZE TEST....
ITHE ADDRESS TO GO TO
ION OVERFLOW IS OV21A.
ISET C(AC0),STACK, AND
IFRAME POINTERS TO
ITHE CONTENTS OF BBEG.
ITEST OF STACK OVERFLOW.
ICHANGED THE STATE
IOF CARRY.
ITERATE TEST ROUTINE.....

INITIALIZE TEST....
ITHE ADDRESS TO GO TO
ION OVERFLOW IS OV22A.
ISET C(AC0),STACK, AND
IFRAME POINTERS TO
ITHE CONTENTS OF BBEG.
ITEST OF STACK OVERFLOW.
IOVERFLOW ON THE STACK
ICHANGED THE STATE OF
ICARRY.
ITERATE TEST ROUTINE.....

```

10069 ECL22

```

01
02
03 04420 006230
04 04421 006200
05
06 04422 020402
07 04423 101021
08 04424 004444
09 04425 040043
10 04426 040045
11 04427 020226
12 04430 040040
13 04431 040042
14 04432 040041
15
16 04433 006234
17 04434 105400
18 04435 131400
19 04436 163710
20
21
22 04444 106615
23 04445 132814
24
25
26 04452 006231

```

```

SETUP 200
JSR @ENTIN
OV25:
OVLIM BBEG,OV25A
LDA 0,+2
MOVZ 0,0,SKP
OV25A
STA 0,SP
LDA 0,FF
LDA 0,BBEG
STA 0,SP
STA 0,SL
STA 0,FP
RAND
JSR @ENTRA
INC 0,1
INC 1,2
SAVE 0
ERROR
ADCM 0,1,SNR
ADCM 1,2,SR
ERROR
LOOP
JSR @ENTLO

```

```

INITIALIZE TEST....
)THE ADDRESS TO GO TO
)ION OVERFLOW IS OV25A.
)SET C(AC0),STACK, AND
)FRAME POINTERS TO
)THE CONTENTS OF BBEG.
)TEST OF STACK OVERFLOW.
)C(AC0)=RANDOM #
)PLACE C(AC0-2) IN
)ASCENDING ORDER.
)AC(AC3) SHOULD NOT
)BE CHANGED VIA
)THE OVERFLOW.
)ITERATE TEST ROUTINE....

```

10070 ECL22

```

01
02
03 04453 006230
04 04454 000200
05
06 04455 020402
07 04456 101021
08 04457 004471
09 04460 040043
10 04461 040045
11 04462 020226
12 04463 040040
13 04464 040042
14 04465 040041
15
16 04466 006234
17 04467 113000
18 04470 177110
19 04471 116414
20
21
22 04476 006231
23
24
25 04477 006230
26 04500 000200
27
28 04501 020402
29 04502 101021
30 04503 004515
31 04504 040043
32 04505 040045
33 04506 020226
34 04507 040040
35 04510 040042
36 04511 040041
37
38 04512 006234
39 04513 113000
40 04514 103710
41 04515 116414
42
43
44 04522 000231
45
46

```

```

SETUP 200
JSR @ENTIN
OV26:
OVLIM BBEG,OV26A
LOA 0,+2
MOVZ 0,0,SKP
OV26A
STA 0,SP
STA 0,FF
LDA 0,BBEG
LDA 0,FF
LDA 0,BBEG
STA 0,SP
STA 0,SL
STA 0,FP
RAND
JSR @ENTRA
MOV 0,3
PSM 3,3
SUB# 0,3,SR
ERROR
LOOP
JSR @ENTLO

```

```

INITIALIZE TEST....
)THE ADDRESS TO GO TO
)ION OVERFLOW IS OV26A.
)SET C(AC0),STACK, AND
)FRAME POINTERS TO
)THE CONTENTS OF BBEG.
)TEST OF STACK OVERFLOW.
)C(AC0)=RANDOM #
)BE CHANGED VIA
)THE OVERFLOW.
)ITERATE TEST ROUTINE....

```

```

18071 ECL22
01
02 04923 000230
03
04 04524 000230
05 04525 000200
06 04526 020402
07 04527 101021
08 04528 004543
09 04529 040043
10 04530 040043
11 04531 040043
12 04532 040043
13 04533 020226
14 04534 040040
15 04535 040042
16 04536 040041
17 04537 020040
18 04540 024057-
19 04541 103710
20 04542 000000
21 04543 107000
22 04544 136414
23
24
25 04551 000231
26
27
28 04552 000230
29 04553 000200
30 04554 101021
31 04555 040572
32 04556 040043
33 04557 040045
34 04558 040045
35 04559 020226
36 04561 040040
37 04562 040042
38 04563 040041
39 04564 040041
40 04565 000234
41 04566 104000
42 04567 131400
43 04568 103710
44 04569 103710
45 04570 103710
46 04571 000000
47 04572 021401
48 04573 025402
49 04574 031403
50 04575 100015
51 04576 132014
52
53
54 04603 000231
55
56
18072 ECL22
01
02
03 04604 000230
04 04605 000200
05 04606 020402
06 04607 101021
07 04608 040521
08 04609 040043
09 04611 040043
10 04612 040043
11 04613 020226
12 04614 040040
13 04615 040042
14 04616 040041
15 04617 103710
16 04621 021404
17 04622 116414
18
19
20
21 04627 000231
22
23
24 04630 000230
25 04631 000200
26
27 04632 020402
28 04633 101021
29 04634 040547
30 04635 040043
31 04636 040045
32 04637 020226
33 04640 040040
34 04641 040042
35 04642 040041
36 04643 020040
37 04644 024044-
38 04645 103710
39 04647 000000
40 04647 030040
41 04650 107000
42 04651 132414
43
44
45 04656 000231
46
47
48
49
50
51
52
53
54
55
56
18071 ECL22
01
02
03 04524 000230
04 04525 000200
05 04526 020402
06 04527 101021
07 04528 004543
08 04529 040043
09 04530 040043
10 04531 040043
11 04532 040043
12 04533 020226
13 04534 040040
14 04535 040042
15 04536 040041
16 04537 020040
17 04540 024057-
18 04541 103710
19 04542 000000
20 04543 107000
21 04544 136414
22
23
24
25 04551 000231
26
27
28 04552 000230
29 04553 000200
30 04554 101021
31 04555 040572
32 04556 040043
33 04557 040045
34 04558 040045
35 04559 020226
36 04561 040040
37 04562 040042
38 04563 040041
39 04564 040041
40 04565 000234
41 04566 104000
42 04567 131400
43 04568 103710
44 04569 103710
45 04570 103710
46 04571 000000
47 04572 021401
48 04573 025402
49 04574 031403
50 04575 100015
51 04576 132014
52
53
54 04603 000231
55
56
18072 ECL22
01
02
03 04604 000230
04 04605 000200
05 04606 020402
06 04607 101021
07 04608 040521
08 04609 040043
09 04611 040043
10 04612 040043
11 04613 020226
12 04614 040040
13 04615 040042
14 04616 040041
15 04617 103710
16 04621 021404
17 04622 116414
18
19
20
21 04627 000231
22
23
24 04630 000230
25 04631 000200
26
27 04632 020402
28 04633 101021
29 04634 040547
30 04635 040043
31 04636 040045
32 04637 020226
33 04640 040040
34 04641 040042
35 04642 040041
36 04643 020040
37 04644 024044-
38 04645 103710
39 04647 000000
40 04647 030040
41 04650 107000
42 04651 132414
43
44
45 04656 000231
46
47
48
49
50
51
52
53
54
55
56

```

18071 ECL22

18072 ECL22

```

JMP OV33
SETUP 200
JSR 0ENTIN
200
OVLIM BBEG,OV28A
LDA 0,1,2
MOVZ 0,0,8SKP
OV28A
STA 0,0,FF
STA 0,0,FF
LDA 0,0,BBEG
LDA 0,0,SP
STA 0,0,SL
STA 0,0,FF
LDA 1,1,0
SAVE 0
ADD 0,1
SUB# 1,3,8ZR
ERROR
LOOP
JSR 0ENTLO
ITERATE TEST ROUTINE....

SETUP 200
JSR 0ENTIN
200
OVLIM BBEG,OV29A
LDA 0,1,2
MOVZ 0,0,8SKP
OV29A
STA 0,0,FF
LDA 0,0,BBEG
LDA 0,0,SP
STA 0,0,SL
STA 0,0,FF
RAND
JSR 0ENTRA
INC 0,1
INC 1,2
SAVE 0
LDA 0,1,3
LDA 1,2,3
PC(AC3) IS SETUP VIA SAVE.
ADC# 0,1,SNR
PC# 1,2,8ZR
ERROR
LOOP
JSR 0ENTLO
ITERATE TEST ROUTINE....

BYPASS SAVE OVERFLOW TESTS
INITIALIZE TEST....
)THE ADDRESS TO GO TO
)ION OVERFLOW IS OV28A.
)SET C(AC0),STACK, AND
)FRAME POINTERS TO
)THE CONTENTS OF BBEG.
)TEST OF STACK OVERFLOW.
)C(AC3) SHOULD BE THE
)RESULT OF SAVE AND
)STACK POINTER, BEFORE
)OVERFLOW.
)ITERATE TEST ROUTINE....

INITIALIZE TEST....
)THE ADDRESS TO GO TO
)ION OVERFLOW IS OV29A.
)SET C(AC0),STACK, AND
)FRAME POINTERS TO
)THE CONTENTS OF BBEG.
)TEST OF STACK OVERFLOW.
)C(AC0)RANDOM #
)SET C(AC0) IN ASCENDING
)ORDER, SAVE SHOULD CAUSE
)STACK OVERFLOW.
)CHECK MACHINE STATE
)STORED ON THE STACK.
)C(AC3) IS SETUP VIA SAVE.
)NEXT FRAME IS GENERATED
)VIA OVERFLOW.
)ITERATE TEST ROUTINE....

```

```

SETUP 200
JSR 0ENTIN
200
OVLIM BBEG,OV30A
LDA 0,1,2
MOVZ 0,0,8SKP
OV30A
STA 0,0,FF
LDA 0,0,BBEG
LDA 0,0,SP
STA 0,0,SL
STA 0,0,FF
SAVE 0
LDA 0,4,3
SUB# 0,3,8ZR
ERROR
LOOP
JSR 0ENTLO
ITERATE TEST ROUTINE....

SETUP 200
JSR 0ENTIN
200
OVLIM BBEG,OV31A
LDA 0,1,2
MOVZ 0,0,8SKP
OV31A
STA 0,0,FF
LDA 0,0,BBEG
LDA 0,0,SP
STA 0,0,SL
STA 0,0,FF
LDA 1,1,0.
SAVE 0
LDA 2,SP
ADD 0,1
SUB# 1,2,8ZR
ERROR
LOOP
JSR 0ENTLO
ITERATE TEST ROUTINE....

SETUP 200
JSR 0ENTIN
200
OVLIM BBEG,OV31A
LDA 0,1,2
MOVZ 0,0,8SKP
OV31A
STA 0,0,FF
LDA 0,0,BBEG
LDA 0,0,SP
STA 0,0,SL
STA 0,0,FF
LDA 1,1,0.
SAVE 0
LDA 2,SP
ADD 0,1
SUB# 1,2,8ZR
ERROR
LOOP
JSR 0ENTLO
ITERATE TEST ROUTINE....

```

10073 ECL22

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
01 00750 030226 OV341
02 04751 050240
03 04752 024227
04 04753 020042-
05 04754 120000
06 04755 040273
07 04756 155400
08 04757 146400
09 04758 020041-
10 04759 041000
11 04760 102400
12 04761 113710
13 04762 000000
14 04763 000000
15 04764 000000
16 04765 000000
17 04766 000000
18 04767 000000
19 04768 000000
20 04769 000000
21 04770 000000
22 04771 000000
23 04772 000000
24 04773 000000
25 04774 000000
26 04775 000000
27 04776 000000
28 04777 000000
29 04778 000000
30 04779 000000
31 04780 000000
32 04781 000000
33 04782 000000
34 04783 000000
35 04784 000000
36 04785 000000
37 04786 000000
38 04787 000000
39 04788 000000
40 04789 000000
41 04790 000000
42 04791 000000
43 04792 000000
44 04793 000000
45 04794 000000
46 04795 000000
47 04796 000000
48 04797 000000
49 04798 000000
50 04799 000000
51 04800 000000
52 04801 000000
53 04802 000000
54 04803 000000
55 04804 000000

```

10074 ECL22

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
01 04750 030226 OV341
02 04751 050240
03 04752 024227
04 04753 020042-
05 04754 120000
06 04755 040273
07 04756 155400
08 04757 146400
09 04758 020041-
10 04759 041000
11 04760 102400
12 04761 113710
13 04762 000000
14 04763 000000
15 04764 000000
16 04765 000000
17 04766 000000
18 04767 000000
19 04768 000000
20 04769 000000
21 04770 000000
22 04771 000000
23 04772 000000
24 04773 000000
25 04774 000000
26 04775 000000
27 04776 000000
28 04777 000000
29 04778 000000
30 04779 000000
31 04780 000000
32 05007 000231
33 05010 010240
34 05012 020240
35 05013 024273
36 05014 106414
37 05014 000750
38

```

10075 ECL22

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
01 04750 030226 OV341
02 04751 050240
03 04752 024227
04 04753 020042-
05 04754 120000
06 04755 040273
07 04756 155400
08 04757 146400
09 04758 020041-
10 04759 041000
11 04760 102400
12 04761 113710
13 04762 000000
14 04763 000000
15 04764 000000
16 04765 000000
17 04766 000000
18 04767 000000
19 04768 000000
20 04769 000000
21 04770 000000
22 04771 000000
23 04772 000000
24 04773 000000
25 04774 000000
26 04775 000000
27 04776 000000
28 04777 000000
29 04778 000000
30 04779 000000
31 04780 000000
32 05007 000231
33 05010 010240
34 05012 020240
35 05013 024273
36 05014 106414
37 05014 000750
38

```

|    |              |        |               |                            |  |
|----|--------------|--------|---------------|----------------------------|--|
| 01 | 10075 ECL22  |        |               |                            |  |
| 02 | 05015 000230 | OV351  | SETUP 200     | TEST FOR UNDERFLOW ON POPJ |  |
| 03 | 05016 000200 |        | JSR #ENTLN    | INITIALIZE TEST....        |  |
| 04 | 05017 000200 |        | 200           |                            |  |
| 05 | 05017 020040 |        | UFLIM OV35A   |                            |  |
| 06 | 05020 040040 |        | LDA 0, #PBK   | ISETUP STACK IN PAGE 0     |  |
| 07 | 05021 020220 |        | STA 0, #P     | ITO UNDER FLOW ON ANY      |  |
| 08 | 05022 040042 |        | LDA 0, #BREG  | POPI, POPJ, RTN, OR POPB.  |  |
| 09 | 05022 040042 |        | STA 0, #SL    | IINSTRUCTION, THE FAULT    |  |
| 10 | 05023 020037 |        | LDA 0, #OV35A | LOCATION IS OV35A.         |  |
| 11 | 05024 040043 |        | STA 0, #F     |                            |  |
| 12 | 05025 103710 |        | PBR           | PC STORED POINTS TO ERROR. |  |
| 13 | 05026 117710 |        | POPJ          | I SHOULD CAUSE UNDERFLOW.  |  |
| 14 |              |        | ERROR         | POPJ FAILS TO UNDERFLOW.   |  |
| 15 | 05033 000231 | OV35A: | LOOP          |                            |  |
| 16 | 05034 000230 | OV361  | JSR #ENTLN    | ITERATE TEST ROUTINE....   |  |
| 17 | 05035 000200 |        | 200           |                            |  |
| 18 | 05037 040040 |        | UFLIM OV36A   |                            |  |
| 19 | 05040 020220 |        | LDA 0, #PBK   | ISETUP STACK IN PAGE 0     |  |
| 20 | 05041 040042 |        | STA 0, #P     | ITO UNDER FLOW ON ANY      |  |
| 21 | 05042 020036 |        | LDA 0, #BREG  | POPI, POPJ, RTN, OR POPB.  |  |
| 22 | 05043 040043 |        | STA 0, #SL    | IINSTRUCTION, THE FAULT    |  |
| 23 | 05044 103710 |        | LDA 0, #OV36A | LOCATION IS OV36A.         |  |
| 24 |              |        | STA 0, #F     |                            |  |
| 25 |              |        | PBR           | PC SHOULD CAUSE UNDERFLOW. |  |
| 26 |              |        | POPJ          | POP FAILED, NO UNDERFLOW.  |  |
| 27 |              |        | ERROR         |                            |  |
| 28 | 05051 000230 | OV36A: | LOOP          |                            |  |
| 29 | 05052 000230 | OV37:  | JSR #ENTLN    | ITERATE TEST ROUTINE....   |  |
| 30 | 05053 000200 |        | 200           |                            |  |
| 31 | 05054 040040 |        | UFLIM OV37A   |                            |  |
| 32 | 05055 040040 |        | LDA 0, #PBK   | ISETUP STACK IN PAGE 0     |  |
| 33 | 05056 020220 |        | STA 0, #P     | ITO UNDER FLOW ON ANY      |  |
| 34 | 05057 040042 |        | LDA 0, #BREG  | POPI, POPJ, RTN, OR POPB.  |  |
| 35 | 05058 020035 |        | STA 0, #SL    | IINSTRUCTION, THE FAULT    |  |
| 36 | 05059 040043 |        | LDA 0, #OV37A | LOCATION IS OV37A.         |  |
| 37 | 05060 040040 |        | STA 0, #F     |                            |  |
| 38 | 05061 064402 |        | JSR #2        |                            |  |
| 39 | 05062 064404 |        | JMP #4        |                            |  |
| 40 | 05063 064404 |        | SAVE 0        |                            |  |
| 41 | 05064 103710 |        | RTN           |                            |  |
| 42 | 05066 127710 |        | ERROR         | PRTN FAILED TO UNDERFLOW.  |  |
| 43 |              |        | LOOP          |                            |  |
| 44 | 05073 000231 | OV37A: | JSR #ENTLN    | ITERATE TEST ROUTINE....   |  |
| 45 |              |        |               |                            |  |
| 46 |              |        |               |                            |  |
| 47 |              |        |               |                            |  |
| 48 |              |        |               |                            |  |
| 49 |              |        |               |                            |  |
| 50 |              |        |               |                            |  |
| 51 |              |        |               |                            |  |
| 52 |              |        |               |                            |  |

|    |              |        |                |                             |  |
|----|--------------|--------|----------------|-----------------------------|--|
| 01 |              |        |                |                             |  |
| 02 |              | OV38:  | SETUP 200      | TEST FOR UNDERFLOW ON POPB. |  |
| 03 | 05074 000230 |        | JSR #ENTLN     | INITIALIZE TEST....         |  |
| 04 | 05075 000200 |        | 200            |                             |  |
| 05 | 05075 020040 |        | UFLIM OV38A    |                             |  |
| 06 | 05077 040040 |        | LDA 0, #PBK    | ISETUP STACK IN PAGE 0      |  |
| 07 | 05077 040040 |        | STA 0, #P      | ITO UNDER FLOW ON ANY       |  |
| 08 | 05100 020220 |        | LDA 0, #BREG   | POPI, POPJ, RTN, OR POPB.   |  |
| 09 | 05101 040042 |        | STA 0, #SL     | IINSTRUCTION, THE FAULT     |  |
| 10 | 05102 020034 |        | LDA 0, #OV38A  | LOCATION IS OV38A.          |  |
| 11 | 05103 040043 |        | STA 0, #F      |                             |  |
| 12 | 05104 064402 |        | JSR #2         |                             |  |
| 13 | 05105 064404 |        | JMP #4         |                             |  |
| 14 | 05106 103710 |        | SAVE 0         |                             |  |
| 15 | 05110 000004 |        | POPB           | POPB FAILED TO UNDERFLOW    |  |
| 16 | 05110 107710 |        | ERROR          |                             |  |
| 17 |              |        | LOOP           |                             |  |
| 18 |              | OV38A: | JSR #ENTLN     | ITERATE TEST ROUTINE....    |  |
| 19 | 05115 000231 | OV39:  | SETUP 200      | INITIALIZE TEST....         |  |
| 20 |              |        | 200            |                             |  |
| 21 | 05115 000230 |        | JSR #ENTLN     |                             |  |
| 22 | 05117 000200 |        | 200            |                             |  |
| 23 |              |        | UFLIM OV39A    |                             |  |
| 24 |              |        | LDA 0, #PBK    | ISETUP STACK IN PAGE 0      |  |
| 25 | 05120 020040 |        | STA 0, #P      | ITO UNDER FLOW ON ANY       |  |
| 26 | 05121 020040 |        | LDA 0, #BREG   | POPI, POPJ, RTN, OR POPB.   |  |
| 27 | 05122 020220 |        | STA 0, #SL     | IINSTRUCTION, THE FAULT     |  |
| 28 | 05123 040042 |        | LDA 0, #OV39A  | LOCATION IS OV39A.          |  |
| 29 | 05124 020033 |        | STA 0, #F      |                             |  |
| 30 | 05125 040043 |        | POP            |                             |  |
| 31 |              |        | RAND           |                             |  |
| 32 | 05126 000234 |        | JSR #ENTLN     | IC(A00)RANDOM #             |  |
| 33 | 05127 105400 |        | INC 0, 1       | TEST THE ACIS FOR           |  |
| 34 | 05130 131400 |        | INC 1, 2       | PROT BEING CHANGED VIA      |  |
| 35 | 05131 177210 |        | POP 3, 5       | UNDERFLOW.                  |  |
| 36 | 05132 100015 | OV39A: | ADCW 0, 1, SNR | IC(A00-2) SHALL BE          |  |
| 37 | 05133 132014 |        | ADCW 1, 2, SZR | IIN ASCENDING ORDER         |  |
| 38 |              |        | ERROR          |                             |  |
| 39 |              |        | LOOP           |                             |  |
| 40 | 05140 000031 |        | JSR #ENTLN     | ITERATE TEST ROUTINE....    |  |
| 41 |              |        |                |                             |  |
| 42 |              |        |                |                             |  |



18077 ECL22

|    |       |               |                          |                     |  |  |  |  |  |
|----|-------|---------------|--------------------------|---------------------|--|--|--|--|--|
| 01 |       |               |                          |                     |  |  |  |  |  |
| 02 | 05141 | 000230        | SETUP 200                | INITIALIZE TEST.... |  |  |  |  |  |
| 03 | 05142 | 000200        | JSR #ENTIN               |                     |  |  |  |  |  |
| 04 | 05143 | 020040-       | 200                      |                     |  |  |  |  |  |
| 05 | 05144 | 040040        | UFLIM OV48A              |                     |  |  |  |  |  |
| 06 | 05145 | 020226        | LOA 0, #PBK              |                     |  |  |  |  |  |
| 07 | 05146 | 040042        | STA 0, #SP               |                     |  |  |  |  |  |
| 08 | 05147 | 020032-       | ITO UNDER FLOW ON ANY    |                     |  |  |  |  |  |
| 09 | 05148 | 040043        | POP, POPJ, RTN, OR POPB, |                     |  |  |  |  |  |
| 10 | 05149 | 020033-       | LOA 0, #BREG             |                     |  |  |  |  |  |
| 11 | 05150 | 040043        | STA 0, #SL               |                     |  |  |  |  |  |
| 12 | 05151 | 000234        | LOA 0, #OV48A            |                     |  |  |  |  |  |
| 13 | 05152 | 100110        | STA 0, #SF               |                     |  |  |  |  |  |
| 14 | 05153 | 177210        | RAND                     |                     |  |  |  |  |  |
| 15 | 05154 | 000230        | JSR #ENTRA               |                     |  |  |  |  |  |
| 16 | 05155 | 000231        | PSH 0, 0                 |                     |  |  |  |  |  |
| 17 | 05156 | 116414 OV40A: | POP 3, 3                 |                     |  |  |  |  |  |
| 18 | 05157 | 000231        | ERROR                    |                     |  |  |  |  |  |
| 19 | 05158 | 000231        | LOOP                     |                     |  |  |  |  |  |
| 20 | 05159 | 000231        | JSR #ENTLO               |                     |  |  |  |  |  |
| 21 | 05160 | 000231        | ITERATE TEST ROUTINE.... |                     |  |  |  |  |  |
| 22 | 05161 | 000230        | SETUP 200                |                     |  |  |  |  |  |
| 23 | 05162 | 000200        | JSR #ENTIN               |                     |  |  |  |  |  |
| 24 | 05163 | 000200        | 200                      |                     |  |  |  |  |  |
| 25 | 05164 | 020040-       | UFLIM OV41Y              |                     |  |  |  |  |  |
| 26 | 05165 | 040040        | LOA 0, #PBK              |                     |  |  |  |  |  |
| 27 | 05166 | 020226        | STA 0, #SP               |                     |  |  |  |  |  |
| 28 | 05167 | 040042        | ITO UNDER FLOW ON ANY    |                     |  |  |  |  |  |
| 29 | 05168 | 020031-       | POP, POPJ, RTN, OR POPB, |                     |  |  |  |  |  |
| 30 | 05169 | 040043        | LOA 0, #BREG             |                     |  |  |  |  |  |
| 31 | 05170 | 020031-       | STA 0, #SL               |                     |  |  |  |  |  |
| 32 | 05171 | 040043        | LOA 0, #OV41Y            |                     |  |  |  |  |  |
| 33 | 05172 | 000234        | STA 0, #SF               |                     |  |  |  |  |  |
| 34 | 05173 | 115040        | RAND                     |                     |  |  |  |  |  |
| 35 | 05174 | 127210        | JSR #ENTRA               |                     |  |  |  |  |  |
| 36 | 05175 | 000230        | MOV 0, 0                 |                     |  |  |  |  |  |
| 37 | 05176 | 000230        | POP 1, 1                 |                     |  |  |  |  |  |
| 38 | 05177 | 000230        | ERROR                    |                     |  |  |  |  |  |
| 39 | 05178 | 000230        | LOOP                     |                     |  |  |  |  |  |
| 40 | 05179 | 000231        | JSR #ENTLO               |                     |  |  |  |  |  |
| 41 | 05180 | 000231        | ITERATE TEST ROUTINE.... |                     |  |  |  |  |  |
| 42 | 05203 | 000230        | SETUP 200                |                     |  |  |  |  |  |
| 43 | 05204 | 000200        | JSR #ENTIN               |                     |  |  |  |  |  |
| 44 | 05205 | 000200        | 200                      |                     |  |  |  |  |  |
| 45 | 05206 | 020040-       | UFLIM OV44A              |                     |  |  |  |  |  |
| 46 | 05207 | 040040        | LOA 0, #PBK              |                     |  |  |  |  |  |
| 47 | 05208 | 020226        | STA 0, #SP               |                     |  |  |  |  |  |
| 48 | 05209 | 040042        | ITO UNDER FLOW ON ANY    |                     |  |  |  |  |  |
| 49 | 05210 | 020032-       | POP, POPJ, RTN, OR POPB, |                     |  |  |  |  |  |
| 50 | 05211 | 020033-       | LOA 0, #BREG             |                     |  |  |  |  |  |
| 51 | 05212 | 040043        | STA 0, #SL               |                     |  |  |  |  |  |
| 52 | 05213 | 101040        | LOA 0, #OV44A            |                     |  |  |  |  |  |
| 53 | 05214 | 103710        | STA 0, #SF               |                     |  |  |  |  |  |
| 54 | 05215 | 117710        | MOV 0, 0                 |                     |  |  |  |  |  |
| 55 | 05216 | 101003        | PSHR                     |                     |  |  |  |  |  |
| 56 | 05217 | 000231        | POPJ                     |                     |  |  |  |  |  |
| 57 | 05218 | 000231        | ERROR                    |                     |  |  |  |  |  |
| 58 | 05219 | 000231        | LOOP                     |                     |  |  |  |  |  |
| 59 | 05220 | 000231        | JSR #ENTLO               |                     |  |  |  |  |  |
| 60 | 05221 | 000231        | ITERATE TEST ROUTINE.... |                     |  |  |  |  |  |

18078 ECL22

|    |       |               |                          |                     |  |  |  |  |  |
|----|-------|---------------|--------------------------|---------------------|--|--|--|--|--|
| 01 |       |               |                          |                     |  |  |  |  |  |
| 02 | 05214 | 000230        | SETUP 200                | INITIALIZE TEST.... |  |  |  |  |  |
| 03 | 05215 | 000200        | JSR #ENTIN               |                     |  |  |  |  |  |
| 04 | 05216 | 000200        | 200                      |                     |  |  |  |  |  |
| 05 | 05217 | 020040-       | UFLIM OV42A              |                     |  |  |  |  |  |
| 06 | 05218 | 040040        | LOA 0, #PBK              |                     |  |  |  |  |  |
| 07 | 05219 | 020226        | STA 0, #SP               |                     |  |  |  |  |  |
| 08 | 05220 | 020226        | ITO UNDER FLOW ON ANY    |                     |  |  |  |  |  |
| 09 | 05221 | 040042        | POP, POPJ, RTN, OR POPB, |                     |  |  |  |  |  |
| 10 | 05222 | 020032-       | LOA 0, #BREG             |                     |  |  |  |  |  |
| 11 | 05223 | 040043        | STA 0, #SL               |                     |  |  |  |  |  |
| 12 | 05224 | 000234        | LOA 0, #OV42A            |                     |  |  |  |  |  |
| 13 | 05225 | 115020        | STA 0, #SF               |                     |  |  |  |  |  |
| 14 | 05226 | 153210        | RAND                     |                     |  |  |  |  |  |
| 15 | 05227 | 000230        | JSR #ENTRA               |                     |  |  |  |  |  |
| 16 | 05228 | 000230        | MOV 0, 0                 |                     |  |  |  |  |  |
| 17 | 05229 | 000230        | POP 2, 2                 |                     |  |  |  |  |  |
| 18 | 05230 | 116415 OV42A: | ERROR                    |                     |  |  |  |  |  |
| 19 | 05231 | 101002        | SUB# 0, 3, SNR           |                     |  |  |  |  |  |
| 20 | 05232 | 000231        | MOV 0, 0, #ZC            |                     |  |  |  |  |  |
| 21 | 05233 | 000231        | ERROR                    |                     |  |  |  |  |  |
| 22 | 05234 | 000231        | LOOP                     |                     |  |  |  |  |  |
| 23 | 05235 | 000231        | JSR #ENTLO               |                     |  |  |  |  |  |
| 24 | 05236 | 000230        | SETUP 200                |                     |  |  |  |  |  |
| 25 | 05237 | 000200        | JSR #ENTIN               |                     |  |  |  |  |  |
| 26 | 05238 | 000200        | 200                      |                     |  |  |  |  |  |
| 27 | 05239 | 020040-       | UFLIM OV43A              |                     |  |  |  |  |  |
| 28 | 05240 | 040040        | LOA 0, #PBK              |                     |  |  |  |  |  |
| 29 | 05241 | 020226        | STA 0, #SP               |                     |  |  |  |  |  |
| 30 | 05242 | 020226        | ITO UNDER FLOW ON ANY    |                     |  |  |  |  |  |
| 31 | 05243 | 040042        | POP, POPJ, RTN, OR POPB, |                     |  |  |  |  |  |
| 32 | 05244 | 020032-       | LOA 0, #BREG             |                     |  |  |  |  |  |
| 33 | 05245 | 040043        | STA 0, #SL               |                     |  |  |  |  |  |
| 34 | 05246 | 101020        | LOA 0, #OV43A            |                     |  |  |  |  |  |
| 35 | 05247 | 103710        | STA 0, #SF               |                     |  |  |  |  |  |
| 36 | 05248 | 117710        | MOV 0, 0                 |                     |  |  |  |  |  |
| 37 | 05249 | 117710        | PSHR                     |                     |  |  |  |  |  |
| 38 | 05250 | 101002        | POPJ                     |                     |  |  |  |  |  |
| 39 | 05251 | 000231        | MOV 0, 0, #ZC            |                     |  |  |  |  |  |
| 40 | 05252 | 000231        | ERROR                    |                     |  |  |  |  |  |
| 41 | 05253 | 000231        | LOOP                     |                     |  |  |  |  |  |
| 42 | 05254 | 000231        | JSR #ENTLO               |                     |  |  |  |  |  |
| 43 | 05255 | 000230        | SETUP 200                |                     |  |  |  |  |  |
| 44 | 05256 | 000200        | JSR #ENTIN               |                     |  |  |  |  |  |
| 45 | 05257 | 000200        | 200                      |                     |  |  |  |  |  |
| 46 | 05258 | 020040-       | UFLIM OV44A              |                     |  |  |  |  |  |
| 47 | 05259 | 040040        | LOA 0, #PBK              |                     |  |  |  |  |  |
| 48 | 05260 | 020226        | STA 0, #SP               |                     |  |  |  |  |  |
| 49 | 05261 | 040042        | ITO UNDER FLOW ON ANY    |                     |  |  |  |  |  |
| 50 | 05262 | 020032-       | POP, POPJ, RTN, OR POPB, |                     |  |  |  |  |  |
| 51 | 05263 | 040043        | LOA 0, #BREG             |                     |  |  |  |  |  |
| 52 | 05264 | 101040        | STA 0, #SL               |                     |  |  |  |  |  |
| 53 | 05265 | 103710        | LOA 0, #OV44A            |                     |  |  |  |  |  |
| 54 | 05266 | 117710        | STA 0, #SF               |                     |  |  |  |  |  |
| 55 | 05267 | 101003        | MOV 0, 0, #SNC           |                     |  |  |  |  |  |
| 56 | 05268 | 000231        | POPJ                     |                     |  |  |  |  |  |
| 57 | 05269 | 000231        | ERROR                    |                     |  |  |  |  |  |
| 58 | 05270 | 000231        | LOOP                     |                     |  |  |  |  |  |
| 59 | 05271 | 000231        | JSR #ENTLO               |                     |  |  |  |  |  |
| 60 | 05272 | 000231        | ITERATE TEST ROUTINE.... |                     |  |  |  |  |  |

```

10079 ECL22
01
02 SETUP 200
03 JSR #ENTIN
04 200
05 UFLIM OV45A
06 LDA 0,SPBK
07 STA 0,SP
08 LDA 0,BREG
09 STA 0,SL
10 LDA 0,OV45A
11 STA 0,SP
12 RAND
13 05314 060234
14 05315 105020
15 05316 103710
16 05317 000000
17 05320 127710
18 05321 051401 OV45A:
19 05322 053402
20 05323 150415
21 05324 101002
22 ERROR
23 LOOP
24 JSR #ENTLO
25 05331 006231
26 05332 006230
27 05333 002020
28 05334 020040-
29 05335 040040
30 05336 020220
31 05337 040042
32 05338 040042
33 05339 040042
34 05340 020024-
35 05341 040043
36 05342 006234
37 05343 110020
38 05344 103710
39 05345 000000
40 05346 127710
41 05347 025774 OV45A:
42 05348 132415
43 05349 132415
44 05351 101002
45 ERROR
46 LOOP
47 JSR #ENTLO
48 05356 006231

10080 ECL22
01
02 INITIALIZE TEST....
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

OV47I
05357 006230
05358 002020
05361 020040-
05362 040040
05363 020220
05364 040042
05365 020023-
05366 040043
05367 005234
05370 105020
05371 103710
05373 107710
05374 106415 OV47A:
05375 101003
05402 006231
05403 006230
05404 000200
05405 020040-
05406 040040
05407 020220
05410 040042
05411 020022-
05412 040043
05413 006234
05414 105020
05415 103710
05417 107710
05420 106415 OV48A:
05421 101002
05426 006231

INITIALIZE TEST....
?SETUP STACK IN PAGE 0
?TO UNDER FLOW ON ANY
?POP,POPJ,RTN, OR POPB.
?INSTRUCTION, THE FAULT
?LOCATION IS OV47A.
?C(AC0)=RANDOM #
?TEST STORAGE OF C(CARRY) AND C(AC0-1)
?FOR NOT CHANGING ON
?UNDERFLOW.
?ITERATE TEST ROUTINE.....

?SETUP 200
?JSR #ENTIN
?200
?UFLIM OV47A
?LDA 0,SPBK
?STA 0,SP
?LOA 0,BREG
?STA 0,SL
?LOA 0,OV47A
?STA 0,SP
?RAND
?JSR #ENTRA
?MOVZ 0,1
?SAVE 0
?POPB
?SUB# 0,1,SNR
?MOV 0,0,SLC
?ERROR
?LOOP
?JSR #ENTLO
?SETUP 200
?JSR #ENTIN
?200
?UFLIM OV48A
?LDA 0,SPBK
?STA 0,SP
?LOA 0,BREG
?STA 0,SL
?LOA 0,OV48A
?STA 0,SP
?RAND
?JSR #ENTRA
?MOVZ 0,1
?SAVE 0
?POPB
?SUB# 0,1,SNR
?MOV 0,0,SLC
?ERROR
?LOOP
?JSR #ENTLO
?ITERATE TEST ROUTINE.....

INITIALIZE TEST....
?SETUP STACK IN PAGE 0
?TO UNDER FLOW ON ANY
?POP,POPJ,RTN, OR POPB.
?INSTRUCTION, THE FAULT
?LOCATION IS OV48A.
?C(AC0)=RANDOM #
?TEST STORAGE OF AC2
?ON UNDERFLOW AND
?C(CARRY) FOR NOT CHANGING.
?ITERATE TEST ROUTINE.....

?SETUP 200
?JSR #ENTIN
?200
?UFLIM OV45A
?LDA 0,SPBK
?STA 0,SP
?LOA 0,BREG
?STA 0,SL
?LOA 0,OV45A
?STA 0,SP
?RAND
?JSR #ENTRA
?MOVZ 0,2
?SAVE 0
?RTN
?LOA 1,-4,3
?SUB# 1,2,SNR
?MOV 0,0,SLC
?ERROR
?LOOP
?JSR #ENTLO
?ITERATE TEST ROUTINE.....

```

```

10001 ECL22
01
02
03 05437 006230
04 05438 000200
05
06 05431 020040-
07 05432 040040
08 05433 020226
09 05434 040042
10 05435 020021-
11 05436 040043
12 05437 004402
13 05440 005440
14 05441 163710
15 05442 000000
16 05443 127710
17 05444 020774
18 05445 026040
19 05446 106554
20
21
22 05453 006231
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

```

```

?INITIALIZE TEST....
?SETUP STACK IN PAGE 0
?TO UNDER FLOW ON ANY
?POP,POPJ,RTN, OR POPB,
?INSTRUCTION, THE FAULT
?LOCATION IS 0V51A.
?SET C(AC3) TO PC+1
?SAVE C
?SAVE C(AC3) IN LAST STACK
RTN
LDA 0,-4
LDA 1,-8
SUBOL# 0,1,SZR
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE....

```

```

01
02 05454 020226
03 05455 040240
04
05 05456 006230
06 05457 000002
07
08 05460 020040-
09 05461 040040
10 05462 020226
11 05463 040042
12 05464 020020-
13 05465 040043
14 05466 020240
15 05467 040040
16 05470 117710
17 05471 026040
18 05472 106554
19
20
21 05477 006231
22 05500 010240
23 05501 020240
24 05502 024227
25 05503 106414
26 05504 000752
27
28
29 05505 006230
30 05506 000200
31
32 05507 020040-
33 05510 040040
34 05511 020226
35 05512 040042
36 05513 020017-
37 05514 040043
38 05515 103210
39
40 05522 105523
41 05523 045524
42
43 05524 006231
44
45

```

```

?INITIALIZE TEST....
?SETUP STACK IN PAGE 0
?TO UNDER FLOW ON ANY
?POP,POPJ,RTN, OR POPB,
?INSTRUCTION, THE FAULT
?LOCATION IS 0V52A.
?SET C(PC) TO C(TEM)
?AND THEN UNDERFLOW.
?C(TEM) SHOULD BE STORED
?ON THE STACK AS THE
?PC ENTRY.
?ITERATE TEST ROUTINE....

```

```

01
02 05525 006230
03 05526 000200
04
05 05527 006230
06 05528 000002
07
08 05531 020040-
09 05534 040040
10 05535 020226
11 05536 040042
12 05537 020020-
13 05540 040043
14 05541 163710
15 05542 000000
16 05543 127710
17 05544 020774
18 05545 026040
19 05546 106554
20
21
22 05553 006231
23
24
25
26
27
28
29 05554 006230
30 05555 000200
31
32 05556 020040-
33 05559 040040
34 05560 020226
35 05561 040042
36 05562 020017-
37 05563 040043
38 05564 103210
39
40 05571 105523
41 05572 045524
42
43 05573 006231
44
45

```

```

?INITIALIZE TEST....
?SETUP STACK IN PAGE 0
?TO UNDER FLOW ON ANY
?POP,POPJ,RTN, OR POPB,
?INSTRUCTION, THE FAULT
?LOCATION IS 0V53A.
?ITERATE TEST ROUTINE....

```

|             |    |       |        |        |                   |                            |  |  |  |
|-------------|----|-------|--------|--------|-------------------|----------------------------|--|--|--|
| 10083 ECL22 | 01 |       |        |        |                   |                            |  |  |  |
|             | 02 | 05520 | 006230 | 0V541  | SETUP 200         |                            |  |  |  |
|             | 03 | 05520 | 006230 |        | JSR #ENTLN        |                            |  |  |  |
|             | 04 | 05520 | 006230 |        | 200               | INITIALIZE TEST....        |  |  |  |
|             | 05 | 05527 | 020226 |        | OVTRP OV548,OV544 |                            |  |  |  |
|             | 06 | 05530 | 040040 |        | LDA R,0BREG       | TEST OVERFLOW OF THE STACK |  |  |  |
|             | 07 | 05531 | 040042 |        | STA 0,SP          | FOR THE CAP INSTRUCTION.   |  |  |  |
|             | 08 | 05532 | 040044 |        | STA 0,SL          |                            |  |  |  |
|             | 09 | 05532 | 040044 |        | STA 0,TO          |                            |  |  |  |
|             | 10 | 05533 | 024405 |        | LDA 1,0           | IF NO OV GO TO OV54A.      |  |  |  |
|             | 11 | 05534 | 040044 |        | STA 1,0           |                            |  |  |  |
|             | 12 | 05535 | 024404 |        | LDA 1,0           | IF OV GO TO OV54B.         |  |  |  |
|             | 13 | 05536 | 040043 |        | STA 1,0           |                            |  |  |  |
|             | 14 | 05537 | 006403 |        | JMP #3            |                            |  |  |  |
|             | 15 | 05540 | 005543 |        | OV54A:            | EXPECT STACK OVERFLOW      |  |  |  |
|             | 16 | 05541 | 005547 |        | OV54B:            | ICHECK MICRO INSTRUCTION   |  |  |  |
|             | 17 | 05542 | 100030 |        | OV54A:            | ITRACK TEST ROUTINE....    |  |  |  |
|             | 18 | 05547 | 006231 |        | OV54B:            | JSR #ENTLN                 |  |  |  |
|             | 19 |       |        |        | 21                |                            |  |  |  |
|             | 20 | 05550 | 006230 | 0V551  | SETUP 200         | INITIALIZE TEST....        |  |  |  |
|             | 21 | 05551 | 006230 |        | JSR #ENTLN        |                            |  |  |  |
|             | 22 | 05552 | 020226 |        | 200               |                            |  |  |  |
|             | 23 | 05553 | 040040 |        | OVTRP OV55C,OV55B | TEST OVERFLOW OF THE STACK |  |  |  |
|             | 24 | 05554 | 040042 |        | LDA R,0BREG       | FOR THE CAP INSTRUCTION.   |  |  |  |
|             | 25 | 05555 | 040044 |        | STA 0,SP          |                            |  |  |  |
|             | 26 | 05555 | 040044 |        | STA 0,SL          |                            |  |  |  |
|             | 27 | 05556 | 024405 |        | STA 0,TO          |                            |  |  |  |
|             | 28 | 05557 | 040044 |        | LDA 1,0           | IF NO OV GO TO OV55B.      |  |  |  |
|             | 29 | 05558 | 024404 |        | STA 1,0           |                            |  |  |  |
|             | 30 | 05559 | 040043 |        | LDA 1,0           | IF OV GO TO OV55C.         |  |  |  |
|             | 31 | 05560 | 024404 |        | STA 1,0           |                            |  |  |  |
|             | 32 | 05561 | 040043 |        | STA 1,SP          |                            |  |  |  |
|             | 33 | 05562 | 006403 |        | JMP #3            |                            |  |  |  |
|             | 34 | 05564 | 005573 |        | OV55A:            | ERROR                      |  |  |  |
|             | 35 | 05566 | 005567 |        | OV55B:            | LOOP                       |  |  |  |
|             | 36 | 05573 | 022040 | 0V55E: | KOP R,0           | ITERATE TEST ROUTINE....   |  |  |  |
|             | 37 | 05574 | 024772 |        | *1                |                            |  |  |  |
|             | 38 | 05575 | 123554 |        | OV55E:            | ERROR                      |  |  |  |
|             | 39 |       |        |        | LDA 0,SP          | FAILED TO TRAP OVERFLOW    |  |  |  |
|             | 40 |       |        |        | LDA 1,OV55A+1     | IC(CO)PC STORED ON STACK.  |  |  |  |
|             | 41 |       |        |        | SUBOL# 1,0,3ZR    | IC(AC0)CORRECT PC          |  |  |  |
|             | 42 |       |        |        | ERROR             | IPC NOT STORED CORRECTLY.  |  |  |  |
|             | 43 |       |        |        | LOOP              |                            |  |  |  |
|             | 44 |       |        |        | JSR #ENTLN        | ITERATE TEST ROUTINE....   |  |  |  |
|             | 45 | 05662 | 006231 |        | 45                |                            |  |  |  |
|             | 46 |       |        |        | 46                |                            |  |  |  |

|             |    |       |        |        |                   |                            |  |  |  |
|-------------|----|-------|--------|--------|-------------------|----------------------------|--|--|--|
| 10084 ECL22 | 01 |       |        |        |                   |                            |  |  |  |
|             | 02 | 05603 | 020226 | 0V561  | LDA R,0BREG       | PLACE A CAP INSTRUCTION    |  |  |  |
|             | 03 | 05604 | 024074 |        | LDA 1,0           | IN THE BUFFER. WHEN        |  |  |  |
|             | 04 | 05605 | 107000 |        | ADD 0,1           | EXECUTED THIS TRAP         |  |  |  |
|             | 05 | 05606 | 044240 | 0V56A: | STA 1,TEM         | SHOULD CAUSE OVERFLOW.     |  |  |  |
|             | 06 | 05607 | 006230 |        | SETUP 2           |                            |  |  |  |
|             | 07 | 05610 | 006002 |        | JSR #ENTLN        | INITIALIZE TEST....        |  |  |  |
|             | 08 | 05611 | 020226 |        | 2                 |                            |  |  |  |
|             | 09 | 05612 | 040048 |        | OVTRP OV56C,OV56B | TEST OVERFLOW OF THE STACK |  |  |  |
|             | 10 | 05613 | 040042 |        | LDA R,0BREG       | FOR THE CAP INSTRUCTION.   |  |  |  |
|             | 11 | 05614 | 040044 |        | STA 0,SP          |                            |  |  |  |
|             | 12 | 05615 | 024405 |        | STA 0,SL          |                            |  |  |  |
|             | 13 | 05616 | 040044 |        | STA 0,TO          |                            |  |  |  |
|             | 14 | 05617 | 024404 |        | LDA 1,0           | IF NO OV GO TO OV56B.      |  |  |  |
|             | 15 | 05618 | 040043 |        | STA 1,0           |                            |  |  |  |
|             | 16 | 05619 | 006403 |        | LDA 1,0           | IF OV GO TO OV56C.         |  |  |  |
|             | 17 | 05620 | 006403 |        | STA 1,SP          |                            |  |  |  |
|             | 18 | 05621 | 006403 |        | JMP #3            |                            |  |  |  |
|             | 19 | 05622 | 006403 |        | OV56B:            | ICHECK THE STORED PC       |  |  |  |
|             | 20 | 05623 | 006403 |        | OV56C:            | ISY THIS TRAP TO           |  |  |  |
|             | 21 | 05624 | 020240 |        | LDA 0,TEM         | BE CORRECT                 |  |  |  |
|             | 22 | 05625 | 042044 |        | STA 0,TO          |                            |  |  |  |
|             | 23 | 05626 | 101021 |        | MOVZ 0,0,8KP      |                            |  |  |  |
|             | 24 | 05627 | 100030 |        | XOP 0,0,0         |                            |  |  |  |
|             | 25 | 05630 | 020777 |        | LDA 0,1           |                            |  |  |  |
|             | 26 | 05631 | 042240 |        | STA 0,TEM         |                            |  |  |  |
|             | 27 | 05632 | 002240 |        | JMP #TEM          |                            |  |  |  |
|             | 28 | 05637 | 006411 | 0V56B: | ERROR             | IFAIL TO OVERFLOW.         |  |  |  |
|             | 29 | 05640 | 020240 |        | JMP OV56D         |                            |  |  |  |
|             | 30 | 05641 | 020240 | 0V56C: | LDA 0,TEM         |                            |  |  |  |
|             | 31 | 05642 | 101003 |        | LDA 1,SP          |                            |  |  |  |
|             | 32 | 05643 | 106414 |        | MOV 0,0,SNC       | IC(CARRY) SHOULD #0        |  |  |  |
|             | 33 | 05644 | 106414 |        | SUB# 0,1,3ZR      | IC(AC0)CORRECT PC          |  |  |  |
|             | 34 | 05650 | 006231 | 0V56D: | ERROR             | IC(AC1)PC STORED           |  |  |  |
|             | 35 | 05651 | 010240 |        | LOOP              |                            |  |  |  |
|             | 36 | 05652 | 020240 |        | JSR #ENTLN        | ITERATE TEST ROUTINE....   |  |  |  |
|             | 37 | 05653 | 024227 |        | ISZ TEM           |                            |  |  |  |
|             | 38 | 05654 | 106414 |        | LDA 0,TEM         |                            |  |  |  |
|             | 39 | 05655 | 000732 |        | LDA 1,BEND        |                            |  |  |  |
|             | 40 | 05656 | 000732 |        | SUB# 0,1,3ZR      |                            |  |  |  |
|             | 41 | 05657 | 000732 |        | JMP OV56A         |                            |  |  |  |
|             | 42 |       |        |        | 42                |                            |  |  |  |
|             | 43 |       |        |        | 43                |                            |  |  |  |
|             | 44 |       |        |        | 44                |                            |  |  |  |
|             | 45 | 05662 | 006231 |        | 45                |                            |  |  |  |
|             | 46 |       |        |        | 46                |                            |  |  |  |

```

10085 ECL22
01
02
03 05656 006230
04 05657 000100
05
06 05658 006221
07 05661 005974
08 05662 020040
09 05663 103710
10 177773
11
12 05665 024040
13 05666 106414
14
15 05673 101081
16
17
18
19
20 05700 006231
21
22
23
24 05703 006221
25 05704 005733
26
27 05705 006234
28 05706 101200
29 05707 024040
30 05710 030857
31 05711 034042
32 05712 100000
33 05713 103000
34 05714 113000
35 05715 024226
36 05716 172433
37 05717 132433
38 05720 000766
39 05721 040402
40 05722 103710
41 000000
42 05724 034040
43 05725 156414
44
45 05732 101081
46
47
48 05737 006231

CHG:
SETUP 100
JSR #ENTIN
100
STACK CHG.0
JSR #1STK
CHG.0
LOA 0,SP
SAVE -5
LOA 1,SP
SUB# 0,1,SZR
ERROR
MOV 0,0,SKP
ERROR
LOOP
JSR #ENTLO

CHG:1
SETUP 400
JSR #ENTIN
400
STACK CGI.2
CGI.2
RAND
JSR #ENTRA
MOV# 0,0
LOA 1,SP
LOA 2,5
LOA 3,5L
ADC 2,3
ADD 1,2
ADD 0,2
LOA 1,0BEC
SUB# 3,2,SNC
SUB# 1,2,SNC
JMP CGI.1+1
STA 0,+2
SAVE 0
LOA 3,SP
SUB# 2,3,SZR
ERROR
JSR #ENTLO

CHG:2
CGI.2:
ERROR
LOOP
JSR #ENTLO

INITIALIZE TEST....
INITIALIZE STACK
IF(AULT ADDRESS IS CHG.0
)THE STACK POINTER SHOULD
)REMAIN UNCHANGED, SP+5
)VIA SAVE AND -5 VIA
)ARGUMENT FOLLOWING THE
)SAVE.
)C(AC1)=NEW STACK POINTER.
)C(AC0)=CORRECT STACK POINTER.
)ITERATE TEST ROUTINE....

CHG:3
SETUP 100
JSR #ENTIN
100
STACK CG2.1
CG2.1
LOA 0,SP
LOA 1,5
ADD 0,1
SAVE 7H
LOA 2,FP
SUB# 1,2,SNR
SUB# 1,3,SZR
ERROR
MOV 0,0,SKP
ERROR
LOOP
JSR #ENTLO

CHG:4:
SETUP 100
JSR #ENTIN
100
RAND
JSR #ENTRA
MOV 0,2
MOV #1,5000
DIVS 0,3
SUB# 2,3,SZR
ERROR
LOOP
JSR #ENTLO

CHG:5
SETUP 100
JSR #ENTIN
100
RAND
JSR #ENTRA
MOVZ 0,2
DIVS 0,10,SNC
MOV 0,10,SNC
ERROR
LOOP
JSR #ENTLO

INITIALIZE TEST....
INITIALIZE STACK
)C(AC3)=SAVE RESULT
)C(AC2)=FRAME POINTER
)C(AC1)=CORRECT FRAME POINTER.
)ITERATE TEST ROUTINE....

INITIALIZE TEST....
INITIALIZE TEST....
)C(AC0)=RANDOM #
)TEST TO INSURE NO AC2 IS
)FIND CHANGED ON SIGNED DIVIDE
)OVERFLOW.
)ITERATE TEST ROUTINE....

INITIALIZE TEST....
INITIALIZE TEST....
)C(AC0)=RANDOM #
)C(CARRY) SHOULD SET ON
)SIGNED DIVIDE AND OVERFLOW.
)ITERATE TEST ROUTINE....

```

```

10007 ECL22
01
02
03 06015 006230
04 06016 000005
05 06017 102400
06 06020 042226
07
08 06021 000234
09 06022 024027
10 06023 030226
11 06024 140400
12 06025 150420
13 06026 133710
14 06027 030227
15 06030 175005
16 06031 101002
17
18
19 06036 006231
20
21 06037 006230
22 06040 000100
24 06041 102000
25 06042 042226
26
27 06043 006234
28 06044 024227
29 06045 030226
30 06046 140400
31 06047 150420
32 06050 133710
33 06051 030227
34 06052 174015
35 06053 101003
36
37
38 06060 006231

CHG5:
INITIALIZE TEST....
SETUP 5
JSR #ENTIN
5
STA 0,0
STA 0,0BBEG
RAND
JSR #ENTRA
LDA 1,0BND
LDA 2,0BBEG
SUB 2,1
INCL 2,3
BLK
LDA 3,0BEND
MOV 3,3,SNR
MOV 0,0,0ZC
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE....

CHG6:
INITIALIZE TEST....
SETUP 100
JSR #ENTIN
100
ADC 0,0
STA 0,0BBEG
RAND
JSR #ENTRA
LDA 1,0BND
LDA 2,0BBEG
SUB 2,1
INCL 2,3
BLK
LDA 3,0BEND
MOV 3,3,SNR
MOV 0,0,0NC
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE....

CHG7:
INITIALIZE TEST....
SETUP 5
JSR #ENTIN
5
LDA 1,0BND
LDA 2,0BBEG
SUB 2,1
MOV 1,0
ADD 2,0
MOV 2,3
BLK
SUB# 0,2,SNR
SUB# 0,3,0ZR
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE....

CHG8:
INITIALIZE TEST....
SETUP 2
JSR #ENTIN
2
LDA 2,0BBEG
LDA 1,0BND
SUB 2,1
MOV 2,3
BLK
MOV 1,1,SNR
MOV 0,0,0NC
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE....

```

10008 ECL22

```

01
02
03 06061 006230
04 06062 000005
05 06063 024227
06 06064 030226
07 06065 140400
08 06066 121000
09 06067 143000
10 06070 150400
11 06071 133710
12 06072 112415
13 06073 116414
14
15
16 06100 006231
17
18
19 06101 006230
20 06102 000002
21 06103 030226
22 06104 024227
23 06105 140400
24 06106 150400
25 06107 133710
26 06110 150005
27 06111 101003
28
29
30 06116 006231

```

```

10000 ECL22
01
02 06117 060230          SETUP 200
03 06120 060200          JSR #ENTN
04 06121 064210          LDA 0,RAN
05 06122 020270          LDA 0,RAN
06 06123 125004          MOV 1,1,SZR
07 06124 020277          LDA 0,RAN
08 06125 020277          STA 0,RAN
09 06126 040277          STA 0,RAN
10 06127 115300          MOV# 0,3
11 06128 030227          LDA 2,BEND
12 06130 024226          LDA 1,BEG
13 06132 132400          SUB 1,2
14 06133 156422          SUBZ 2,3,SZC
15 06134 000777          JMP #1
16 06135 157000          ADD 2,3
17 06136 142422          SUBZ 2,0,SZC
18 06137 000777          JMP #1
19 06138 113500          AOD 0,2
20 06139 113500          SUB# 5,2,SZC
21 06141 172432          XCH 2,3
22 06142 154710          LDA 1,BBEG
23 06143 024226          ADD 1,2
24 06144 133000          ADD 1,3
25 06145 137000          LDA 1,BEND
26 06146 024227          SUB 3,1
27 06147 156400          STA 1,BANSIZ
28 06150 044313          STA 1,BANT0
29 06151 044315          STA 2,T00
30 06152 050317          STA 3,FROM
31 06153 054316          STA 3,BANT1
32 06154 054314          BRND
33
34 06155 060235          JSR #ENTR0
35 06156 042314          STA 0,BANT1
36 06157 010314          ISZ BANT1
37 06158 014315          DSZ BANT0
38 06159 014315          JMP CG9.1
39 06161 060774
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62 06162 030316          CG9.2: LDA 2,FROM
63 06163 034317          LDA 3,T00
64 06164 024313          LDA 1,BANSIZ
65 06165 133710          BLM
66
67 06166 020277          CG9.3: LDA 0,SRAN
68 06167 040277          STA 0,RAN
69 06170 020317          LDA 0,T00
70 06171 040314          STA 0,BANSIZ
71 06172 020313          LDA 0,BANT0
72 06173 040315          STA 0,BANT0
73
74 06174 060235          CG9.4: BRND
75 06175 020314          JSR #ENTR0
76 06176 1P6414          LDA 1,BANT1
77 06177 000406          SUB# 0,1,SZR
78 06178 000406          JMP CG9.6
79 06179 010314          CG9.5: ISZ BANT1
80 06201 014315          ISZ BANT0
81 06202 060772          JMP CG9.4
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

INITIALIZE TEST....  
WITH THE BLM INSTRUCTION,  
TRANSFER A RANDOM BLOCK OF  
DATA, THE FROM, TO ADDRESS  
ARE ALSO RANDOM.  
SELECT A RANDOM FROM  
AND TO ADDRESS MODULO  
THE BUFFER SIZE.  
MAKE THE FROM ADDRESS  
GREATER THEN TO ADDRESS.

MOVE THE DATA.....  
PRECONSTRUCT RANDOM FOR  
TESTING.

ITERATE TEST ROUTINE.....

ERROR.....

FROM TO WORD ORIG C(AC)"

FROM

TO

WORD

FROM TO

WORD ORIG C(AC)"

FROM

TO

WORD

FROM TO

WORD ORIG C(AC)"

```

18091 ECL22
01
02 SETUP 100
03 JSR #ENTIN
04 RAND
05
06 CG10A:
07 DTRP CG10A,CG10A
08 LDA 0,BREG
09 STA 0,SP
10 STA 0,S
11 STA 0,T0
12 LDA 1,+5
13 STA 1,+0
14 LDA 1,+4
15 STA 1,SP
16 JMP +3
17
18 CG10A:
19 XOP 0,1,0
20 LDA 0,7,2
21 LDA 1,7,3
22 SUB# 0,2,SNR
23 SUB# 1,3,SNR
24 ERROR
25 LOOP
26
27 CG11A:
28 SETUP 100
29 JSR #ENTIN
30 RAND
31
32 CG11B:
33 DTRP CG11B,CG11B
34 LDA 0,BREG
35 STA 0,SP
36 STA 0,S
37 STA 0,T0
38 LDA 1,+5
39 STA 1,+0
40 LDA 1,+4
41 STA 1,SP
42 JMP +3
43 CG11B
44 RAND
45 JSR #ENTRA
46 INC 0,1
47 XOP 0,1,0
48 LDA 2,5,2
49 LDA 3,5,3
50 SUB# 0,2,SNR
51 SUB# 1,3,SNR
52 ERROR
53 LOOP
54 JSR #ENTLO
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```



10093 ECL22

```

01
02 SETUP 100
03 JSR #ENTIN
04 RAND
05 RAND
06 (ACB)RANDOM #
07 DIVIDE A NEGATIVE NUMBER
08 DIV *1. QUOTIENT SHOULD
09 BE EQUAL ORIGINAL NUMBER,
10 OVERFLOW SHOULD NOT BE
11 POSSIBLE.
12 (REMAINDER SHOULD BE ZERO.
13 ITERATE TEST ROUTINE....
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

```

10094 ECL22

```

01
02 SETUP 100
03 JSR #ENTIN
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

```

```

10095 ECL22
01 CG19: SETUP 100
02 JSR #ENTIN
03 100
04 MOV 0,0,SKP
05 MOVC 0,0
06 LDA 0,*,*1
07 XCT 0
08 MOV 0,0, SZC
09 ERROR
10 LOOP
11 JSR #ENTLO
12
13 CG20: SETUP 100
14 JSR #ENTIN
15 100
16 ZERD AC
17 SUB 0,0
18 SUB 1,1
19 SUB 2,2
20 SUB 3,3
21 LDA 0,*,*1
22 XCT 0
23 MOV 1,1,SNR
24 ADD #2,3, SZR
25 ERROR
26 LOOP
27 JSR #ENTLO
28
29 CG21: SETUP 100
30 JSR #ENTIN
31 100
32 ADC 0,0,SKP
33 0
34 LDA 0,*,*2
35 XCT 0
36 ERROR
37 JSR #ENTLO
38
39 CG22: SETUP 100
40 JSR #ENTIN
41 100
42 LDA 0,*,*1
43 SKPDZ CPU
44 ERROR
45 XCT 0
46 MOV 0,2,SNR
47 ERROR
48 LOOP
49 JSR #ENTLO
50
10096 ECL22
01 CG23: SETUP 100
02 JSR #ENTIN
03 100
04 JSR #*2
05 LDA 0,*,*2
06 XCT 0
07 ERROR
08 LOOP
09 JSR #ENTLO
10
11 CG24: SETUP 100
12 JSR #ENTIN
13 100
14 LDA 0,*,*1
15 JSR #*1
16 TMC 3,2
17 XCT 0
18 ADD #2,3, SZR
19 ERROR
20 LOOP
21 JSR #ENTLO
22
23 CG25: SETUP 100
24 JSR #ENTIN
25 100
26 LDA 0,*,*1
27 XCT 0
28 SUB #0,1, SZR
29 ERROR
30 LOOP
31 JSR #ENTLO
32
33 CG26: SETUP 100
34 JSR #ENTIN
35 100
36 RAND
37 JSR #ENTRA
38 MOV 0,1
39 MOV 1,2,SKP
40 DHXL 4,0
41 LDA 0,*,*1
42 XCT 0
43 SUB #0,2,SNR
44 MOV 1,1, SZR
45 ERROR
46 LOOP
47 JSR #ENTLO
48
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

10897 ECL22

```

01 06637 002401      JMP 0,+1
02 06640 007027      END
03
04
05
06 06641 054310 PMULI
07 06642 034014+    STA 3,PMULR
08 06643 125203    LDA 3,PMULR
09 06644 101201    MOV# 1,1,SNC
10 06645 143220    MOV# 0,0,SKP
11 06646 175404    ADD# 2,0
12 06647 000774    INC 3,3,5ZR
13 06648 125260    JMP ""
14 06649 000774    MOV# 1,1
15 06650 040304    STA 0,0,SKP
16 06651 044305    STA 1,0K1
17 06652 050306    STA 2,0K2
18 06653 002310    JMP PMULR
19
20 06654 054311 MDCKI
21 06655 034304    STA 3,MDRET
22 06656 102414    LDA 3,MDRET
23 06657 102414    SUB# 3,0,5ZR
24 06658 000400    JMP MDCKI
25 06659 102414    LDA 3,0K1
26 06660 102414    SUB# 3,1,5ZR
27 06661 034305    STA 0,0,SKP
28 06662 034306    STA 1,0K1
29 06663 172414    SUR# 3,2,5ZR
30 06664 010311 MDCKI
31 06665 002311    JMP MDRET
32
33 06670 054312 PDIV1
34 06671 142432    STA 3,PDIVR
35 06672 000411    SUB# 2,0,5ZC
36 06673 034014+    JMP PDIV2
37 06674 125120    LDA 3,PMULR
38 06675 101100    MOV# 1,1
39 06676 142412    SUR# 2,0,5ZC
40 06677 142400    SUB 2,0
41 06678 125100    INC 1,1
42 06679 175404    INC 3,3,5ZR
43 06680 000773    JMP PDIV1
44 06681 040304    STA 0,0,SKP
45 06682 044305    STA 1,0K1
46 06683 050306    STA 2,0K2
47 06684 002312    JMP PDIVR
48
49 06670 054311 MDCKI
50 06671 034271    STA 3,MDRET
51 06672 101003    LDA 3,MDRET
52 06673 175400    MOV# 0,0,1,SNC
53 06674 175213    INC 3,3
54 06675 000752    MOV# 3,3,SNC
55 06676 000752    JMP MDCKI
56 06677 101003    MOV# 0,0,1,SNC
57 06678 000740    JMP MDCKI+1
58 06679 002311    JMP MDRET
59

```

10898 ECL22

```

01
02
03 06720 040304 EDIVI
04 06721 102400    STA 0,0,SKP
05 06722 102400    SURC 0,0
06 06723 102400    MOV# 1,1,5ZC
07 06724 101001    ADC 0,0
08 06725 040304    MOV# 0,0,SKP
09 06726 054307    STA 0,0,SKP
10 06727 050306    STA 3,0K3
11 06728 044305    STA 2,0K2
12 06729 044305    STA 1,0K1
13 06730 155102    MOV# 2,3,SZC
14 06731 150400    NEG 2,2
15 06732 175500    SUB# 3,3
16 06733 175120    MOV# 3,3
17 06734 101113    MOV# 0,0,SNC
18 06735 000400    JMP PSDIV1
19 06736 000400    INC 3,3
20 06737 175400    NEG 1,1,5ZR
21 06738 102404    COM# 0,0,SKP
22 06739 102404    NEG# 0,0
23 06740 142432    SUR# 2,0,5ZC
24 06741 000415    JMP PSDIV4
25 06742 034014+    STA 3,0
26 06743 125100    LDA 3,PMULR
27 06744 125100    MOV# 1,1
28 06745 142412    SUR# 2,0,5ZC
29 06746 142400    SUB 2,0
30 06747 102400    INC 1,1
31 06748 175404    INC 3,3,5ZR
32 06749 000773    JMP PSDIV2+3
33 06750 030306    LDA 2,0K2
34 06751 125113    MOV# 1,1,SNC
35 06752 000404    JMP +4
36 06753 175500    SUR# 3,3
37 06754 054271    STA 3,PMULR
38 06755 002307    JMP 0K3
39 06756 034013+    LDA 3,0
40 06757 175203    MOV# 3,3,SNC
41 06758 174001    COM# 3,3,SKP
42 06759 102400    NEG 0,0
43 06760 175203    MOV# 3,3,SNC
44 06761 124400    NEG 1,1
45 06762 175500    SUR# 3,3
46 06763 054271    STA 3,PMULR
47 06764 040304    STA 0,0,SKP
48 06765 044305    STA 1,0K1
49 06766 050306    STA 2,0K2
50 06767 002307    JMP 0K3
51

```

```

!SIGN EXTEND DIVIDE
!SOFTWARE UNSIGNED MULTIPLY
!SOFTWARE SIGNED DIVIDE,
!SAVE ORIG NUMBERS,
!CHECK DIVISOR SIGN
!FORM ABS VALUE
!SAVE SIGN IN C(ACS)
!BIT 14
!DIVIDENT POSITIVE,SAVE
!IN BIT 15
!FORM DIVIDENT ABS
!VALUE
!DIVIDE OVERFLOW...
!FLAG WORD FOR SIGNS
!UNSIGNED DIVIDE SUBROUTINE
!C(CARRY)=1, DIVIDE ERROR
!GET FLAG WORD
!TEST REMAINDER SIGN
!POSITIVE
!NEGATIVE
!TEST QUOTIENT SIGN
!NEGATIVE
!ZERO C(CARRY),
!STORE RESULTS IN
!C(0K0-2)

```

10897 ECL22

```

01
02
03 06637 002401      JMP 0,+1
04
05
06 06641 054310 PMULI
07 06642 034014+    STA 3,PMULR
08 06643 125203    LDA 3,PMULR
09 06644 101201    MOV# 1,1,SNC
10 06645 143220    MOV# 0,0,SKP
11 06646 175404    ADD# 2,0
12 06647 000774    INC 3,3,5ZR
13 06648 125260    JMP ""
14 06649 000774    MOV# 1,1
15 06650 040304    STA 0,0,SKP
16 06651 044305    STA 1,0K1
17 06652 050306    STA 2,0K2
18 06653 002310    JMP PMULR
19
20 06654 054311 MDCKI
21 06655 034304    STA 3,MDRET
22 06656 102414    LDA 3,MDRET
23 06657 102414    SUB# 3,0,5ZR
24 06658 000400    JMP MDCKI
25 06659 102414    LDA 3,0K1
26 06660 102414    SUB# 3,1,5ZR
27 06661 034305    STA 0,0,SKP
28 06662 034306    STA 1,0K1
29 06663 172414    SUR# 3,2,5ZR
30 06664 010311 MDCKI
31 06665 002311    JMP MDRET
32
33 06670 054312 PDIV1
34 06671 142432    STA 3,PDIVR
35 06672 000411    SUB# 2,0,5ZC
36 06673 034014+    JMP PDIV2
37 06674 125120    LDA 3,PMULR
38 06675 101100    MOV# 1,1
39 06676 142412    SUR# 2,0,5ZC
40 06677 142400    SUB 2,0
41 06678 125100    INC 1,1
42 06679 175404    INC 3,3,5ZR
43 06680 000773    JMP PDIV1
44 06681 040304    STA 0,0,SKP
45 06682 044305    STA 1,0K1
46 06683 050306    STA 2,0K2
47 06684 002312    JMP PDIVR
48
49 06670 054311 MDCKI
50 06671 034271    STA 3,MDRET
51 06672 101003    LDA 3,MDRET
52 06673 175400    MOV# 0,0,1,SNC
53 06674 175213    INC 3,3
54 06675 000752    MOV# 3,3,SNC
55 06676 000752    JMP MDCKI
56 06677 101003    MOV# 0,0,1,SNC
57 06678 000740    JMP MDCKI+1
58 06679 002311    JMP MDRET
59

```

```

!CHECK PRESENT C(AC0=2)
!WITH C(0K0=2)
!RETURN +1 IF NO ERROR,
!+2 IF ERROR.
!SOFTWARE DIVIDE
!C(AC0-1)/C(AC2)
!C(AC0)=REMAINDER
!C(AC1)=QUOTIENT
!STORE RESULT IN 0K
!BLOCK
!CHECK SIGNED DIVIDE.
!C(CARRY) IS WRONG
!NO 0,V, CHECK RESULT.
!0,V, DON'T CHECK.

```

10099 ECL22

```

01
02
03 07000 054307 PSKUL1 STA 3,0K3
04 07001 040257 STA 0,1TEM
05 07002 102400 SUBC 0,0
06 07003 176400 SUB 3,3
07 07004 125112 MOVLM 1,1,SZC
08 07005 157000 ADD 2,3
09 07006 151112 MOVLM 2,2,SZC
10 07007 137000 ADD 1,3
11 07010 054013- STA 3,00
12 07011 004630 JSR PAUL
13 07012 034013- LDA 3,00
14 07013 162400 SUM 3,0
15 07014 034257 LDA 3,1TEM
16 07015 175113 MOVLM 3,3,SNC
17 07016 152401 SUB 2,2,SKP
18 07017 152000 ADC 2,2
19 07020 167022 ADDZ 3,1,SZC
20 07021 151400 INC 2,2
21 07022 143000 ADD 2,0
22 07023 030306 LDA 2,0K2
23 07024 040304 STA 0,0K0
24 07025 044305 STA 1,0K1
25 07026 002307 JMP 00K3

```

SOFTWARE SIGNED MULTIPLY

UNSIGNED MULTIPLY

10100 ECL22

```

01
02 07027 014204 END:
03 07030 000437 DSZ
04 07031 010203 ISZ
05 07032 101001 MOV
06 07033 063077 MALT
07 07034 020205 LDA
08 07035 040204 STA
09
10 07036 000477 READS
11 07037 101112 MOVLM
12 07040 000403 JMP
13 07041 122470 ELDA
14
15 07043 143770 ANDI
16
17 07045 101004 MOV
18 07046 000406 JMP
19
20 07047 006200 JSR
21 07050 001333 PASHES
22 07051 125020 MOVZ
23 07052 024203 LDA
24 07053 006217 JSR
25
26 07054 034010-PSCK11 LOA
27 07055 021400 LOA
28 07056 101005 MOV
29 07057 000410 JMP
30 07060 013403 DSZ
31 07061 000400 JMP
32 07062 002677 IORST
33 07063 021403 LOA
34 07064 033404 LDA
35 07065 041776 STA
36 07066 001400 JMP
37 07067 010407 DMOR:
38 07070 002403 JMP
39 07071 002401 JMP
40 07072 000562 START-1
41 07073 000503 START
42
43
44

```

```

INTERNAL COUNT DONE?
INDEP, LOOP BACK
IYEP, BUMP PASS COUNT
IPASS CNT > 05K
IRESTORE INTERNAL COUNT
ICHECK SWITCH 4
IIF SET, DO NOT
IPRINT PASS COUNT
IPRINT PASS COUNT
I STANDARD DTOS RETURN
IAUTO MODE?
IN0, RE=DO TEST
IYES, PASS COUNT ZERO?
IN0, BACK TO TEST
IYES, BACK TO DTOS
IEND OF TEST
IRESTART TEST...
IOR START CAT THEN TEST

```

\*\*\* END OF TEST ROUTINES \*\*\*

```

18101 ECL22
01 / *****EGGS & DIRT DATA BLOCKS*****
02 EGGS1
03 05 07074 000000 AUTO: 0
04 06 07075 000000 DEV: 0
05 07 07076 000000 CATSW: 0
06 08 07077 000000 PCNT: 0
07 09 07100 000000 RTN: 0
08 10 07101 000000 SREG: 0
09 11 07102 000100 VCTAB: .BLK 100
10 12 07202 007202 PRGEND: PRGEND
11 13
12 14 07203 047503 .TXT /COPYRIGHT (C) DGC,1974,76,77
13 15
14 16 04522
15 17 04107
16 18 020124
17 19 041050
18 20 020051
19 21 043504
20 22 026103
21 23 034461
22 24 033454
23 25 032067
24 26 033454
25 27 020066
26 28 033467
27 29 07221 046101 ALL RIGHTS RESERVED/
28 30
29 31 020114
30 32 044522
31 33 044107
32 34 051524
33 35 051040
34 36 051505
35 37 051105
36 38 04520
37 39 020104
38 40
39 41 07233 141705 DIRT: .TXT !ECLIPSE23:
40 42 144714
41 43 051520
42 44 131305
43 45 031662
44 46 000000
45 47 07241 000000
46 48 07242 000000
47 49 07243 173772
48 50 07244 000000
49 51 07245 000000
50 52 07246 000000
51 53 07247 000000
52 54 07250 000000
53 55
54 56
55 57 .END DT08B
56 58 00013-000000
57 59 17760
58 60 006655

```

```

0102 ECL22
01 006720
02 105522
03 003471
04 005444
05 005420
06 003374
07 003347
08 003321
09 003276
10 003255
11 003233
12 003205
13 003160
14 003132
15 003115
16 003075
17 003051
18 003030
19 003027
20 103710
21 177700
22 000005
23 000012
24 104133
25 005400
26 003534
27 107710
28 000274
29 000007
30 000001
31 000002
32 000003
33 000004
34 000005
35 003240
36 177770
37 000212
38 104400
39 022000
40 000050
41 000011
42 033031
43 000144
44 000100
45 177740
46 000010
47 000140
48 000377
49 000200
50 000400
51 001777

```

\*\*\*\*\* TOTAL ERRORS, 00000 PASS 1 ERRORS



0105 ECL22

CG9 4 006174 90713 90720 30/50 38/57 55/04 50/21 51/08 52/07 53/04 53/20 54/17  
 CG9 5 006202 90718 90725 30/50 38/57 55/04 50/21 51/08 52/07 53/04 53/20 54/17  
 CG9 6 006230 90725 90732 26/60 31/01 56/05 51/15 52/14 53/13 54/12 55/11 56/10  
 CHA1 001277 30730 30737 30/50 38/57 31/01 26/60 56/05 51/15 52/14 53/13 54/12 55/11 56/10  
 CHA2 001306 30733 30740 30/50 38/57 30/50 38/57 30/50 38/57 30/50 38/57 30/50 38/57 30/50 38/57  
 CHA3 001314 30740 30747 30/50 38/57 30/50 38/57 30/50 38/57 30/50 38/57 30/50 38/57 30/50 38/57  
 CHA4 001325 30727 30734 30/50 38/57 30/50 38/57 30/50 38/57 30/50 38/57 30/50 38/57 30/50 38/57  
 CHA5 001356 16/44 30/53 30/50 38/57 30/50 38/57 30/50 38/57 30/50 38/57 30/50 38/57 30/50 38/57  
 CHG 005856 55/02 56/01 57/00 58/00 59/00 60/00 61/00 62/00 63/00 64/00 65/00 66/00 67/00 68/00 69/00 70/00  
 CHG1 005701 85/20 86/20 87/20 88/20 89/20 90/20 91/20 92/20 93/20 94/20 95/20 96/20 97/20 98/20 99/20  
 CHG2 005740 86/22 87/22 88/22 89/22 90/22 91/22 92/22 93/22 94/22 95/22 96/22 97/22 98/22 99/22  
 CHG3 005766 86/22 87/22 88/22 89/22 90/22 91/22 92/22 93/22 94/22 95/22 96/22 97/22 98/22 99/22  
 CHG4 006082 86/25 87/25 88/25 89/25 90/25 91/25 92/25 93/25 94/25 95/25 96/25 97/25 98/25 99/25  
 CHG5 006015 87/22 88/22 89/22 90/22 91/22 92/22 93/22 94/22 95/22 96/22 97/22 98/22 99/22  
 CHG6 006037 88/21 89/21 90/21 91/21 92/21 93/21 94/21 95/21 96/21 97/21 98/21 99/21  
 CHG7 006061 88/18 89/18 90/18 91/18 92/18 93/18 94/18 95/18 96/18 97/18 98/18 99/18  
 CHG8 006101 89/02 90/02 91/02 92/02 93/02 94/02 95/02 96/02 97/02 98/02 99/02  
 CHG9 006117 89/02 90/02 91/02 92/02 93/02 94/02 95/02 96/02 97/02 98/02 99/02  
 CHG0 006574 85/07 86/07 87/07 88/07 89/07 90/07 91/07 92/07 93/07 94/07 95/07 96/07 97/07 98/07 99/07  
 CHORZ 000254 16/45 17/45 18/45 19/45 20/45 21/45 22/45 23/45 24/45 25/45 26/45 27/45 28/45 29/45 30/45  
 CHR3V 001332 16/41 17/41 18/41 19/41 20/41 21/41 22/41 23/41 24/41 25/41 26/41 27/41 28/41 29/41 30/41  
 CRY 000251 101/06 102/06 103/06 104/06 105/06 106/06 107/06 108/06 109/06 110/06 111/06 112/06 113/06 114/06 115/06  
 DEV 007075 15/05 16/05 17/05 18/05 19/05 20/05 21/05 22/05 23/05 24/05 25/05 26/05 27/05 28/05 29/05 30/05  
 DIRT 007233 15/05 16/05 17/05 18/05 19/05 20/05 21/05 22/05 23/05 24/05 25/05 26/05 27/05 28/05 29/05 30/05  
 DIVH1 001615 100/29 101/29 102/29 103/29 104/29 105/29 106/29 107/29 108/29 109/29 110/29 111/29 112/29 113/29 114/29 115/29  
 DOMOR 007067 15/45 16/45 17/45 18/45 19/45 20/45 21/45 22/45 23/45 24/45 25/45 26/45 27/45 28/45 29/45 30/45  
 DTOSB 000200 93/24 94/24 95/24 96/24 97/24 98/24 99/24  
 EDIV 006720 93/24 94/24 95/24 96/24 97/24 98/24 99/24  
 EDIVM 001515 33/50 34/50 35/50 36/50 37/50 38/50 39/50 40/50 41/50 42/50 43/50 44/50 45/50 46/50 47/50 48/50 49/50 50/50  
 EGG5 000704 15/16 16/16 17/16 18/16 19/16 20/16 21/16 22/16 23/16 24/16 25/16 26/16 27/16 28/16 29/16 30/16  
 END 007027 97/03 98/03 99/03 100/03 101/03 102/03 103/03 104/03 105/03 106/03 107/03 108/03 109/03 110/03 111/03 112/03 113/03 114/03 115/03  
 ENTB 000233 16/24 17/24 18/24 19/24 20/24 21/24 22/24 23/24 24/24 25/24 26/24 27/24 28/24 29/24 30/24  
 ENTER 000232 16/23 17/23 18/23 19/23 20/23 21/23 22/23 23/23 24/23 25/23 26/23 27/23 28/23 29/23 30/23

0106 ECL22

ENTLO 000231 16/22 17/22 18/22 19/22 20/22 21/22 22/22 23/22 24/22 25/22 26/22 27/22 28/22 29/22 30/22  
 ENTRA 000234 16/25 17/25 18/25 19/25 20/25 21/25 22/25 23/25 24/25 25/25 26/25 27/25 28/25 29/25 30/25  
 ENTR0 000235 16/26 17/26 18/26 19/26 20/26 21/26 22/26 23/26 24/26 25/26 26/26 27/26 28/26 29/26 30/26  
 EPC 000252 16/42 17/42 18/42 19/42 20/42 21/42 22/42 23/42 24/42 25/42 26/42 27/42 28/42 29/42 30/42  
 ERR 001020 25/31 26/31 27/31 28/31 29/31 30/31  
 ERR1 001055 26/03 27/03 28/03 29/03 30/03  
 ERR2 001071 26/11 27/11 28/11 29/11 30/11  
 ERR3 001074 26/07 27/07 28/07 29/07 30/07  
 ERR4 001110 25/25 26/25 27/25 28/25 29/25 30/25  
 ERRA 001003 16/24 17/24 18/24 19/24 20/24 21/24 22/24 23/24 24/24 25/24 26/24 27/24 28/24 29/24 30/24  
 ERB 001014 15/56 16/56 17/56 18/56 19/56 20/56 21/56 22/56 23/56 24/56 25/56 26/56 27/56 28/56 29/56 30/56  
 ERRET 000212 MC 16/22 17/22 18/22 19/22 20/22 21/22 22/22 23/22 24/22 25/22 26/22 27/22 28/22 29/22 30/22  
 ERROR 000020 MC 39/41 40/41 41/41 42/41 43/41 44/41 45/41 46/41 47/41 48/41 49/41 50/41 51/41 52/41 53/41 54/41 55/41 56/41 57/41 58/41 59/41 60/41

0107 ECL22

ENTIN 000230 16/21 17/21 18/21 19/21 20/21 21/21 22/21 23/21 24/21 25/21 26/21 27/21 28/21 29/21 30/21  
 ENTRA 000234 16/25 17/25 18/25 19/25 20/25 21/25 22/25 23/25 24/25 25/25 26/25 27/25 28/25 29/25 30/25  
 ENTR0 000235 16/26 17/26 18/26 19/26 20/26 21/26 22/26 23/26 24/26 25/26 26/26 27/26 28/26 29/26 30/26  
 EPC 000252 16/42 17/42 18/42 19/42 20/42 21/42 22/42 23/42 24/42 25/42 26/42 27/42 28/42 29/42 30/42  
 ERR 001020 25/31 26/31 27/31 28/31 29/31 30/31  
 ERR1 001055 26/03 27/03 28/03 29/03 30/03  
 ERR2 001071 26/11 27/11 28/11 29/11 30/11  
 ERR3 001074 26/07 27/07 28/07 29/07 30/07  
 ERR4 001110 25/25 26/25 27/25 28/25 29/25 30/25  
 ERRA 001003 16/24 17/24 18/24 19/24 20/24 21/24 22/24 23/24 24/24 25/24 26/24 27/24 28/24 29/24 30/24  
 ERB 001014 15/56 16/56 17/56 18/56 19/56 20/56 21/56 22/56 23/56 24/56 25/56 26/56 27/56 28/56 29/56 30/56  
 ERRET 000212 MC 16/22 17/22 18/22 19/22 20/22 21/22 22/22 23/22 24/22 25/22 26/22 27/22 28/22 29/22 30/22  
 ERROR 000020 MC 39/41 40/41 41/41 42/41 43/41 44/41 45/41 46/41 47/41 48/41 49/41 50/41 51/41 52/41 53/41 54/41 55/41 56/41 57/41 58/41 59/41 60/41

0108 ECL22

ENTIN 000230 16/21 17/21 18/21 19/21 20/21 21/21 22/21 23/21 24/21 25/21 26/21 27/21 28/21 29/21 30/21  
 ENTRA 000234 16/25 17/25 18/25 19/25 20/25 21/25 22/25 23/25 24/25 25/25 26/25 27/25 28/25 29/25 30/25  
 ENTR0 000235 16/26 17/26 18/26 19/26 20/26 21/26 22/26 23/26 24/26 25/26 26/26 27/26 28/26 29/26 30/26  
 EPC 000252 16/42 17/42 18/42 19/42 20/42 21/42 22/42 23/42 24/42 25/42 26/42 27/42 28/42 29/42 30/42  
 ERR 001020 25/31 26/31 27/31 28/31 29/31 30/31  
 ERR1 001055 26/03 27/03 28/03 29/03 30/03  
 ERR2 001071 26/11 27/11 28/11 29/11 30/11  
 ERR3 001074 26/07 27/07 28/07 29/07 30/07  
 ERR4 001110 25/25 26/25 27/25 28/25 29/25 30/25  
 ERRA 001003 16/24 17/24 18/24 19/24 20/24 21/24 22/24 23/24 24/24 25/24 26/24 27/24 28/24 29/24 30/24  
 ERB 001014 15/56 16/56 17/56 18/56 19/56 20/56 21/56 22/56 23/56 24/56 25/56 26/56 27/56 28/56 29/56 30/56  
 ERRET 000212 MC 16/22 17/22 18/22 19/22 20/22 21/22 22/22 23/22 24/22 25/22 26/22 27/22 28/22 29/22 30/22  
 ERROR 000020 MC 39/41 40/41 41/41 42/41 43/41 44/41 45/41 46/41 47/41 48/41 49/41 50/41 51/41 52/41 53/41 54/41 55/41 56/41 57/41 58/41 59/41 60/41

0109 ECL22

ENTIN 000230 16/21 17/21 18/21 19/21 20/21 21/21 22/21 23/21 24/21 25/21 26/21 27/21 28/21 29/21 30/21  
 ENTRA 000234 16/25 17/25 18/25 19/25 20/25 21/25 22/25 23/25 24/25 25/25 26/25 27/25 28/25 29/25 30/25  
 ENTR0 000235 16/26 17/26 18/26 19/26 20/26 21/26 22/26 23/26 24/26 25/26 26/26 27/26 28/26 29/26 30/26  
 EPC 000252 16/42 17/42 18/42 19/42 20/42 21/42 22/42 23/42 24/42 25/42 26/42 27/42 28/42 29/42 30/42  
 ERR 001020 25/31 26/31 27/31 28/31 29/31 30/31  
 ERR1 001055 26/03 27/03 28/03 29/03 30/03  
 ERR2 001071 26/11 27/11 28/11 29/11 30/11  
 ERR3 001074 26/07 27/07 28/07 29/07 30/07  
 ERR4 001110 25/25 26/25 27/25 28/25 29/25 30/25  
 ERRA 001003 16/24 17/24 18/24 19/24 20/24 21/24 22/24 23/24 24/24 25/24 26/24 27/24 28/24 29/24 30/24  
 ERB 001014 15/56 16/56 17/56 18/56 19/56 20/56 21/56 22/56 23/56 24/56 25/56 26/56 27/56 28/56 29/56 30/56  
 ERRET 000212 MC 16/22 17/22 18/22 19/22 20/22 21/22 22/22 23/22 24/22 25/22 26/22 27/22 28/22 29/22 30/22  
 ERROR 000020 MC 39/41 40/41 41/41 42/41 43/41 44/41 45/41 46/41 47/41 48/41 49/41 50/41 51/41 52/41 53/41 54/41 55/41 56/41 57/41 58/41 59/41 60/41





0109 ECL22

0V14 004055 64/02  
 0V14A 004055 64/18  
 0V14B 004073 64/16 64/36  
 0V14C 004113 65/01  
 0V16 004121 65/17  
 0V16A 004133 65/08  
 0V17 004150 65/21  
 0V17A 004185 65/27 65/36  
 0V18 004173 65/41  
 0V18A 004210 65/47 65/56  
 0V19 004216 66/02  
 0V19A 004234 66/09 66/18  
 0V1A 003443 57/08 57/19  
 0V2 003444 57/21  
 0V20 004242 66/23  
 0V20A 004260 66/29  
 0V21 004266 67/02  
 0V21A 004303 67/08  
 0V22 004311 67/22  
 0V22A 004326 67/28  
 0V23 004334 68/02  
 0V23A 004357 68/08  
 0V24 004356 68/27  
 0V24A 004411 68/33  
 0V25 004420 69/02  
 0V25A 004444 69/08  
 0V26 004453 70/02  
 0V26A 004471 70/08  
 0V27 004477 70/24  
 0V27A 004515 70/30  
 0V28 004524 71/04  
 0V28A 004543 71/10 71/21  
 0V29 004552 71/26  
 0V29A 004572 71/34  
 0V2A 003465 57/27 57/37  
 0V3 003466 58/02  
 0V30 004684 72/02  
 0V30A 004691 72/08  
 0V31 004699 72/23  
 0V31A 004647 72/29  
 0V32 004657 73/02  
 0V32A 004674 73/08  
 0V33 004706 73/16  
 0V33A 004714 73/22  
 0V33B 004732 73/30  
 0V33C 004742 73/44  
 0V33D 004743 73/43 73/49  
 0V34 004750 74/02  
 0V34A 004764 74/14  
 0V34B 005067 74/31  
 0V35 005015 75/02  
 0V35A 005033 75/10  
 0V36 005034 75/18  
 0V36A 005051 75/26  
 0V37 005052 75/33  
 0V37A 005073 75/41  
 0V38 005074 76/02  
 0V38A 005115 76/18

73/20

0110 ECL22

0V39 005116 76/21  
 0V39A 005132 76/29  
 0V3A 003310 58/08  
 0V4 003311 58/22  
 0V40 005141 77/02  
 0V40A 005160 77/10  
 0V41 005166 77/22  
 0V41A 005205 77/37  
 0V42 005214 78/02  
 0V42A 005233 78/10 78/17  
 0V43 005242 78/23  
 0V43A 005255 78/31 78/36  
 0V44 005263 78/41  
 0V44A 005276 78/49 78/54  
 0V45 005304 79/02  
 0V45A 005321 79/10 79/18  
 0V46 005322 79/26  
 0V46A 005347 79/34 79/42  
 0V47 005357 80/02  
 0V47A 005374 80/10 80/18  
 0V48 005403 80/24  
 0V48A 005420 80/32 80/40  
 0V4A 003335 58/27 58/32  
 0V5 003336 59/02  
 0V51 005427 81/02  
 0V51A 005444 81/10 81/17  
 0V52 005456 82/04 82/26  
 0V52A 005471 82/12 82/17  
 0V53 005505 82/28  
 0V53A 005522 82/36 82/40  
 0V54 005525 83/02  
 0V54A 005543 83/10 83/18  
 0V54B 005547 83/16 83/19  
 0V55 005550 83/22  
 0V55A 005565 83/37 83/41  
 0V55B 005567 83/35 83/39  
 0V55C 005573 83/36 83/40  
 0V56 005603 84/02  
 0V56A 005607 84/06 84/11  
 0V56B 005633 84/19 84/28  
 0V56C 005649 84/20 84/30  
 0V56D 005659 84/29 84/35  
 0V5A 003544 59/08 59/31  
 0V6 003574 59/14 59/25  
 0V6A 003582 60/01 60/31  
 0V6B 003625 60/13 60/25  
 0V7 003653 60/34 60/53  
 0V7A 003657 60/40 60/53  
 0V8 003682 61/02  
 0V8A 003705 61/08  
 0V9 003706 61/25  
 0V9A 003724 61/31  
 0V11M 00174 MC

56/41

76/21 76/36  
 76/29 58/19  
 58/08 58/22  
 77/02 77/10  
 77/18 77/22  
 77/36 77/37  
 78/02 78/10 78/17  
 78/23 78/31 78/36  
 78/41 78/49 78/54  
 79/02 79/10 79/18  
 79/26 79/34 79/42  
 80/02 80/10 80/18  
 80/24 80/32 80/40  
 80/32 80/32 80/40  
 58/27 58/32  
 59/02 81/02  
 81/10 81/17  
 82/04 82/26  
 82/12 82/17  
 82/28 82/36 82/40  
 83/02 83/10 83/18  
 83/16 83/19  
 83/22 83/37 83/41  
 83/35 83/39  
 83/36 83/40  
 84/02 84/06 84/11  
 84/19 84/28  
 84/20 84/30  
 84/29 84/35  
 59/08 59/31  
 59/14 59/25  
 60/01 60/31  
 60/13 60/25  
 60/34 60/53  
 60/40 60/53  
 61/02 61/22  
 61/08 61/25  
 61/31 61/41  
 57/24 58/05 58/29 59/11 60/18  
 61/28 62/06 62/29 63/06 63/29  
 64/05 65/26 67/05 67/35 68/05  
 70/05 70/27 71/07 71/31 72/05 72/26

0111 ECL22

|                |    |        |        |        |       |        |        |       |       |       |        |        |        |        |
|----------------|----|--------|--------|--------|-------|--------|--------|-------|-------|-------|--------|--------|--------|--------|
| 0VT1 00027     | MC | 12/14  | 62/02  | 62/25  | 63/02 | 63/25  | 63/25  | 84/06 | 85/02 | 85/20 | 86/02  | 86/22  | 86/35  | 87/02  |
| 0VTRP 000301   | MC | 12/34  | 83/05  | 83/25  | 84/09 | 84/09  | 91/29  | 87/21 | 88/02 | 88/18 | 89/02  | 91/02  | 91/26  | 92/02  |
| PASR 001333    |    | 32/02  | 180/21 |        |       |        |        | 93/02 | 93/19 | 94/02 | 94/15  | 94/26  | 94/39  | 95/01  |
| PASS 000203    |    | 15/48  | 19/11  | 23/10  | 25/32 | 100/04 | 100/23 | 95/02 | 95/19 | 96/08 | 96/02  | 96/13  | 96/25  | 96/35  |
| PASSI 000204   |    | 15/49  | 100/02 | 100/08 |       |        |        | 14/07 | 21/12 | 21/44 | 27/09  | 27/08  | 28/35  | 28/35  |
| PASSV 000205   |    | 15/50  | 100/07 |        |       |        |        | 59/15 | 60/14 | 60/41 | 61/09  | 61/28  | 62/18  | 62/33  |
| PBK 000327     |    | 17/37  | 75/06  | 75/02  | 75/37 | 76/06  | 76/25  | 59/15 | 60/14 | 60/41 | 61/09  | 61/28  | 62/18  | 62/33  |
|                |    | 77/26  | 78/08  | 78/27  | 78/45 | 79/08  | 79/30  | 65/10 | 65/22 | 65/10 | 65/28  | 65/48  | 66/09  | 66/09  |
|                |    | 80/28  | 81/06  | 82/08  | 82/32 |        |        | 70/31 | 71/11 | 71/35 | 72/09  | 72/30  | 73/09  | 73/09  |
| PCNT 007077    |    | 101/08 |        |        |       |        |        | 74/24 | 75/11 | 75/27 | 76/42  | 76/11  | 76/30  | 77/11  |
| PDEC 001160    |    | 16/10  | 28/04  |        |       |        |        | 80/33 | 81/11 | 82/13 | 82/37  | 83/13  | 83/33  | 84/17  |
| PDEC1 001205   |    | 28/26  | 28/44  |        |       |        |        | 16/15 | 27/37 | 27/48 | 43/04  | 46/05  | 47/05  | 48/05  |
| PDEC2 001217   |    | 28/32  | 28/36  |        |       |        |        | 34/05 | 31/10 | 31/40 | 43/04  | 46/05  | 47/05  | 48/05  |
| PDEC3 001224   |    | 28/07  | 28/17  | 28/29  | 28/41 |        |        | 54/12 | 57/13 | 57/32 | 58/15  | 58/37  | 59/19  | 59/19  |
| PDERE 000225   |    | 28/05  | 28/05  | 28/15  | 28/45 |        |        | 60/18 | 60/20 | 60/23 | 60/45  | 60/47  | 60/48  | 60/48  |
| PDIIV 006670   |    | 97/31  | 97/41  |        |       |        |        | 61/15 | 61/15 | 61/16 | 61/17  | 61/20  | 61/36  | 61/36  |
| PDIIV2 006673  |    | 97/33  | 97/42  |        |       |        |        | 62/14 | 62/18 | 62/37 | 62/41  | 63/14  | 63/18  | 63/37  |
| PDIIVR 006703  |    | 17/21  | 97/31  | 97/45  |       |        |        | 63/41 | 64/21 | 65/06 | 65/32  | 65/52  | 66/13  | 66/34  |
| PDIYR 000312   |    | 24/25  | 36/13  |        |       |        |        | 67/13 | 67/33 | 68/13 | 68/38  | 69/13  | 69/13  | 69/34  |
| PERCE 001789   |    | 14/12  |        |        |       |        |        | 71/15 | 71/39 | 72/13 | 72/34  | 73/13  | 73/37  | 74/19  |
| PF 000093      |    | 14/12  |        |        |       |        |        | 75/09 | 75/25 | 75/40 | 76/09  | 76/28  | 77/09  | 77/09  |
| PHUL 006641    |    | 17/06  | 97/06  | 97/17  |       |        |        | 78/09 | 78/30 | 78/48 | 79/09  | 79/33  | 80/09  | 80/31  |
| PHULR 000310   |    | 17/19  | 97/06  | 97/17  |       |        |        | 81/09 | 82/11 | 82/35 | 83/08  | 83/28  | 84/12  | 85/31  |
| POCT 001172    |    | 16/00  | 16/05  | 16/18  | 16/19 | 18/14  | 18/14  | 91/08 | 91/32 |       |        |        |        |        |
| PRGEN 007262   |    | 16/04  |        |        |       |        |        | 97/47 |       |       |        |        |        |        |
| PSCK1 007054   |    | 180/18 | 180/26 |        |       |        |        | 97/47 |       |       |        |        |        |        |
| PSCRY 000271   |    | 16/59  | 97/48  | 98/37  | 98/46 |        |        | 97/47 |       |       |        |        |        |        |
| PSDIIV 006725  |    | 98/08  |        |        |       |        |        | 35/41 |       |       |        |        |        |        |
| PSDIIV2 006743 |    | 98/17  | 98/22  |        |       |        |        | 14/04 | 21/05 | 21/34 | 38/16  | 38/20  | 46/20  | 46/25  |
| PSDV2 006745   |    | 98/24  | 98/32  |        |       |        |        | 47/19 | 49/19 | 54/29 | 57/12  | 57/16  | 57/31  | 57/35  |
| PSDV3 006756   |    | 98/33  | 98/35  |        |       |        |        | 58/12 | 56/17 | 58/36 | 59/18  | 59/22  | 60/17  | 60/22  |
| PSDV4 006761   |    | 98/23  | 98/35  |        |       |        |        | 60/44 | 60/50 | 61/12 | 61/19  | 61/35  | 62/13  | 62/36  |
| PSHSP 000036   | MC | 10/34  |        |        |       |        |        | 63/13 | 63/36 | 64/20 | 64/24  | 65/05  | 65/31  | 65/51  |
| PSMUL 007080   |    | 99/03  |        |        |       |        |        | 66/12 | 66/33 | 67/12 | 67/32  | 68/12  | 68/37  | 69/12  |
| RAN 000276     | MC | 17/07  | 37/05  | 27/07  | 27/19 | 27/19  | 89/09  | 70/12 | 70/34 | 71/14 | 71/17  | 71/38  | 72/12  | 72/33  |
| RAND 000012    |    | 10/14  | 26/14  | 27/24  | 38/39 | 46/06  | 47/20  | 70/12 | 72/40 | 73/12 | 73/16  | 73/46  | 74/18  | 74/21  |
|                |    | 49/00  | 53/12  | 53/36  | 54/32 | 68/15  | 68/48  | 72/36 | 72/40 | 73/12 | 73/36  | 73/46  | 74/18  | 74/21  |
|                |    | 70/15  | 70/37  | 71/41  | 76/31 | 77/12  | 78/12  | 74/25 | 74/26 | 75/07 | 75/23  | 75/38  | 76/07  | 76/28  |
|                |    | 79/12  | 79/36  | 80/12  | 80/34 | 85/28  | 86/25  | 77/07 | 77/27 | 78/07 | 78/28  | 78/46  | 79/07  | 79/31  |
|                |    | 87/07  | 87/26  | 91/41  | 92/05 | 93/05  | 96/39  | 80/07 | 80/29 | 81/07 | 81/18  | 82/09  | 82/15  | 82/17  |
|                |    | 11/30  | 53/22  |        |       |        |        | 82/33 | 83/07 | 83/27 | 83/40  | 84/11  | 84/31  | 85/08  |
| RANST 000170   | MC | 30/16  | 36/20  | 38/52  |       |        |        | 85/12 | 85/29 | 85/42 | 86/08  | 86/08  | 86/31  | 86/38  |
| REST 001321    |    | 101/09 |        |        |       |        |        | 17/08 | 19/08 | 19/10 | 19/07  | 19/07  | 19/31  | 19/31  |
| RTRN 007100    |    | 14/11  |        |        |       |        |        | 10/29 | 10/29 | 10/29 | 10/29  | 10/29  | 10/29  | 10/29  |
| SC 000002      |    | 35/20  |        |        |       |        |        | 19/15 | 19/15 | 19/26 | 19/26  | 19/26  | 19/26  | 19/26  |
| SDVH1 001651   |    | 19/18  |        |        |       |        |        | 16/12 | 21/32 | 20/03 | 100/40 | 100/41 |        |        |
| SETUP 001773   |    | 19/18  |        |        |       |        |        | 15/34 | 18/26 | 19/03 |        |        |        |        |
| SETUP 000005   | MC | 10/09  | 36/09  | 38/30  | 39/03 | 39/25  | 40/02  | 19/24 | 24/10 | 24/45 | 30/10  | 100/13 | 101/10 | 101/10 |
|                |    | 41/02  | 41/21  | 42/02  | 42/20 | 43/02  | 44/03  | 45/03 | 45/03 | 45/14 | 45/17  | 45/27  | 45/28  | 46/04  |
|                |    | 45/08  | 46/09  | 46/37  | 47/09 | 48/02  | 48/02  | 46/15 | 46/18 | 46/31 | 46/32  | 47/04  | 47/15  | 47/18  |
|                |    | 50/02  | 50/02  | 51/07  | 52/06 | 53/03  | 53/27  | 47/34 | 47/35 | 48/04 | 48/34  | 49/18  | 49/33  | 49/34  |
|                |    | 55/20  | 55/20  | 55/36  | 56/02 | 57/02  | 57/21  | 51/06 | 51/17 | 51/27 | 51/28  | 52/05  | 52/16  | 52/27  |
|                |    | 59/02  | 59/08  | 60/07  | 60/34 | 61/02  | 61/25  | 51/06 | 51/11 | 54/22 | 54/24  | 54/28  | 54/45  | 54/48  |
|                |    | 62/26  | 63/03  | 63/26  | 64/16 | 65/01  | 65/41  | 52/08 | 52/08 | 56/28 | 56/03  | 59/17  | 60/27  | 60/28  |
|                |    | 66/02  | 66/23  | 67/02  | 67/22 | 68/02  | 68/02  | 60/02 | 60/10 | 60/27 | 60/28  | 64/11  | 64/19  | 64/32  |
|                |    | 70/02  | 70/24  | 71/04  | 71/28 | 72/02  | 72/02  | 64/33 | 73/27 | 73/35 | 73/51  | 73/52  | 74/17  | 74/17  |
|                |    | 73/32  | 74/14  | 75/02  | 75/18 | 75/33  | 76/02  | 74/33 | 74/34 | 82/03 | 82/14  | 82/22  | 82/23  | 82/23  |
|                |    | 77/02  | 77/22  | 78/02  | 78/23 | 78/41  | 79/02  | 74/33 | 74/34 | 82/03 | 82/14  | 82/22  | 82/23  | 82/23  |
|                |    | 80/02  | 80/24  | 81/02  | 82/04 | 82/28  | 83/02  | 84/21 | 84/21 | 84/27 | 84/30  | 84/37  | 84/38  | 84/38  |

SNACK 000707  
SNM1 001866  
SP 000040

SRAN 000277 MC  
STACK 000027  
START 000563  
STK 000566  
SWMS 000533  
SWREG 007101  
TEM 000240

TEM0 000275

0113 ECL22

|       |        |       |       |       |       |       |       |       |
|-------|--------|-------|-------|-------|-------|-------|-------|-------|
| TEM1  | 000273 | 17/03 | 45/07 | 45/29 | 46/08 | 46/33 | 47/08 | 47/36 |
|       |        | 49/08 | 49/35 | 54/15 | 54/50 | 59/07 | 59/29 | 60/06 |
|       |        | 60/20 | 64/15 | 64/34 | 73/31 | 73/53 | 74/07 | 74/35 |
| TEM2  | 000274 | 17/04 | 54/06 | 54/27 | 54/44 |       |       |       |
| TD    | 000044 | 14/08 | 21/17 | 58/26 | 83/09 | 83/11 | 88/29 | 83/31 |
|       |        | 84/13 | 84/15 | 84/22 | 91/09 | 91/11 | 91/33 | 91/35 |
|       |        | 17/28 | 36/09 | 36/09 | 36/17 | 36/20 |       |       |
| TR0   | 000317 | 36/09 |       |       |       |       |       |       |
| TR0A  | 002015 | 36/09 |       |       |       |       |       |       |
| TR0B  | 002026 | 36/17 | 36/20 |       |       |       |       |       |
| TR0C  | 002032 | 36/24 |       |       |       |       |       |       |
| TR1   | 002044 | 36/20 | 38/26 |       |       |       |       |       |
| TR10  | 002346 | 43/02 |       |       |       |       |       |       |
| TR10A | 002357 | 43/07 | 43/13 |       |       |       |       |       |
| TR10B | 002363 | 43/10 | 43/14 |       |       |       |       |       |
| TR11  | 002372 | 44/03 |       |       |       |       |       |       |
| TR11A | 002403 | 44/08 | 44/14 |       |       |       |       |       |
| TR11B | 002407 | 44/11 | 44/15 |       |       |       |       |       |
| TR12  | 002416 | 44/21 |       |       |       |       |       |       |
| TR12A | 002427 | 44/28 | 44/32 |       |       |       |       |       |
| TR12B | 002433 | 44/29 | 44/33 |       |       |       |       |       |
| TR13  | 002442 | 45/02 |       |       |       |       |       |       |
| TR13A | 002450 | 45/08 | 45/31 |       |       |       |       |       |
| TR13B | 002463 | 45/13 | 45/21 |       |       |       |       |       |
| TR13C | 002467 | 45/16 | 45/22 |       |       |       |       |       |
| TR13D | 002476 | 45/27 |       |       |       |       |       |       |
| TR14  | 002503 | 46/03 |       |       |       |       |       |       |
| TR14A | 002511 | 46/09 | 46/35 |       |       |       |       |       |
| TR14B | 002525 | 46/14 | 46/24 |       |       |       |       |       |
| TR14C | 002531 | 46/17 | 46/25 |       |       |       |       |       |
| TR14D | 002541 | 46/31 |       |       |       |       |       |       |
| TR15  | 002546 | 46/37 |       |       |       |       |       |       |
| TR15A | 002560 | 46/42 | 46/50 |       |       |       |       |       |
| TR15B | 002564 | 46/45 | 46/51 |       |       |       |       |       |
| TR16  | 002576 | 47/03 |       |       |       |       |       |       |
| TR16A | 002584 | 47/09 | 47/38 |       |       |       |       |       |
| TR16B | 002621 | 47/14 | 47/25 |       |       |       |       |       |
| TR16C | 002625 | 47/17 | 47/26 |       |       |       |       |       |
| TR16D | 002637 | 47/34 |       |       |       |       |       |       |
| TR17  | 002644 | 48/02 |       |       |       |       |       |       |
| TR17A | 002656 | 48/07 | 48/15 |       |       |       |       |       |
| TR17B | 002662 | 48/10 | 48/16 |       |       |       |       |       |
| TR18  | 002674 | 49/03 |       |       |       |       |       |       |
| TR18A | 002702 | 49/09 | 49/37 |       |       |       |       |       |
| TR18B | 002716 | 49/14 | 49/24 |       |       |       |       |       |
| TR18C | 002722 | 49/17 | 49/25 |       |       |       |       |       |
| TR18D | 002734 | 49/33 |       |       |       |       |       |       |
| TR19  | 002741 | 49/39 |       |       |       |       |       |       |
| TR19A | 002752 | 49/44 | 49/50 |       |       |       |       |       |
| TR19B | 002756 | 49/47 | 49/51 |       |       |       |       |       |
| TR1A  | 002061 | 38/38 | 38/44 |       |       |       |       |       |
| TR1B  | 002061 | 38/46 |       |       |       |       |       |       |
| TR1C  | 002064 | 38/35 | 38/48 |       |       |       |       |       |
| TR2   | 002073 | 39/03 |       |       |       |       |       |       |
| TR20  | 002765 | 50/02 |       |       |       |       |       |       |
| TR20A | 002776 | 50/07 | 50/13 |       |       |       |       |       |
| TR20B | 003062 | 50/10 | 50/14 |       |       |       |       |       |
| TR21  | 003011 | 50/20 |       |       |       |       |       |       |

0114 ECL22

|       |        |       |       |  |  |  |  |  |
|-------|--------|-------|-------|--|--|--|--|--|
| TR21A | 003023 | 50/25 | 50/32 |  |  |  |  |  |
| TR21B | 003027 | 50/28 | 50/33 |  |  |  |  |  |
| TR22  | 003037 | 51/03 |       |  |  |  |  |  |
| TR22A | 003045 | 51/07 | 51/31 |  |  |  |  |  |
| TR22B | 003057 | 51/12 | 51/21 |  |  |  |  |  |
| TR22C | 003065 | 51/15 | 51/22 |  |  |  |  |  |
| TR22D | 003072 | 51/27 |       |  |  |  |  |  |
| TR22E | 003100 | 51/16 | 51/33 |  |  |  |  |  |
| TR23  | 003101 | 51/32 | 52/02 |  |  |  |  |  |
| TR23A | 003105 | 52/06 | 52/31 |  |  |  |  |  |
| TR23B | 003121 | 52/11 | 52/20 |  |  |  |  |  |
| TR23C | 003125 | 52/14 | 52/21 |  |  |  |  |  |
| TR23D | 003143 | 52/19 | 52/33 |  |  |  |  |  |
| TR24  | 003144 | 52/32 | 53/03 |  |  |  |  |  |
| TR24A | 003157 | 53/11 | 53/17 |  |  |  |  |  |
| TR24B | 003170 | 53/08 | 53/23 |  |  |  |  |  |
| TR25  | 003175 | 53/27 |       |  |  |  |  |  |
| TR25A | 003212 | 53/35 | 53/43 |  |  |  |  |  |
| TR25B | 003216 | 53/42 | 53/47 |  |  |  |  |  |
| TR25C | 003225 | 53/32 | 53/51 |  |  |  |  |  |
| TR26  | 003232 | 54/02 |       |  |  |  |  |  |
| TR26A | 003242 | 54/10 | 54/52 |  |  |  |  |  |
| TR26B | 003250 | 54/16 |       |  |  |  |  |  |
| TR26C | 003272 | 54/37 |       |  |  |  |  |  |
| TR26D | 003301 | 54/26 | 54/41 |  |  |  |  |  |
| TR26E | 003306 | 54/21 | 54/43 |  |  |  |  |  |
| TR26F | 003315 | 54/48 |       |  |  |  |  |  |
| TR27  | 003322 | 55/03 |       |  |  |  |  |  |
| TR27A | 003332 | 55/11 | 55/13 |  |  |  |  |  |
| TR27B | 003335 | 55/08 | 55/16 |  |  |  |  |  |
| TR28  | 003342 | 55/20 |       |  |  |  |  |  |
| TR28A | 003352 | 55/28 | 55/30 |  |  |  |  |  |
| TR28B | 003354 | 55/25 | 55/32 |  |  |  |  |  |
| TR29  | 003361 | 55/30 |       |  |  |  |  |  |
| TR29A | 003371 | 55/44 | 55/46 |  |  |  |  |  |
| TR29B | 003373 | 55/41 | 55/48 |  |  |  |  |  |
| TR2A  | 002102 | 39/11 | 39/12 |  |  |  |  |  |
| TR2B  | 002115 | 39/08 | 39/21 |  |  |  |  |  |
| TR3   | 002122 | 39/25 |       |  |  |  |  |  |
| TR30  | 003400 | 56/02 |       |  |  |  |  |  |
| TR30A | 003412 | 56/12 | 56/14 |  |  |  |  |  |
| TR30B | 003415 | 56/09 | 56/17 |  |  |  |  |  |
| TR3A  | 002131 | 39/33 | 39/34 |  |  |  |  |  |
| TR3B  | 002144 | 39/30 | 39/43 |  |  |  |  |  |
| TR4   | 002151 | 40/02 |       |  |  |  |  |  |
| TR4A  | 002160 | 40/10 | 40/11 |  |  |  |  |  |
| TR4B  | 002173 | 40/07 | 40/20 |  |  |  |  |  |
| TR5   | 002200 | 40/25 |       |  |  |  |  |  |
| TR5A  | 002207 | 40/33 | 40/34 |  |  |  |  |  |
| TR5B  | 002222 | 40/30 | 40/43 |  |  |  |  |  |
| TR6   | 002227 | 41/02 |       |  |  |  |  |  |
| TR6A  | 002244 | 41/07 | 41/10 |  |  |  |  |  |
| TR7   | 002252 | 41/21 |       |  |  |  |  |  |
| TR7A  | 002263 | 41/26 | 41/32 |  |  |  |  |  |
| TR7B  | 002267 | 41/29 | 41/33 |  |  |  |  |  |
| TR7C  | 002276 | 42/02 |       |  |  |  |  |  |
| TR8A  | 002307 | 42/07 | 42/13 |  |  |  |  |  |
| TR8B  | 002313 | 42/10 | 42/14 |  |  |  |  |  |

41/14

0115 ECL22

|       |        |       |        |        |       |       |
|-------|--------|-------|--------|--------|-------|-------|
| TRD   | 002322 | 42/20 | 42/31  | 39/28  | 48/85 | 48/28 |
| TRSA  | 002333 | 42/25 | 42/32  | 43/85  | 44/86 | 44/24 |
| TRSB  | 002337 | 10/40 | 30/12  | 42/85  | 48/85 | 49/42 |
| TRAPE | 000100 | 41/05 | 41/24  | 47/12  | 53/30 | 54/19 |
|       |        | 45/11 | 46/12  | 48/85  |       |       |
|       |        | 50/85 | 50/33  | 52/89  |       |       |
|       |        | 55/86 | 55/33  | 56/87  |       |       |
|       |        | 16/13 | 21/84  |        |       |       |
| TRP   | 000816 | 11/83 | 39/12  | 40/34  |       |       |
| TRST  | 000104 | 12/25 | 75/85  | 75/38  | 76/24 | 77/85 |
| UPLH  | 000253 | 77/25 | 78/85  | 79/26  | 79/39 | 80/85 |
|       |        | 80/27 | 81/85  | 82/87  |       |       |
|       |        | 17/34 | 181/12 |        |       |       |
| UP    | 000324 | 17/34 |        |        |       |       |
| VCTAB | 007102 | 22/84 |        |        |       |       |
| VEC   | 000685 | 13/82 |        |        |       |       |
| VECTO | 000331 | 22/84 | 22/11  | 22/20  |       |       |
| VRET  | 000785 | 11/13 | 05/18  |        |       |       |
| ZERDA | 000151 | 15/25 | 27/85  |        |       |       |
| ZERAN | 001113 | 15/22 | 26/31  | 180/26 |       |       |
| ZEGS  | 000010 | 16/25 | 27/88  | 27/87  |       |       |
| ZRAND | 001115 | 16/14 | 27/83  |        |       |       |
| ZRANS | 001133 |       |        |        |       |       |





LISTING

096-000244-02

PROGRAM

EXERCISER FOR ECLIPSE  
PART 6

TAPE

095-000229-02

ABSTRACT

'ECLIPSE23' IS AN EXERCISER PROGRAM USED TO TEST THE RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF THE ECLIPSE COMPUTER. 'ECLIPSE23' EXERCISES THE EXTENDED INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES OF ITS RELIABLE OPERATION.





```

0001 ECL23 MACRO REV 03.00      15:42:10 08/06/76
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

```

```

:0002 ECL23

```

```

.TIIL ECL23
:ECLIPSE23
:ECLIPSE23 - CONTINUATION OF ECLIPSE22
:PART 6 OF EXERCISER FOR ECLIPSE
:

```

```

02
03
04
05
06
07

```

```

:*****

```

```

: NAME: ECLIPSE23.SR          PART NUMBER: 094-000628

```

```

: DESCRIPTION: ECLIPSE EXERCISER, PART 6

```

```

: REVISION HISTORY:

```

| REV. | DATE     |
|------|----------|
| 00   | 08/02/74 |
| 01   | 12/20/74 |
| 02   | 08/06/76 |

```

: COPYRIGHT (C) DATA GENERAL CORPORATION, 1974, 1976
: ALL RIGHTS RESERVED.
:*****

```

10003 ECL23

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

```

EXERCISER FOR ECLIPSE: PART 6

PROGRAM NAME

ECLIPSE23

GENERAL DESCRIPTION

'ECLIPSE23' IS AN EXERCISER PROGRAM USED TO TEST THE RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF THE ECLIPSE COMPUTER. 'ECLIPSE23' EXERCISES THE EXTENDED INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES OF ITS RELIABLE OPERATION.

THE INSTRUCTIONS EXERCISED ARE AS FOLLOWS:

SYSTEM STACK FEATURE: SCL,XCT AND VCT

LOCATIONS 200 TO 205 IN PAGE 0 ARE FIXED FOR ECLIPSE23 PROGRAM.

LOCATION 203 KEEPS TRACK OF NUMBER OF PASSES RUN THROUGH ECLIPSE23 PROGRAM.

LOCATION 201 KEEPS TRACK OF THE TEST RUNNING AT PRESENT AND IS USEFUL FOR DEBUG WHEN LOOPING OCCURS IN THE PROGRAM.

LOCATION 202 CONTAINS THE STARTING ADDRESS OF ECLIPSE23 PROGRAM.

LOCATION 200 IS USED BY DTOS.

LOCATION 204 KEEPS TRACK OF INTERNAL PASS COUNT WHICH IS FIXED BY LOCATION 205.

FIRST PASS THROUGH ECLIPSE23 TEST WILL RUN SUPERFAST. NEXT PASSES WILL RUN SLOWER AS EACH TEST IS RUN SEVERAL TIMES TO RUN ALL RANDOM NUMBER COMBINATIONS.

MACHINE REQUIREMENTS

ECLIPSE PROCESSOR  
 4K READ-WRITE MEMORY  
 CONSOLE EQUIPMENT

10004 ECL23

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

```

SWITCH SETTINGS

THIS PROGRAM USES DATA SWITCHES AS FOLLOWS

SW"0" - USE CONTENTS OF "SMREG" IF 0  
 USE DATA SWITCHES IF 1  
 SW"1" - LOOP ON FAILING TEST IF 0  
 PROCEED TO NEXT TEST IF 1  
 SW"2" - OUTPUT TO TTY IF 0  
 INHIBIT PRINTING TO TTY IF 1  
 SW"3" - DO NOT PRINT \* ERRORS IF 0  
 PRINT FAILURE RATE IF 1  
 SW"4" - PRINT PASS COUNT IF 0  
 INHIBIT PRINTING PASS COUNT IF 1  
 SW"5" - INHIBIT OUTPUT TO LINE PRINTER IF 0  
 OUTPUT TO LINE PRINTER IF 1

EDTOS PRESUMES THAT ALL SWITCHES ARE 0. RETURN IS MADE TO THE OPER SYS UPON AN ERROR AFTER FIRST PRINTING THE FAILURE REPORT. IN STAND ALONE MODE, THE OPERATOR IS ASKED TO PRESET THESE SWITCHES (THE MACHINE HALTS FOR HIM TO DO SO) AND PRESS CONTINUE. THIS PRESET VALUE CAN BE CHANGED AT ANY TIME BY SETTING DATA SWITCH "0" TO A ONE. THE PROGRAM WILL THEN FOLLOW ACTUAL SETTINGS.

STAND ALONE STARTING ADDRESS = 200

IF 'CAT' OR 'KITTEN' WAS LOADED FROM DTOS AND RESTR WAS NEEDED, THEN USE AS FOLLOWS:

STARTING ADDR = 170 (FOR START WITH NO 'CAT')  
 STARTING ADDR = 171 (FOR START WITH 'CAT')

MONITOR LOCATION 203 TO CHECK THE CURRENT PASS COUNT

MONITOR LOCATION X6000 TO MAKE SURE THAT 'CAT' OR 'KITTEN' IS RUNNING. IN CASES WHERE PROGRAM IS STARTED WITH 'CAT' OR 'KITTEN' LOCATION X6000 WILL SHOW A PATTERN CHANGING FROM ZEROS: TO ALL ONES TO AN IVC/SWAP PATTERN.

(X= THE NUMBER OF THE HIGHEST MEMORY MODULE IN THE SYSTEM AND MAY BE A VALUE 0 - 7)

10005 ECL23

01  
02 OPERATING PROCEDURE/OPERATOR INPUT  
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

15. OPERATING PROCEDURE/OPERATOR INPUT  
-----  
15.1 LOAD THE PROGRAM VIA THE BINARY LOADER OR INSERT A  
15.2 PRELOADED MEMORY MODULE.  
15.3 SET SWITCHES TO 200.  
PROGRAM WILL HALT AFTER PRINTING THE MESSAGE  
'SET DATA SWITCHES AND PRESS CONTINUE'.  
11 SET REQUIRED SWITCH SETTINGS AND PRESS CONTINUE.  
12 THE PROGRAM WILL RUN UNTIL MANUALLY STOPPED. IN CASE  
13 OF MALFUNCTIONING, THE PROGRAM WILL PRINT ERROR  
14 MESSAGE AND TAKE APPROPRIATE ACTION AS PER THE SW  
15 SETTINGS.  
16  
17 PROGRAM OUTPUT/ERROR DESCRIPTION  
18  
19  
20 FOR ANY ERRORS DETECTED, THE PROGRAM WILL PRINT ERROR  
21 REPORT OR X FAILURES DEPENDING UPON THE SW SETTINGS.  
22 ERROR REPORT CONSISTS OF ALL ACCUMULATORS, CARRY,  
23 RELOCATED PROGRAM COUNTER OF THE TEST THAT IS FAILING  
24 AND PC IN THE LISTING AT THE TIME OF FAILURE.  
25 THE PROGRAM WILL LOOP IN THE TEST THAT IS FAILING IF  
26 SW"1" IS 0.  
27 THE PRINTING OF ERROR REPORT CAN BE ABORTED BY SETTING  
28 SW"2" TO 1.  
29 IF LOOPING OCCURS IN THE PROGRAM, STOP THE COMPUTER  
30 AND CHECK LOCATION 201 TO FIND OUT THE TEST THAT WAS  
31 RUNNING BEFORE THE LOOPING OCCURRED.  
32

10006 ECL23

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

17. PROGRAM DESCRIPTION/THEORY OF OPERATION  
-----  
17.1 EACH TEST IS COMPLETE IN ITSELF, SO THE PROGRAM CAN  
17.2 BE STARTED FROM ANY TEST WITHOUT CAUSING ANY  
INITIALIZATION ERRORS.  
WHEN 'ECLIPSE3' IS STARTED AT LOCATION 200 OR BY  
DTOS, IT WILL SIZE UP THE MEMORY AND WILL PRINT  
THE TOP OF THE MEMORY.  
AFTER SETTING UP THE SWITCHES AND PRESSING CONTINUE,  
THE EXERCISE WILL RUN THE FIRST PASS VERY FAST. IN  
THE FIRST PASS EACH TEST IS RUN ONLY ONCE. ALL OTHER  
PASSES WILL TAKE MORE TIME AS EACH TEST IS RUN  
ACCORDING TO THE TEST COUNT SPECIFIED IN EACH TEST.  
REFER TO THE LISTING TO FIND OUT THE INFORMATION  
ABOUT EACH TEST.  
18 RESTRICTIONS/MISC  
-----  
19 CERTAIN INSTRUCTIONS LIKE RLM, XCT, BAM, ETC.,  
20 DO ALLOW INTERRUPTS TO OCCUR DURING THEIR  
21 EXECUTION. THIS FEATURE OF THOSE INSTRUCTIONS IS  
22 NOT CHECKED IN THIS TEST.  
23  
24  
25  
26  
27  
28

10007 ECL23

10008 ECL23

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

?      ***  MACRO DEFINITIONS  ***

.MACRO LOOP
JSR @ENTLO
%
%
.MACRO SETUP
JSR @ENTIN
%1
%
.MACRO RAND
JSR @ENTRA
%
%
.MACRO BRAND
JSR @ENTRB
%
.MACRO ERROR
JMP @+2
**
**
**
**
**
.MACRO STACK
JSR @ISTK
%1
%
.MACRO PSMSP
SETUP 20
STACK PSHS%1
MOV 0,0
PSH %1,%1
LDA 0,SP
LDA 1,@BEG
MOV 0,0,SZC
SUB# 0,1,SZR
PSMS%1: ERROR
LOOP
%

.MACRO TRAPER
JSR @ITRP
%1
%2
%3
%4
%

.MACRO TRIST
LDA 0,@BEG
LDA 1,@+1+1
ADD 0,1
XOP %1,%1,%2
SUB# 2,3,SNR
SUB# 1,2,SZR
ERROR
%
%
.MACRO ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUB 3,3
%
%
.MACRO RAMER
STA 3,AC3
JSR @ENTBE
%
%
.MACRO MDERR
STA 3,AC3
**
**
JSR @ENTBE
%
%
.MACRO RANST
JSR @I,RAN
%RANDOM # TO AC0+2
%
%
.MACRO MXTI
LDA %1,%1,125252
SUBZL %2,%2
LDA %3,%3,IMP %3
LDA %4,%4,=ADD %2,%1
XCT %4
MOVZL %2,%2
MOVZL %2,%2,SNR
XCT %3
LDA %2,%2,=LDA %2,%2,+2
XCT %2
JMP %+2
M1
SUB# %2,%1,SZR
%ON ERR CK AC%1, AC%2.
%

```

```

***  MACRO DEFINITIONS  ***

.MACRO LOOP
JSR @ENTLO
%
%
.MACRO SETUP
JSR @ENTIN
%1
%
.MACRO RAND
JSR @ENTRA
%
%
.MACRO BRAND
JSR @ENTRB
%
.MACRO ERROR
JMP @+2
**
**
**
**
**
.MACRO STACK
JSR @ISTK
%1
%
.MACRO PSMSP
SETUP 20
STACK PSHS%1
MOV 0,0
PSH %1,%1
LDA 0,SP
LDA 1,@BEG
MOV 0,0,SZC
SUB# 0,1,SZR
PSMS%1: ERROR
LOOP
%

.MACRO TRAPER
JSR @ITRP
%1
%2
%3
%4
%

.MACRO TRIST
LDA 0,@BEG
LDA 1,@+1+1
ADD 0,1
XOP %1,%1,%2
SUB# 2,3,SNR
SUB# 1,2,SZR
ERROR
%
%
.MACRO ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUB 3,3
%
%
.MACRO RAMER
STA 3,AC3
JSR @ENTBE
%
%
.MACRO MDERR
STA 3,AC3
**
**
JSR @ENTBE
%
%
.MACRO RANST
JSR @I,RAN
%RANDOM # TO AC0+2
%
%
.MACRO MXTI
LDA %1,%1,125252
SUBZL %2,%2
LDA %3,%3,IMP %3
LDA %4,%4,=ADD %2,%1
XCT %4
MOVZL %2,%2
MOVZL %2,%2,SNR
XCT %3
LDA %2,%2,=LDA %2,%2,+2
XCT %2
JMP %+2
M1
SUB# %2,%1,SZR
%ON ERR CK AC%1, AC%2.
%

```

```

***  MACRO DEFINITIONS  ***

.MACRO LOOP
JSR @ENTLO
%
%
.MACRO SETUP
JSR @ENTIN
%1
%
.MACRO RAND
JSR @ENTRA
%
%
.MACRO BRAND
JSR @ENTRB
%
.MACRO ERROR
JMP @+2
**
**
**
**
**
.MACRO STACK
JSR @ISTK
%1
%
.MACRO PSMSP
SETUP 20
STACK PSHS%1
MOV 0,0
PSH %1,%1
LDA 0,SP
LDA 1,@BEG
MOV 0,0,SZC
SUB# 0,1,SZR
PSMS%1: ERROR
LOOP
%

.MACRO TRAPER
JSR @ITRP
%1
%2
%3
%4
%

.MACRO TRIST
LDA 0,@BEG
LDA 1,@+1+1
ADD 0,1
XOP %1,%1,%2
SUB# 2,3,SNR
SUB# 1,2,SZR
ERROR
%
%
.MACRO ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUB 3,3
%
%
.MACRO RAMER
STA 3,AC3
JSR @ENTBE
%
%
.MACRO MDERR
STA 3,AC3
**
**
JSR @ENTBE
%
%
.MACRO RANST
JSR @I,RAN
%RANDOM # TO AC0+2
%
%
.MACRO MXTI
LDA %1,%1,125252
SUBZL %2,%2
LDA %3,%3,IMP %3
LDA %4,%4,=ADD %2,%1
XCT %4
MOVZL %2,%2
MOVZL %2,%2,SNR
XCT %3
LDA %2,%2,=LDA %2,%2,+2
XCT %2
JMP %+2
M1
SUB# %2,%1,SZR
%ON ERR CK AC%1, AC%2.
%

```

```

10000 FCL23
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

:DISPATCH TABLE AT LOCATION
:VECTOR... ENTRY 0 SET
:TO ADDRESS *1. OTHERS
:TO ADDRESS *2.

: FILL THE BUFFER WITH
:*1.

: ADVANCE TO NEXT BUFFER
: LOCATION AND TEST FOR
: END OF BUFFER-110. IF NOT
: END OF BUFFER GO TO *1.

: FIND THE FINAL ADDRESS
: AND VALUE IN THE INDIRECT
: CHAIN. C(CAC2)= FIRST
: ADDRESS AT WHICH TO
: START LOOKING.
: IF INDIRECT CHAIN IS
: *1, EXIT TO *1.

: AUTO INDEX REGISTERS
: ARE USED AS VARIABLES.

: MAKE SURE THAT FINAL
: DATA WILL NOT POINT
: TO ANY SPOT IN THE CHAIN.
: IF IT DOES, EXIT TO *1.

: C(CAC2)=FINAL ADDRESS
: C(CAC3)=DATA AT THAT
: ADDRESS.

10010 FCL23
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

: THE ADDRESS TO GO TO

: ON OVERFLOW IS *2.

: SET C(CAC0), STACK, AND
: FRAME POINTERS TO
: THE CONTENTS OF *1.
: TEST OF STACK OVERFLOW.

: STACK LIMIT IS *1
: GREATER THEN STACK POINTER.
: OVERFLOW SHOULD OCCURE.

: SETUP STACK IN PAGE 0
: TO UNDER FLOW ON ANY
: POP, POPJ, RTN, OR POPR.
: INSTRUCTION. THE FAULT
: LOCATION IS *1.

: TEST OVERFLOW OF THE STACK
: ON THE CAP INSTRUCTION.

: IF NO OV GO TO *2.
: IF OV GO TO *1.

:VECTOR
JSR @VEEC
VP
*2
*1
0

: FILL
LDA 2, @REG
INC 2, 3
LDA 1, @END
SUB 2, 1
LDA 0, *1
STA 0, 0, 2
RLM

AGAIN
ISZ TEM
LDA 1, @END
LDA 0, @110
SUB 0, 1
LDA 0, TEM
AOCH 0, 1, SZR
JMP *1

: MACRO FIND
LDA 2, @20
STA 2, 20
SUMZL 3, 3
INCL 3, 3, SZC
JMP *1
LDA 2, 0, 2
STA 2, @20
MOVZL# 2, 2, SZC
JMP *5
LDA 3, 0, 2
LDA 0, 20
LDA 1, @20
STA 1, 20
LDA 1, @20
MOVZL 1, 1
MOVZL 1, 1
SUB# 1, 2, SZR
SUB# 1, 3, SVR
JMP *1
LDA 1, 20
SUB# 0, 1, SZR
JMP *10
:ADDRESS.

:VECTOR
JSR @VEEC
VP
*2
*1
0

: FILL
LDA 2, @REG
INC 2, 3
LDA 1, @END
SUB 2, 1
LDA 0, *1
STA 0, 0, 2
RLM

AGAIN
ISZ TEM
LDA 1, @END
LDA 0, @110
SUB 0, 1
LDA 0, TEM
AOCH 0, 1, SZR
JMP *1

: MACRO FIND
LDA 2, @20
STA 2, 20
SUMZL 3, 3
INCL 3, 3, SZC
JMP *1
LDA 2, 0, 2
STA 2, @20
MOVZL# 2, 2, SZC
JMP *5
LDA 3, 0, 2
LDA 0, 20
LDA 1, @20
STA 1, 20
LDA 1, @20
MOVZL 1, 1
MOVZL 1, 1
SUB# 1, 2, SZR
SUB# 1, 3, SVR
JMP *1
LDA 1, 20
SUB# 0, 1, SZR
JMP *10
:ADDRESS.

```

```
10011 ECL23
01
02
03
04 SP=40 :STACK POINTER
05 FP=41 :FRAME POINTER
06 SL=42 :STACK LIMIT
07 SF=43 :STACK FAULT
08 TO=44 :TRAP ORIGIN
09 PF=45 :FLOATING POINT FAULT
10
11 SC=2
12 PF=3
13 ISP=4
14 MK=5
15 ISL=6
16 ISF=7
```

```
*****LITERAL DEFINITIONS*****
04 :STACK POINTER
05 :FRAME POINTER
06 :STACK LIMIT
07 :STACK FAULT
08 :TRAP ORIGIN
09 :FLOATING POINT FAULT
10
11 SC=2
12 PF=3
13 ISP=4
14 MK=5
15 ISL=6
16 ISF=7
```

```
10012 ECL23
```

```
***** DIAGNOSTIC PROGRAM PREAMBLE *****
02
03
04 .LOC 0 :POINTER TO DIRT BLOCK
05 00000 007002
06
07 .LOC 40 :STACK CONTROL LOCATIONS
08
09 00000 000000
10 00041 000000
11 00002 000000
12 00003 000044
13 00044 063077
14
15 .LOC 45 :POINTER TO EGGS BLOCK
16 00045 006645
17
18 .ZREL 8. :8 LOCATIONS RESERVED
19 00000-000010 :FOR DEBUG BREAKPOINTS
20
21
22 00010-006645 :EGGS: EGGS
23 00011-000000 MEMTOP: 0
24 00012-000000 ICAT: 0
25
26 : SPACE RESERVED FOR CAT/NO CAT RESTART ENTRIES
27
28 .LOC 170
29 00170 102841 OFF:
30 00171 102000 ON:
31 00172 142470
32 006454
33 00174 002175 JMP @+1
34 00175 000333 SWMSS
35
36 : SPACE RESERVED FOR SPECIAL RESTART ENTRIES
37
38 .LOC 176
39 00176 000002 .BLK 2
40
41
42 : LOCATIONS 200 - 213 RESERVED FOR ECLIPSE TESTS
43
44 .LOC 200
45 00200 002202 DTDS8: JMP @BGNADR
46 00201 000000 ITRET: 0
47 00202 000500 @GNADR: NSTRT
48 00203 000000 PASS: 0
49 00204 000001 PASSINI: 1
50 00205 000001 PASSVAL: 1
51
52 00206 000000 ITR: 0
53 00207 000000 ITRACT: 0
54 00210 000000 ITRER: 0
55 00211 000000 ITRFC: 0
56 00212 000000 ERRET: 0
57 00213 000000 LOPRET: 0
58
```

```
*****
02
03
04 .LOC 0 :POINTER TO DIRT BLOCK
05 00000 007002
06
07 .LOC 40 :STACK CONTROL LOCATIONS
08
09 00000 000000
10 00041 000000
11 00002 000000
12 00003 000044
13 00044 063077
14
15 .LOC 45 :POINTER TO EGGS BLOCK
16 00045 006645
17
18 .ZREL 8. :8 LOCATIONS RESERVED
19 00000-000010 :FOR DEBUG BREAKPOINTS
20
21
22 00010-006645 :EGGS: EGGS
23 00011-000000 MEMTOP: 0
24 00012-000000 ICAT: 0
25
26 : SPACE RESERVED FOR CAT/NO CAT RESTART ENTRIES
27
28 .LOC 170
29 00170 102841 OFF:
30 00171 102000 ON:
31 00172 142470
32 006454
33 00174 002175 JMP @+1
34 00175 000333 SWMSS
35
36 : SPACE RESERVED FOR SPECIAL RESTART ENTRIES
37
38 .LOC 176
39 00176 000002 .BLK 2
40
41
42 : LOCATIONS 200 - 213 RESERVED FOR ECLIPSE TESTS
43
44 .LOC 200
45 00200 002202 DTDS8: JMP @BGNADR
46 00201 000000 ITRET: 0
47 00202 000500 @GNADR: NSTRT
48 00203 000000 PASS: 0
49 00204 000001 PASSINI: 1
50 00205 000001 PASSVAL: 1
51
52 00206 000000 ITR: 0
53 00207 000000 ITRACT: 0
54 00210 000000 ITRER: 0
55 00211 000000 ITRFC: 0
56 00212 000000 ERRET: 0
57 00213 000000 LOPRET: 0
58
```

```
*****
02
03
04 .LOC 0 :POINTER TO DIRT BLOCK
05 00000 007002
06
07 .LOC 40 :STACK CONTROL LOCATIONS
08
09 00000 000000
10 00041 000000
11 00002 000000
12 00003 000044
13 00044 063077
14
15 .LOC 45 :POINTER TO EGGS BLOCK
16 00045 006645
17
18 .ZREL 8. :8 LOCATIONS RESERVED
19 00000-000010 :FOR DEBUG BREAKPOINTS
20
21
22 00010-006645 :EGGS: EGGS
23 00011-000000 MEMTOP: 0
24 00012-000000 ICAT: 0
25
26 : SPACE RESERVED FOR CAT/NO CAT RESTART ENTRIES
27
28 .LOC 170
29 00170 102841 OFF:
30 00171 102000 ON:
31 00172 142470
32 006454
33 00174 002175 JMP @+1
34 00175 000333 SWMSS
35
36 : SPACE RESERVED FOR SPECIAL RESTART ENTRIES
37
38 .LOC 176
39 00176 000002 .BLK 2
40
41
42 : LOCATIONS 200 - 213 RESERVED FOR ECLIPSE TESTS
43
44 .LOC 200
45 00200 002202 DTDS8: JMP @BGNADR
46 00201 000000 ITRET: 0
47 00202 000500 @GNADR: NSTRT
48 00203 000000 PASS: 0
49 00204 000001 PASSINI: 1
50 00205 000001 PASSVAL: 1
51
52 00206 000000 ITR: 0
53 00207 000000 ITRACT: 0
54 00210 000000 ITRER: 0
55 00211 000000 ITRFC: 0
56 00212 000000 ERRET: 0
57 00213 000000 LOPRET: 0
58
```

!0013 ECL23

01 : \*\*\*\*\*LOCAL ZREL\*\*\*\*\*

02 :  
03 :  
04 00214 007053 !INLOC: PRGEND+100  
05 00215 007753 !ALOC: PRGEND+1000

06 :  
07 : POINTERS TO SUBROUTINES

08 :  
09 00216 001172 !POCT: POCT  
10 00217 001160 !PDEC: PDEC  
11 00220 001233 !MESS: MESS  
12 00221 000646 !STK: STK  
13 00222 000616 !TRAP: TRAP  
14 00223 001133 !LRAN: LRAN  
15 00224 001144 !SIZE: SIZE

16 :  
17 00225 002014 !REG: REGIN  
18 00226 007053 !REG: PRGEND+100  
19 00227 007353 !REWD: PRGEND+400  
20 :  
21 00230 000706 !ENTN: INIT  
22 00231 000727 !ENTL: LOP  
23 00232 001010 !ENTER: ERN6  
24 00233 001003 !ENTBE: ERRA  
25 00234 001115 !ENTRA: -RANO  
26 00235 001113 !ENTRB: -BKAN

27 :  
28 : TEMPORARYS FOR TESTS AND SUBROUTINES

29 :  
30 00240 000240 !LOC 240  
31 00240 000000 !TEM: 0  
32 00241 005000 !B1: 240\*20  
33 00242 005017 !B17: 240\*20+17  
34 00243 004777 !B1: 240\*20-1  
35 00244 005020 !B20: 240\*20+20

36 :  
37 00245 000000 !ACO: 0  
38 00246 000000 !AC1: 0  
39 00247 000000 !AC2: 0  
40 00250 000000 !AC3: 0  
41 00251 000000 !CAY: 0  
42 00252 000000 !EPC: 0

43 :  
44 00253 000000 !CHART: 0  
45 00254 000000 !CHURZ: 0  
46 00255 000000 !PDERET: 0  
47 00256 000000 !MESRET: 0

48 :  
49 00257 000000 !ITEM: 0  
50 00260 000000 !BOK0: 0  
51 00261 000000 !BOK1: 0  
52 00262 000000 !BOK2: 0  
53 00263 000000 !BOK3: 0  
54 00264 000000 !K0: 0  
55 00265 000000 !BAMSUM: 0  
56 00266 000000 !BAMRET: 0  
57 00267 000000 !BAMXZ: 0  
58 00270 000000 !BAMXB: 0  
59 00271 000000 !PSCR1: 0

!CONSTANT 0

!0014 ECL23

01 :  
02 00272 177777 !M1: -1  
03 00273 000000 !E41: 0  
04 00274 000000 !E42: 0  
05 00275 000000 !E40: 0

06 :  
07 00276 000000 !RAV: 0  
08 00277 000000 !SRAN: 0

09 :  
10 :  
11 00300 000000 !ORIG0: 0  
12 00301 000000 !ORIG1: 0  
13 00302 000000 !ORIG2: 0  
14 00303 000000 !ORIG3: 0  
15 00304 000000 !ORIG4: 0  
16 00305 000000 !ORIG5: 0  
17 00306 000000 !ORIG6: 0  
18 00307 000000 !ORIG7: 0  
19 00310 000000 !ORIG8: 0  
20 00311 000000 !ORIG9: 0  
21 00312 000000 !ORIG10: 0

22 :  
23 :  
24 00313 000000 !BAMSIZ: 0  
25 00314 000000 !BAMT1: 0  
26 00315 000000 !BAMT0: 0  
27 00316 000000 !FRUM: 0  
28 00317 000000 !TOO: 0  
29 00320 000000 !RAC0: 0  
30 00321 000000 !RAC1: 0  
31 00322 000000 !RAC2: 0  
32 00323 000000 !RAC3: 0

33 :  
34 00324 006653 !VP: VCTAB  
35 00325 005400 !JS43: JSR 0,3  
36 00326 005000 !JSR2: JSR 0,2  
37 00327 000010 !P8K: .BLK 8.

!RANDOM #  
!SAVED RANDOM #  
!VARIABLES IN MUL/DIV TEST  
!ORIGINAL #  
!CORRECT NUMBER

!BAM TEST VARIABLES

```

:0015 ECL23
01 000500 .LOC 500
02
03
04
05 00500 062677 NSTRT: IDNST
06 00501 062224 JSR @ISIZE
07 00502 150400 NEG
08 00503 150000 CDB
09 00504 050011- STA 2, MEMTOP
10 00505 122470 ELDA 0, CATSW
11 00506 066141
12 00507 101005 MOV 0, 0, SNR
13 00510 00415 JMP NOCAT
14 00511 122470 ELDA 0, PRGEND
15 00513 024075- LDA 1, 1, 1777
17 00514 123000 ADD 1, 0
18 00515 112033 ADCZ#
19 00516 000407 JMP NOCAT
20
21 00517 132400 SUB
22 00520 024074- LDA 1, 2400
23 00521 147000 ADD 2, 1
24 00522 044012- STA 1, ICAT
25 00523 050215 STA 2, MAXLOC
26 00524 000407 JMP SWNESS
27
28 00525 024073- NOCAT: LDA 1, 200
29 00526 132400 SUB 1, 2
30 00527 126400 SDB 1, 1
31 00530 146470 ESTA 1, CATSW
32 00531 006116
33 00532 050215 STA 2, MAXLOC
34
:0016 ECL23
01
02
03 00533 004220 SWMES: JSR @INESS
04 00534 001754 MESIZ
05 00535 024011- LDA 1, MEMTOP
06 00536 101040 MOV 0, 0
07 00537 006216 JSR @IPOCT
08 00540 006220 JSR @INESS
09 00541 001770 KCRLF
10 00542 126400 SUB 1, 1
11 00543 044203 STA 1, PASS
12
13 00544 126470 ELDA 1, AUTO
14
15 00546 125004 MOV 1, 1, SZR
16 00547 000414 JMP START
17
18 00550 006220 JSR @INESS
19 00551 001772 SETSW
20 00552 006220 JSR @INESS
21 00553 001770 KCRLF
22 00554 043077 HALT
23 00555 000401 JMP
24 00556 060477 READS 0
25 00557 142470 ESTA 0, SWREG
26 006072
27 00561 000402 JMP START

```

```

: *****OUTPUT STPT MESSAGE & READ SWITCHES*****
:PRINT SIZE OF MEMORY
:INIT PASS COUNT
:RUNNING IN AUTO MODE?
:YES START PROGRAM.
:NO, PRINT SET SWITCHES MESS.
:READ NEW STATE OF SWITCHES

```

```

: *****SIZE SYSTEM & RESERVE MEMORY*****
:CALL SIZE ROUTINE
:STORE ADDRESS OF LAST...
:LOCATION IN MEMTOP.
:TEST IF DTOS SET CATSW
:IF 0 CAT WAS NOT LOADED.
:IF CAT WAS LOADED
:TEST FOR SUFFICIENT
:MEMORY TO RUN CAT
:TAKE 2K OFF THE TOP
:OF MEMORY FOR CAT
:SET UP CAT START ADDR.
:MAX LOC FOR BUFFERS
:TAKE 200 OFF THE
:TOP OF MEMORY FOR
:MINI-MONITOR
:& CLEAR CATSW.
:MAX LOC FOR BUFFERS

```



```

:0017 ECL23
01
02 00562 006012- START: JSK @ICAT
03
04
05
06
07
08
09
10 00563 102400 HEG1: SUB 0,0
11 00564 040210 STA 0,I,TRR
12
13
14
15 00565 006234 RAND
16 00566 105300 JSR @ENTRA
17 00567 030215 MOV5 0,1
18 00570 142422 LDA 2,MAXLOC
19 00571 000777 SUBZ 2,0,SZC
20 00572 143000 JMP *-1
21 00573 146422 ADD 2,0
22 00574 000777 SUBZ 2,1,SZC
23 00575 147000 JMP *-1
24
25 00576 030214 BEG2: LDA 2,MINLOC
26 00577 142433 SUBZ# 2,0,SNC
27 00600 000763 JMP BEG1
28 00601 146433 SUBZ# 2,1,SNC
29 00602 000761 JMP BEG1
30 00603 106433 SUBZ# 0,1,SNC
31 00604 104710 XCH 0,1
32
33 00605 131000 HEG3: MOV 1,2
34 00606 112400 SUB 0,2
35 00607 034072- LDA 3,3=377
36 00610 172433 SUBZ# 3,2,SNC
37 00611 000752 JMP BEG1
38 00612 040226 STA 0,HBEG
39 00613 044227 STA 1,HBEND
40 00614 002501 JMP @,*1
41 00615 002014 HEGIN
42
:0018 ECL23
01
02
03
04 00616 023401 TRP: LDA 0,@1,*3
05 00617 040040 STA 0,SP
06 00620 040041 STA 0,FP
07 00621 020227 LDA 0,REWD
08 00622 024071- LDA 1,=100
09 00623 122400 SUB 1,0
10 00624 040042 STA 0,SL
11 00625 021400 LDA 0,0,3
12 00626 040043 STA 0,SP
13 00627 040045 STA 0,FF
14 00630 053401 LDA 2,@1,*3
15 00631 024070- LDA 1,=10
16 00632 133000 ADD 1,2
17 00633 050044 STA 2,T0
18 00634 024067- LDA 1,=-32-
19 00635 041000 STA 0,0,2
20 00636 151400 INC 2,2
21 00637 125404 INC 1,1,SZR
22 00640 000775 JMP *-3
23 00641 025402 LDA 1,2,3
24 00642 133000 ADD 1,2
25 00643 025403 LDA 1,3,3
26 00644 045240 STA 1,=-32,,2
27 00645 001404 JMP 4,3
28
29
30
31
32 00646 024226 STK: LDA 0,HBEG
33 00647 100110 SBI 1,0
34 00650 040040 STA 0,SP
35 00651 040041 STA 0,FP
36 00652 040004 LDA 0,ISP
37 00653 020227 LDA 0,REWD
38 00654 024066- LDA 1,=100
39 00655 122400 SUB 1,0
40 00656 040042 STA 0,SL
41 00657 040006 STA 0,ISL
42 00660 021400 LDA 0,0,3
43 00661 040045 STA 0,SP
44 00662 040043 STA 0,FF
45 00663 040007 STA 0,ISF
46 00664 001404 JMP 1,3
:TRAPER FAULT, ORIGIN, TRAP#, SUBROUTINE ADDRESS
:SET STACK POINTER
:AND FRAME POINTER
:END OF BUFFER-100
:IS THE STACK LIMIT
:STACK FAULT ADDRESS
:INIT FAULT AND
:FLOATING FAULT
:TRAP ORIGIN ADDRESS
:SET ORIGIN
:PUT FAULT RETURN
:IN THE DISPATCH TABLE
:TRAP NUMBER.....
:SUBROUTINE ADDRESS
:SET A SINGLE SUBROUTINE
:ADDRESS AND EXIT
:STACK INITIALIZATION ROUTINE
:INITIALIZE A STACK.....
:MAKE STACK POINTER
:AND FRAME POINTER
:POINT TO THE DATA
:BUFFER -1.
:BUFFER END -100 IS
:THE STACK LIMIT
:SETUP STACK FAULT
:AND FLOATING FAULT

```

```

10019 ECL23
01
02
03
04 00665 054430 VEC:
05 00666 033400
06 00667 021401
07 00670 041000
08 00671 155400
09 00672 024066
10 00673 133710
11 00674 034411
12 00675 021402
13 00676 025403
14 00677 133000
15 00700 041500
16 00701 062677
17 00702 102000
18 00703 062077
19 00704 001404
20 00705 000000 VRET:
0
; VECTOR TABLE INITIALIZER
;SETUP A VECTOR TABLE.....
;CALL+1=ORIGIN POINTER
;CALL+2=ERROR ADDRESS
;CALL+3=CORRECT ENTRY
;CALL+4=ENTRY NUMBER
;TABLE IS FILLED WITH ERR
;RETURNS...
;GOOD ENTRY
;POSITION (DEVICE CODE).
;THE GOOD ENTRY.
;MASK OTHER DEVICES.
ADC 0,0
MSKO 0
JMP 4,3
0
; *****TEST UTILITY SUBROUTINES*****
; SUBROUTINE TO INITIALIZE A TEST LOOP
01
02
03
04
05 00706 175400 INI: INC 3,3 ;TEST LOOP INITIALIZER
06 00707 054201 STA 3,ITRET ;SAVE RETURN LOCATION
07 00710 040245 STA 0,ACO ;SAVE CONTENTS OF ACO
08 00711 021777 LDA 0,-1,3 ;GET # OF ITERATIONS
09 00712 040206 STA 0,ITR ;SET ITER. VALUE
10 00713 040207 STA 0,ITRCT ;SET ITER. COUNT
11 00714 176400 SUB 3,3 ;CLEAR ERROR SWITCH
12 00715 054210 STA 3,ITREC ;CLEAR ERROR COUNT
13 00716 054211 STA 3,ITREC ;TEST FOR FIRST PASS
14 00717 034203 LDA 3,PASS
15 00720 175004 MOV 3,3,SZR
16 00721 000404 JMP INI1
17
18 00722 176520 SURZL 3,3 ;THIS IS 1'ST PASS
19 00723 054206 STA 3,ITR ;SET ITERATIONS FOR
20 00724 054207 STA 3,ITRCT ;1 LOOP ONLY.
21
22 00725 020245 INI1: LDA 0,ACO ;RESTORE AC'S AND
23 00726 002201 JMP @ITRET ;EXIT TO TEST

```

```

10020 ECL23
01
02
03
04
05 00706 175400 INI: INC 3,3 ;TEST LOOP INITIALIZER
06 00707 054201 STA 3,ITRET ;SAVE RETURN LOCATION
07 00710 040245 STA 0,ACO ;SAVE CONTENTS OF ACO
08 00711 021777 LDA 0,-1,3 ;GET # OF ITERATIONS
09 00712 040206 STA 0,ITR ;SET ITER. VALUE
10 00713 040207 STA 0,ITRCT ;SET ITER. COUNT
11 00714 176400 SUB 3,3 ;CLEAR ERROR SWITCH
12 00715 054210 STA 3,ITREC ;CLEAR ERROR COUNT
13 00716 054211 STA 3,ITREC ;TEST FOR FIRST PASS
14 00717 034203 LDA 3,PASS
15 00720 175004 MOV 3,3,SZR
16 00721 000404 JMP INI1
17
18 00722 176520 SURZL 3,3 ;THIS IS 1'ST PASS
19 00723 054206 STA 3,ITR ;SET ITERATIONS FOR
20 00724 054207 STA 3,ITRCT ;1 LOOP ONLY.
21
22 00725 020245 INI1: LDA 0,ACO ;RESTORE AC'S AND
23 00726 002201 JMP @ITRET ;EXIT TO TEST

```

```

1:0021 ECL23
01
02
03
04 00727 054213 LOP: STA 3,LOPRET
05 00730 014207 ITRCT
06 00731 000440 JMP LOP3
07 00732 034210 LDA 3,ITRER
08 00733 175005 MOV 3,3,SMR
09 00734 002213 JMP 4LOPRET
10 00735 034206 LDA 3,ITR
11 00736 054207 STA 3,ITRCT
12
13 00737 074077 LOP1: READS
14 00740 175112 MOVLA
15 00741 000405 JMP
16 00742 136470 ELDA
17
18 00744 177100 ADDL
19 00745 177103 ADDL
20 00746 000421 JMP
21 00747 040245 STA
22 00750 044246 STA
23 00751 050247 STA
24 00752 006220 JSR
25 00753 001725 PERCENT
26 00754 102400 SUB
27 00755 024211 LDA
28 00756 040211 STA
29 00757 030065 LDA
30 00760 143710 MUL
31 00761 030206 LDA
32 00762 153710 DIV
33 00763 006217 JSR
34 00764 028245 LDA
35 00765 028246 LDA
36 00766 030247 LDA
37 00767 176400 LOP2: SUB
38 00770 054211 STA
39
40 00771 034210 LOP3: LDA
41 00772 175004 MOV
42 00773 074077 READS
43 00774 175112 MOVLA
44 00775 000405 JMP
45 00776 136470 ELDA
46
47 01000 177113 ADDL#
48 01001 002201 JTRCT
49 01002 002213 JMP

```

```

; SUBROUTINE TO TERMINATE A TEST LOOP
;
; LONG AND SHORT FORM FROM ROUTINES
;
; IF SWITCH, *3* PRINT X
; NO X PRINT OUT REQUIRED
;
; SAVE AC'S
; MESSAGE *12* <15> X FAIL=*
;
; SET ERROR COUNT
; CLEAR ERROR COUNT.
;
; (COUNT X 100) / ITERATIONS =
; PERCENTAGE OF FAILURE
; DECIMAL PRINTER
;
; IF NO ERROR, ITERATE
; OTHERWISE LOOK AT DATA
; *1* SWITCH FOR PROCEED,
; OR NOT.
;
; LONG PRINT ENTRY POINT
; SAVE PC OF ERROR
; FACO & CARRY.
;
; SPACE DOWN PRINTLIST
; TO FIND THE RETURN
; POINT.
;
; SHORT PRINT ENTRY POINT
; SAVE PC OF ERROR
; FACO & CARRY
;
; SAVE RETURN POINT
; BUMP ERROR COUNT
;
; NEW ERROR?
; NO, RETURN
; YES
;
; SAVE REMAINING AC'S
;
; PRINT *ERROR IN PASS*
; PRINT FAILING PASS
;
; PRINT HEADER
; SET LEADING ZERO SUPPRESSION
;
; PRINT MACHINE STATE

```

```

1:0022 FCL23
01
02
03
04 01003 054252 ERRA: STA 3,EPC
05 01004 040245 STA 0,ACO
06 01005 102560 SURCL 0,0
07 01006 040251 STA 0,CRY
08 01007 021400 LDA 0,0,3
09 01010 175400 INC 3,5
10 01011 101004 MOV 0,0,SZR
11 01012 000775 JMP *-3
12 01013 000405 ERR
13
14
15 01014 054252 ERRA: STA 3,EPC
16 01015 040245 STA 0,ACO
17 01016 102560 SURCL 0,0
18 01017 040251 STA 0,CRY
19
20 01020 054212 ERR: STA 3,ERRCT
21 01021 010211 ISZ ITREC
22
23 01022 020210 LDA 0,ITRER
24 01023 116033 ADCZ# 0,3,SMC
25 01024 000464 JMP ERRA
26
27 01025 054210 STA 3,ITRER
28 01026 040246 STA 1,AC1
29 01027 050247 STA 2,AC2
30 01030 006220 JSR 3,MESS
31 01031 001337 ERMSG
32 01032 024203 LDA 1,PASS
33 01033 125420 INCZ 1,1
34 01034 006217 JSR 3,IPDEC
35 01035 006220 JSR 3,MESS
36 01036 001732 HEADR
37 01037 101020 MOVZ 0,0
38 01040 024251 LDA 1,CRY
39 01041 006217 JSR 3,IPDEC
40 01042 101040 MOVZ 0,0
41 01043 024245 LDA 1,ACO
42 01044 006216 JSR 3,IPUCT
43 01045 024246 LDA 1,AC1
44 01046 006216 JSR 3,IPUCT
45 01047 024247 LDA 1,AC2
46 01050 006216 JSR 3,IPUCT
47 01051 024250 LDA 1,AC3
48 01052 006216 JSR 3,IPUCT
49 01053 024252 LDA 1,EPC
50 01054 006216 JSR 3,IPUCT
51

```

```

:0023 ECL23
01
02
03
04
05 01055 030252 ERR1: LDA 2,EPC
06 01056 10252 EPC
07 01057 020212 ISZ
08 01060 142027 LDA 0,ERR1
09 01061 000413 ADCJ 2,0,8W
10 01062 035000 JMP ERR3
11 01063 175113 LDA 3,0,2
12 01064 000405 MOVL# 3,3,SNC
13 01065 054402 JMP ERR2
14 01066 006220 STA 3,+2
15 01067 000090 JSR @MESS
16 01070 000765 JMP ERR1
17 01071 025400 ERR2: LDA 1,0,3
18 01072 006216 JSR @IPOCT
19 01073 000762 JMP ERR1
20
21 01074 006220 ERR3: JSR @MESS
22 01075 001770 KCRLF
23 01076 024246 LDA 1,AC1
24 01077 030247 LDA 2,ACC
25 01100 136470 ELDA 3,AUTO
26 005544 MOV 0,5544
27 01102 175005 JMP ERR4
28 01103 000405
29
30 01104 062677 IORST
31 01105 034010- LDA 3,-EGGS
32 01106 035404 LDA 3,4,3
33 01107 001400 JMP 0,3
34
35 01110 020245 ERR4: LDA 0,AC0
36 01111 034250 LDA 3,ACS
37 01112 002212 JMP @ERR1

```

```

:0024 ECL23
01
02
03
04
05 01113 020276 .RAN: LDA 0,RAN
06 01114 000405 JMP .RAND*4
07 01115 020276 .RAND: LDA 0,RAN
08 01116 024210 LDA 1,ITRR
09 01117 125004 MOV 1,1,SR
10 01120 001400 JMP 0,3
11 01121 105000 HXL 2,1
12 01122 125410 MOV 0,1
13 01123 107000 ADD 0,1
14 01124 125120 MOVZL 1,1
15 01125 125120 MOVZL 1,1
16 01126 123000 ADD 1,0
17 01127 024064- LDA 1,-33031
18 01130 123000 ADD 1,0
19 01131 040276 STA 0,RAN
20 01132 001400 JMP 0,3
21
22
23 01133 054303 .RANS: STA 3,ORIG3
24 RAND
25 01134 006234 JSR @ENTRA
26 01135 105700 INCS 0,1
27 01136 110400 NEG 0,2
28 01137 133000 ADD 1,2
29 01140 040300 STA 0,ORIG0
30 01141 044301 STA 1,ORIG1
31 01142 050302 STA 2,ORIG2
32 01143 002303 JMP @ORIG3
33
34
35
36
37 01144 030214 .SIZING SUBROUTINE: RETURNS SIZE OF LOGICAL MEM IN AC0.
38 01145 131400 LDA 2,MTNLOC
39 01146 151112 INC 2,2
40 01147 000406 MOVL# 2,2,SZC
41 01150 021000 JMP *+6
42 01151 051000 LDA 0,0,2
43 01152 025000 STA 2,0,2
44 01153 041000 LDA 1,0,2
45 01154 132414 STA 0,0,2
46 01155 001400 SUB# 1,2,SZR
47 01156 050011- JMP 0,3
48 01157 000766 STA 2,MENTOP
49 SIZE+1

```

```

:RANDOM NUMBER GENERATOR SUBROUTINES
:GENERATE A NEW RANDOM
:NUMBER IN C(AC0) AND
:C(RAN), IF C(ITRER)=0
:OTHERWISE LOAD C(AC0)
:WITH OLD #.

```

0024 ECL23  
01 01247 000767  
02 JMP MESS1

```
10025 ECL23  
01  
02  
03 :*****PRINT ROUTINES*****  
04 01160 175100 PDEC: MOVL 3,3 :DECIMAL PRINT C(AC1).  
05 01161 054255 STA 3,PDERET  
06 01162 175200 MOVN 3,3 :RESET C(CARRY) FOR ZERO SUPPRESSION  
07 01163 004441 JSR PDEC3 :SET C(CARRY) IF NOT  
08 01164 023420 10000.  
09 01165 001750 1000.  
10 01166 000144 100.  
11 01167 000012 10.  
12 01170 000001 1.  
13 01171 000000 0  
14 01172 175100 PDC1: MOVL 3,3 :OCTAL PRINT C(AC1)  
15 01173 054255 STA 3,PDERET  
16 01174 175200 MOVR 3,3 :RESET C(CARRY) FOR ZERO SUPPRESSION  
17 01175 004427 JSR PDEC3 :SET C(CARRY) IF NOT  
18 01176 100000 100000  
19 01177 010000 10000  
20 01200 001000 1000  
21 01201 000100 100  
22 01202 000010 10  
23 01203 000001 1  
24 01204 000000 0  
25  
26 01205 020063-PDEC1: LDA 0,-11 :PRECESS WITH TAB  
27 01206 031377 LDA 2,-1,1,2  
28 01207 151015 MOVA 2,2,SNK :EXIT, ALL DIGITS PRINTED  
29 01210 000915 JMP PDEC3+1  
30 01211 102060 SUBC 0,0  
31 01212 146452 SUMOV 2,1,SZC  
32 01213 000404 JMP PDEC2 :FORM THE DIGIT  
33 01214 146420 SUBZ 2,1 :AND SET C(CARRY)  
34 01215 101400 INC 0,0  
35 01216 000774 JMP --4 :SKIP IF LAST DIGIT  
36 01217 151234 PDEC2: MOVZM 2,2,SZR :SKIP IF ZERO SUPPRESS  
37 01220 152462 SUBC 2,2,SZC :AND SET C(CARRY)  
38 01221 030062- LDA 2,-560 :MAKE COUNT INTO ASCII  
39 01222 143004 ADD 2,0,SZR  
40 01223 171401 INC 3,2,SKP  
41 01224 171401 INC 3,2,SKP :PRINT  
42 01225 004823 JSR CHAR :SKIP IF TAB EXIT  
43 01226 155004 MOV 2,3,SZR :NEXT DIGIT  
44 01227 000756 JMP PDEC1 :MESSAGE PRINTER  
45 01230 034255 LDA 3,PDERET :JSR (MESS)  
46 01231 175200 MOVR 3,3 :MESSAGE ADDRESS  
47 01232 001400 JMP 0,3 :MESSAGE ADDRESS  
48  
49 01233 175000 MESS: INC 3,3  
50 01234 054256 STA 3,MESRET :JSR (MESS)  
51 01235 031177 LDA 2,-1,3 :MESSAGE ADDRESS  
52 01236 020072-MESS1: LDA 0,-577  
53 01237 025000 INCZ 2,2  
54 01240 151420 AND 1,0  
55 01241 123400 AND 1,0  
56 01242 106700 SUBS 0,1  
57 01243 004405 JSR CHAR :JSR CHAR  
58 01244 121005 MOV 1,0,SNR :MOV 1,0,SNR  
59 01245 002256 JMP MESRET :JMP MESRET  
60 01246 004402 JSR CHAR :JSR CHAR
```

0025 ECL23  
 01 01331 000746  
 02  
 03 01332 000086 CHRSV: 0

JMP CHAR1

:TEMP SAVE FOR AC2

:LPT/TTO INTERFACE ROUTINE: CHARACTER PASSED IN AC0

:0027 ECL23

```

01 02
03 01250 175100 CHAR: MOVL# 3,3
04 01251 054253 STA 3,CHARET
05 01252 050460 STA 2,CHRVS
06
07 01253 074477 READS 3
08 01254 175112 MOVL# 3,3,SZC
09 01255 000403 JMP *3
10 01256 156470 ELDA 3,SWREG
11
12 01260 030061 LDA 2,-22000
13 01261 173400 AND
14 01262 153120 ADDZL 2,2
15 01263 153265 ADDCR 2,2,SNR
16 01264 000435 JMP REST
17
18 01265 034072 LDA 3,-377
19 01266 117725 ANDZS 0,3,SNR
20 01267 000432 JMP REST
21
22 01270 103004 ADD 0,0,SZR
23 01271 000777 JMP *-1
24 01272 177260 ADDCR 3,3
25 01273 020060 LDA 0,-211*400
26 01274 162445 SUBO 3,0,SNR
27 01275 000430 JMP CHAR4
28
29 01276 161340 MOVOS 3,0
30 01277 01254 CHAR1: ISZ CHORZ
31
32 01300 151133 MOVZL# 2,2,SNR
33 01301 000405 JMP CHAR2
34 01302 061117 DOAS 0,LPT
35 01303 043517 SKPBZ LPT
36 01304 000777 JMP *-1
37 01305 060217 NIIC LPT
38
39 01306 151133 CHAR2: 2,2,SNC
40 01307 000405 JMP CHAR3
41 01310 061111 DOAS 0,TTO
42 01311 063511 SKPBZ TTO
43 01312 000777 JMP *-1
44 01313 060211 NIIC TTO
45
46 01314 175403 CHAR3: INC 3,3,SNC
47 01315 000762 JMP CHAR1
48 01316 030057 LOA 2,-212
49 01317 142405 SUB 2,0,SNR
50 01320 040254 STA 0,CHORZ
51
52 01321 030411 REST: LDA 2,CHRVS
53 01322 034253 LDA 3,CHARET
54 01323 175200 MOVR 3,3
55 01324 001400 JMP 0,3
56
57 01325 034254 CHAR4: LDA 3,CHORZ
58 01326 020056 LOA 0,-8
59 01327 114410 IOR 0,3
60 01330 020055 LDA 0,-240

```

```

:SAVE RETURN ADDR.
:SAVE AC2
:READ SWITCHES INTO AC3
:TEST SWITCH 0
: SMO SET
: SMO CLEAR, DEFAULT
:MASK SM2 & SM5 INTO
:AC2 FROM AC3
:LEFT JUSTIFY SM2
:COMPLEMENT SM2
:NO OUTPUT, RETURN
:MASK CHARACTER INTO L.BYTE
:OF AC3, CLEAR CARRY.
:IF NULL CHAR. RETURN
:DETERMINE REQUIRED
:STATE OF PARITY BIT &
:INSERT IT
:TEST FOR TAB
: TRUE: SETUP TAB SIMULATION
:RESTORE CHARACTER TO R.BYTE &C0,
:SET CARRY, BUMP LINE COUNT.
:SEND TO LPT?
:NOPE, MUST BE TTO
:O.K. FETCH CHARACTER
:WAIT FOR DONE
:CLEAR DEVICE
:SEND TO TTO?
:O.K. SEND CHARACTER
:WAIT FOR DONE
:CLEAR DEVICE
:IF TABBING, AND NOT
:FINISHED, LOOP.
:TEST FOR CR/LF
:TRUE: ZERO LINE COUNT
:OTHERWISE RETURN
:SET UP TO TAB
:(AC3) IS TWO'S COMPLEMENT
:OF # OF SPACES NEEDED

```

```

:0029 ECL23
01
02 01333 005215 PARMES: .TXTE !<15><12>PASS !
03 040520
04 051523
05 000240
06 01337 005215 EMSG: .TXTE !<15><12>ERROR IN PASS: !
07 151305
08 147722
09 120322
10 047311
11 050240
12 051501
13 035123
14 000240
15 01350 005215 BTQHI: .TXTE !<15><12>
16 01351 152102 8TD ADDR GOOD BAD WORD <15><12> !
17 004717
18 042101
19 151104
20 043411
21 147717
22 040504
23 040502
24 040504
25 147727
26 042322
27 106640
28 000012
29 01366 005215 BTZHI: .TXTE !<15><12>
30 01367 152102 HTZ ADDR GOOD BAD WORD <15><12> !
31 042101
32 042101
33 151104
34 043411
35 147717
36 040504
37 040502
38 040504
39 147727
40 042322
41 106640
42 000012
43 01404 005215 BAMI: .TXTE !<15><12>AC'S AFTER BAM INSTRUCTION WRONG
44 141501
45 051447
46 040640
47 152306
48 151305
49 041240
50 046501
51 144640
52 051516
53 151324
54 141325
55 144724
56 047317
57 153640
58 147722
59 043516
60 01425 005215 <15><12>ORIG C(AC0-3) !

0030 ECL23
01 151317
02 043711
03 141640
04 040450
05 030303
06 031445
07 120251
08 000000
09 01436 005215 HAMH2: .TXTE !<15><12>6000 C(AC0-3) !
10 147507
11 042317
12 141640
13 040450
14 030303
15 041455
16 120251
17 000000
18 01447 005215 HAMH3: .TXTE !<15><12>8AD C(AC0-3) !
19 040502
20 120104
21 024303
22 141501
23 026460
24 124463
25 000200
26 01457 005215 BAMH4: .TXTE !<15><12>"BAM" SET C(CARRY) !
27 041042
28 046501
29 120042
30 147523
31 120324
32 024303
33 040703
34 151322
35 124531
36 000000
37 01472 005215 BAMH5: .TXTE !<15><12>
38 01473 151306 FROM TO WORD ORIG C(AC0) 6000 HAO<15><12> !
39 046717
40 152011
41 004717
42 147727
43 042322
44 147411
45 144722
46 004507
47 024303
48 141501
49 124460
50 043411
51 147717
52 004504
53 040502
54 106504
55 000012
56 01515 005215 EDIVM: .TXTE !<15><12>
57 01516 144504 DIVX ACC ACI AC2<15><12>
58 154126
59 040411
60 030303

```

```

0031 ECL23
01 040411
02 130703
03 040411
04 131303
05 052115
06 01527 151317 ORIG ;
07 043711
08 000011
09 01532 052115 BAMH7: .TXTE !<15><12>SHOULD BE ZERO, C(AC0-3)= ;
10 044123
11 052717
12 042314
13 041240
14 120305
15 142532
16 147722
17 120254
18 024303
19 141501
20 026460
21 124463
22 120275
23 000000
24 01551 052115 BAMH8: .TXTE !<15><12> ;
25 01552 042101 ADDR GOOD BAD BAM FAILED <15><12> ;
26 151104
27 043411
28 147717
29 045004
30 040502
31 040506
32 040502
33 040513
34 040706
35 146311
36 042305
37 106640
38 000012
39 01570 052115 MULH1: .TXTE !<15><12> ;
40 01571 052115 MUL AC0 AC1 AC2<15><12>
41 047714
42 141501
43 044460
44 141501
45 044661
46 141501
47 106682
48 01601 147412 ORIG ;
49 144722
50 044507
51 000000
52 01605 052115 MULR2: .TXTE !<15><12>BAD ;
53 040502
54 044504
55 000000
56 01611 052115 MULH3: .TXTE !<15><12>GOOD ;
57 147507
58 042317
59 000011
60 01615 052115 DIVH1: .TXTE !<15><12>DIV AC0 AC1 AC2<15><12>

0032 ECL23
01 144504
02 044526
03 141501
04 044460
05 141501
06 044661
07 141501
08 106662
09 01626 147412 ORIG ;
10 144722
11 044507
12 040000
13 01532 052115 MDHM: .TXTE !<15><12>DIV/MUL AC0 AC1 AC2
14 144504
15 127526
16 052515
17 044714
18 141501
19 044460
20 141501
21 044661
22 141501
23 01644 106682 <15><12>ORIG ;
24 147412
25 144722
26 044507
27 000000
28 01651 052115 SDVH1: .TXTE !<15><12>SDIV AC0 AC1 AC2
29 042123
30 053311
31 040411
32 050303
33 040411
34 130703
35 040411
36 131303
37 01662 052115 <15><12>ORIG ;
38 151317
39 043711
40 000011
41 01666 052115 SHH1: .TXTE !<15><12>SMUL AC0 AC1 AC2
42 046523
43 146125
44 040411
45 030303
46 040411
47 130703
48 040411
49 131303
50 01677 052115 <15><12>ORIG ;
51 151317
52 043711
53 000011
54 01703 052115 BLMH5: .TXTE !<15><12>
55 01704 151306 FROM TO WORD 6000 BAD RLM FAILED ;
56 046717
57 152011
58 044717
59 147727
60 042322

```



0034 FCL23  
01 051523  
02 141640  
03 047317  
04 144724  
05 052516  
06 000305

0035 ECL23  
01 043411  
02 147717  
03 004504  
04 040502  
05 006504  
06 146102  
07 120115  
08 040706  
09 146311  
10 042305  
11 000240  
12 01725 005215 PENCEN: .TXTE !<15><12>\* FAIL=!  
13 120245  
14 040706  
15 146311  
16 000275  
17 01732 005215 HEADER: .TXTE !<15><12><15><12>  
18 005215  
19 01734 151303 CRY AC0 AC1 AC2 AC3 LISTING <15><12>!  
20 004531  
21 141501  
22 004460  
23 141501  
24 004661  
25 141501  
26 004862  
27 141501  
28 004463  
29 144714  
30 152123  
31 047311  
32 004507  
33 005215  
34 000000  
35 01754 005215 MESIZ: .TXTE !<15><12>LAST LOGICAL ADDRESS=!  
36 040714  
37 152123  
38 146240  
39 043717  
40 141711  
41 146101  
42 040640  
43 042104  
44 142722  
45 051523  
46 000275  
47 01770 005215 KCRLF: .TXTE !<15><12>!  
48 000000  
49 01772 142523 SETSW: .TXTE !SET DATA SWITCHS AND PRESS CONTINUE!  
50 120324  
51 040504  
52 040724  
53 051640  
54 144727  
55 141724  
56 051510  
57 040640  
58 042116  
59 050240  
60 142722



```

10037 ECL23
01
02
03
04
05 02073 006230
06 02074 000100
07
08 02075 020054-
09 02076 126520
10 02077 030053-
11 02100 034052-
12 02101 137370
13 02102 125120
14 02103 125123
15 02104 133370
16 02105 024051-
17 02106 127370
18 02107 000402
19 02110 000272
20 02111 122414
21
22
23 02116 006231
24
25
26 02117 006230
27 02120 000100
28
29 02121 024054-
30 02122 152520
31 02123 034053-
32 02124 020050-
33 02125 123370
34 02126 151120
35 02127 151123
36 02130 137370
37 02131 030047-
38 02132 133370
39 02133 000402
40 02134 000272
41 02135 146414
42
43
44 02142 006231

:TEST XCT ON ALL AC'S
CS27A:  SETUP 100
        JSR @ENTIN
        :INITIALIZE TEST....
        :MXT 0,1,2,3
        LDA 2,=125252
        SUBZL 1,1
        LDA 3,=ADD 1,0
        XCT 3
        :PUT INSTR IN AC2
        :XCT (C) OF AC3
        :XCT (C) OF AC1
        :SKIP ON RTH X2 SHIFT.
        MOVZL 1,1,SNC
        MOVZL 1,1,SNC
        XCT 2
        LDA 1,=LDA 1,0,+2
        XCT 1
        JMP +2
M1
SUB# 1,0,SZR
ERROR
LOOP
JSR @ENTLO
:ITERATE TEST ROUTINE....
22-
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

:INITIALIZE TEST....
:PUT INSTR IN AC0
:XCT (C) OF AC1
:SKIP AC3 TWICE &
:SKIP ON RTH X2 SHIFT.
:XCT JMP IN AC0
:XCT INDIR MEM REF
:WITH AC3
:ON ERR CK AC2, AC3.
:ITERATE TEST ROUTINE....

10038 ECL23
01
02
03 02143 006230
04 02144 000100
05
06 02145 030054-
07 02146 176520
08 02147 020053-
09 02150 024044-
10 02151 127370
11 02152 175120
12 02153 175123
13 02154 123370
14 02155 034045-
15 02156 137370
16 02157 000402
17 02160 000272
18 02161 172414
19
20
21 02166 006231
22-
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

:INITIALIZE TEST....
:PUT INSTR IN AC0
:XCT (C) OF AC1
:SKIP AC3 TWICE &
:SKIP ON RTH X2 SHIFT.
:XCT JMP IN AC0
:XCT INDIR MEM REF
:WITH AC3
:ON ERR CK AC2, AC3.
:ITERATE TEST ROUTINE....

CS27C:  SETUP 100
        JSR @ENTIN
        :INITIALIZE TEST....
        :MXT 2,3,0,1
        LDA 2,=125252
        SUBZL 3,3
        LDA 0,=JMP +3
        LDA 1,=ADD 3,2
        XCT 1
        MOVZL 3,3
        MOVZL 3,3,SNC
        XCT 0
        LDA 3,=LDA 3,0,+2
        XCT 3
        JMP +2
M1
SUB# 3,2,SZR
ERROR
LOOP
JSR @ENTLO
:ITERATE TEST ROUTINE....
CS27D:  SETUP 100
        JSR @ENTIN
        :INITIALIZE TEST....
        :MXT 3,0,1,2
        LDA 3,=125252
        SUBZL 0,0
        LDA 1,=JMP +3
        LDA 2,=ADD 0,3
        XCT 2
        MOVZL 0,0
        MOVZL 0,0,SNC
        XCT 1
        LDA 0,=LDA 0,0,+2
        XCT 0
        JMP +2
M1
SUB# 0,3,SZR
ERROR
LOOP
JSR @ENTLO
:ITERATE TEST ROUTINE....

```

```

10039 ECL23
01
02
03 02213 006230
04 02214 000100
05
06 02215 020402
07 02216 101021
08 02217 002231
09 02220 040043
10 02221 040043
11 02222 020226
12 02223 040040
13 02224 040042
14 02225 040041
15 02226 103240
16 02227 040040
17 02230 103110
18 02231 024042
19 02232 030040
20 02233 151113
21 02234 122414
22
23
24 02241 006231
25
26
27 02242 006230
28 02243 000100
29
30 02244 020442-
31 02245 040040
32 02246 020226
33 02247 040042
34 02250 020041-
35 02251 040043
36 02252 103210
37 02253 024226
38 02254 030040
39 02255 164010
40 02256 104010
41 02257 034042
42 02260 175112
43 02261 132414
44
45
46 02266 006231

```

```

CG28:
: INITIALIZE TEST....
: THE ADDRESS TO GO TO
: ON OVERFLOW IS CG28A.
: SET C(AC0), STACK, AND
: FRAME POINTERS TO
: THE CONTENTS OF RBEG.
: TEST OF STACK OVERFLOW.
: OVERFLOW SHOULD SET BIT 0
: OF STACK POINTER TO (0),
: ALSO BIT 0 OF STACK LIMIT
: TO (1).
: C(AC0)+
SUB# 1,0,SZR
ERROR
LOOP
JSR @ENTLO

CG29:
: INITIALIZE TEST....
: SETUP STACK IN PAGE 0
: TO UNDER FLOW ON ANY
: POP,POPJ,RTN, OR POPB.
: INSTRUCTION. THE FAULT
: LOCATION IS CG29A.
: THE STATE BLOCK GENERATED
: BY UNDERFLOW SHOULD START
: AT STACK LIMIT +1.
: C(AC1)=CORRECT NEW STACK POINTER
: C(AC2)=ACTUAL STACK POINTER.
: STACK LIMIT IN C(AC3) SHOULD
: HAVE ITS BIT 0(1).
SUB# 1,2,SZR
ERROR
LOOP
JSR @ENTLO

CG30:
: INITIALIZE TEST....
: SETUP STACK IN PAGE 0
: TO UNDER FLOW ON ANY
: POP,POPJ,RTN, OR POPB.
: INSTRUCTION. THE FAULT
: LOCATION IS CG30A.
: ITERATE TEST ROUTINE....
JSR @ENTIN
40
UFLIM CG30A
LDA 0,=PBR
STA 0,SP
LDA 0,=RBEG
STA 0,=SL
LDA 0,=CG30A
STA 0,=SF
POPJ
LDA 0,=SP
MOVLF 0,0,SZC
ERROR
LOOP
JSR @ENTLO

CG31:
: INITIALIZE TEST....
: SETUP STACK IN PAGE 0
: TO UNDER FLOW ON ANY
: POP,POPJ,RTN, OR POPB.
: INSTRUCTION. THE FAULT
: LOCATION IS CG31A.
: ITERATE TEST ROUTINE....
JSR @ENTIN
40
UFLIM CG31A
LDA 0,=PBR
STA 0,SP
LDA 0,=RBEG
STA 0,=SL
LDA 0,=CG31A
STA 0,=SF
RAND
JSR @ENTRA
INC 0,1
SAVE 0
POPR
LDA 3,=RBEG
LDA 2,1,3
LDA 3,2,3
SUB# 0,2,SNR
SUB# 1,3,SZR
ERROR
LOOP
JSR @ENTLO

CG32:
: INITIALIZE TEST....
: SETUP STACK IN PAGE 0
: TO UNDER FLOW ON ANY
: POP,POPJ,RTN, OR POPB.
: INSTRUCTION. THE FAULT
: LOCATION IS CG32A.
: ITERATE TEST ROUTINE....
JSR @ENTIN
40
UFLIM CG32A
LDA 0,=PBR
STA 0,SP
LDA 0,=RBEG
STA 0,=SL
LDA 0,=CG32A
STA 0,=SF
RAND
JSR @ENTRA
INC 0,1
SAVE 0
POPR
LDA 3,=RBEG
LDA 2,1,3
LDA 3,2,3
SUB# 0,2,SNR
SUB# 1,3,SZR
ERROR
LOOP
JSR @ENTLO

```

```

10041 ECL23
01
02 SETUP 100
03 JSR #ENTLN
04 100
05 UPLIM CG32A
06 LDA 0,=P8K
07 STA 0,SP
08 LDA 0,=B8EG
09 STA 0,SL
10 02344 020035-
11 02345 040043
12 RAND
13 02346 06234
14 02347 103400
15 02350 131400
16 02351 155400
17 02352 103210
18 ERROR
19 02357 034226 CG32A:
20 02360 025402 LDA 1,2,3
21 02361 031403 LDA 2,3,3
22 02362 141000 MOV 3,0
23 02363 035400 LDA 3,4,3
24 02364 132015 ADC# 1,2,SNR
25 02365 156014 ADC# 2,3,SNR
26 ERROR
27 LOOP
28 02372 06231 JSR #ENTLO
29
30 SETUP 100
31 JSR #ENTLN
32 100
33 UPLIM CG33A
34 02375 020042-
35 02376 040040 LDA 0,=P8K
36 02377 020226 LDA 0,=B8EG
37 02400 040042 STA 0,SL
38 02401 020035- LDA 0,=CG33A
39 02402 040043 STA 0,SP
40 02403 101020 MOVZ 0,0
41 02404 004402 JSR #2
42 02405 025405 SAVE 0
43 02406 163710
44 009000
45 02410 127710 RTN
46 02411 020774 CG33A:
47 02412 030226 LDA 0,=-4
48 02413 025005 :STACK VIA UNDERFLOW.
49 02414 106414 :C(CARRY)=CORRECT
50 SUR# 0,1,SNR
51 ERROR
52 02421 006231 :C(CARRY)=PC FROM STACK.
53 LOOP
54 JSR #ENTLO
55 :ITERATE TEST ROUTINE.....
56
10042 ECL23
01
02 SETUP 40
03 JSR #ENTLN
04 40
05 UPLIM CG34A
06 LDA 0,=P8K
07 STA 0,SP
08 LDA 0,=B8EG
09 STA 0,SL
10 02440 020034- LDA 0,=CG34A
11 02441 040043 STA 0,SP
12 02442 102620 SURZ8 0,0
13 02443 045042- STA 0,=P8K
14 02444 107710 POP#
15 02445 022040 CG34A: LDA 0,=B8P
16 02446 101113 MOVLD 0,0,SNR
17 ERROR
18 LOOP
19 02448 006231 JSR #ENTLO
20
21 SETUP 40
22 JSR #ENTLN
23 40
24 UPLIM CG34A
25 LDA 0,=P8K
26 STA 0,SP
27 LDA 0,=B8EG
28 STA 0,SL
29 02440 020034- LDA 0,=CG34A
30 02441 040043 STA 0,SP
31 02442 102620 SURZ8 0,0
32 02443 045042- STA 0,=P8K
33 02444 107710 POP#
34 02445 022040 CG34A: LDA 0,=B8P
35 02446 101113 MOVLD 0,0,SNR
36 02447 101113 ERROR
37 02448 101113 LOOP
38 02449 006231 JSR #ENTLO
39
40 INITIALIZE TEST.....
41
42 SETUP STACK IN PAGE 0
43 :TO UNDER FLOW ON ANY
44 :POP,POPJ,RTN, OR POPB.
45 :INSTRUCTION. THE FAULT
46 :LOCATION IS CG32A.
47
48 :C(CARRY)=RANDOM #
49 :PLACE C(CARRY) IN
50 :ASCENDING ORDER.
51
52 :FAIL TO UNDERFLOW.
53 :OVERFLOW BLOCK SHOULD
54 :START AT C(B8EG) +1.
55 :LOAD AC'S FROM BLOCK
56 :GENERATED.
57 :C(CARRY)=ADDRESS OF BLOCK.
58
59 :ITERATE TEST ROUTINE.....
60
61 INITIALIZE TEST.....
62
63 SETUP STACK IN PAGE 0
64 :TO UNDER FLOW ON ANY
65 :POP,POPJ,RTN, OR POPB.
66 :INSTRUCTION. THE FAULT
67 :LOCATION IS CG33A.
68
69 :C(CARRY)=0
70 :TEST C(PC) STORED ON
71 :STACK VIA UNDERFLOW.
72 :C(CARRY)=CORRECT
73 :C(CARRY)=PC FROM STACK.
74
75 :ITERATE TEST ROUTINE.....
76

```

```

10043 ECL23
01
02 02440 020226 CG35:   LDA 0,8BEG
03 02445 040240   STA 0,TEM
04 02446 020227   LDA 0,8BEND
05 02447 160110   SBI 4,0
06 02450 160110   SBI 4,0
07 02451 040273   STA 0,TEM1
08                               CG35A:  SETUP 2
09 02452 006230   JSR @ENTIN
10 02453 000002   2
11 UFLIM CG35B
12 02454 020042-   LDA 0,SPRK
13 02455 040040   STA 0,SP
14 02456 020226   LDA 0,8BEG
15 02457 040042   STA 0,SL
16 02460 020033-   LDA 0,=CG35B
17 02461 040043   STA 0,SP
18 02462 020240   LDA 0,TEM
19 02463 040042   STA 0,SL
20 02464 024032-   LDA 1,=5
21 02465 107000   ADD 0,1
22 02466 172110   POP 3,3
23 02467 030040   CG35B:  LDA 2,SP
24 02470 132414   SUB# 1,2,SZR
25                               ERROR
26 LOOP
27 02475 006231   JSR @ENTLO
28 02476 010240   ISZ TEM
29 02477 020240   LDA 0,TEM
30 02500 024273   LDA 1,TEM1
31 02501 106414   SUB# 0,1,SZR
32 02502 000750   JMP CG35A
33
34 CG36:  SETUP 100
35 02503 006230   JSR @ENTIN
36 02504 000100   LDA 0,SPRK
37 UFLIM CG36A
38 02505 020042-   STA 0,SP
39 02506 040040   LDA 0,8BEG
40 02507 020226   STA 0,SL
41 02510 040042   LDA 0,=CG36A
42 02511 020031-   STA 0,SP
43 02512 040043   LDA 0,SL
44 02513 020042   ADD0# 0,0
45 02514 103240   POP 1,1
46 02515 040042   STA 0,SL
47 02516 122110   MOV 0,0,SKP
48 02517 101001   ERROR
49 LOOP
50 CG36A:
51 02524 006231   JSR @ENTLO

```

```

10044 ECL23
01
02                               :SET STACK LIMIT TO
03                               :EACH POSITION OF THE
04                               :BUFFER WHEN UNDERFLOW
05                               :OCCURS THE STACK POINTER
06                               :SHOULD BE SET TO THIS
07                               :NEW POSITION.
08                               :INITIALIZE TEST....
09                               :
10                               :SETUP STACK IN PAGE 0
11                               :TO UNDER FLOW ON ANY
12                               :POP,POPJ,RTN, OR POPB.
13                               :INSTRUCTION, THE FAULT
14                               :LOCATION IS CG35B.
15
16                               :C(AC2)=STACK POINTER
17                               :C(AC1)=CORRECT
18
19                               :ITERATE TEST ROUTINE.....
20                               :ADVANCE TO NEXT
21                               :BUFFER POSITION.
22
23                               :INITIALIZE TEST....
24
25                               :SETUP STACK
26                               :FAULT ADDRESS IS CG37A
27                               :ENTER HERE AFTER SUPER CALL.
28
29                               :FAIL TO CALL SYSTEM.
30                               :STACK FAULT?
31
32                               :ITERATE TEST ROUTINE....
33                               :TEST THAT SCL INSTRUCTION
34                               :VIA SC CAN DISPATCH TO
35                               :EVERY LOCATION IN THE
36                               :BUFFER.
37                               :FILL BUFFER WITH (JSR 0,3)
38                               :INSTRUCTIONS. WHEN EXEC-
39                               :UTED THE INSTRUCTION +1 WILL
40                               :BE SAVED IN C(AC3).
41
42                               :INITIALIZE TEST....
43
44                               :INITIALIZE STACK
45                               :FAULT ADDRESS IS CG38B
46                               :POINT TO A (JSR) IN
47                               :THE BUFFER.
48                               :SET C(AC3) FOR RETURN.
49                               :ENTER HERE AFTER SYS CALL.
50                               :STACK OVERFLOW?
51
52                               :C(AC0)+1=CORRECT
53                               :C(AC3)-1=ACTUAL LOCATION.
54
55                               :ITERATE TEST ROUTINE....
56                               :ADVANCE TO NEXT BUFFER.
57
58                               :
59                               :
60

```

```

CG37:  SETUP 100
04 02525 006230   JSR @ENTIN
05 02526 000100   100
06 STACK CG37A
07 JSR @ISTK
08 CG37A
09 JSR +2
10 JMP CG37B
11 STA 3,SC
12 SYNC 0,1
13 ERROR
14 MOV 0,0,SKP
15 ERROR
16 CG37A:  LOOP
17 JSR @ENTLO
18
19 02547 030226 CG38:  LDA 2,8BEG
20 02550 024030-   LDA 1,=10.
21 02551 147000   ADD 2,1
22 02552 044240   STA 1,TEM
23 02553 154400   INC 2,3
24 02554 024227   LDA 1,BEND
25 02555 146400   SUB 2,1
26 02556 020037-   LDA 0,JSR 0,3
27 02557 041000   HLM
28 02560 135710   BLM
29
30 02561 006230   CG38A:  SETUP 2
31 02562 000002   2
32 STACK CG38B
33 JSR @ISTK
34 CG38B
35 02565 020240   LDA 0,TEM
36 02566 040002   STA 0,SC
37 02567 044407   JSR CG38B
38 JMP CG38B+1
39 ERROR
40 02575 000407   CG38B:  JMP CG38C
41 02576 113510   CG38B:  SYNC 0,2
42 02577 116014   ADC# 0,3,SZR
43 ERROR
44 LOOP
45 02604 006231   CG38C:  JSR @ENTLO
46 02605 010240   ISZ TEM
47 02606 020240   LDA 0,TEM
48 02607 024227   LDA 1,BEND
49 02610 106014   ADC# 0,1,SZR
50 02611 000750   JMP CG38A

```

```

10045 ECL25
01
02
03 02612 006230
04 02613 000100
05
06 02614 006221
07 02615 002631
08 02616 028410
09 02617 040002
10 02620 008416
11
12 02625 000416
13 02626 102627 CG39A:
14 02627 102630
15 02630 002643
16 02635 000406 CG39H:
17 02636 117510 CG39C:
18 02637 102630
19
20
21 02643 006231
22
23
24 02644 006230
25 02645 000100
26
27 02646 006221
28 02647 002657
29 02650 008402
30 02651 000412
31 02652 054002
32
33 02653 006234
34 02654 105400
35 02655 131400
36 02656 127510
37
38 02663 106015
39 02664 132014
40
41
42 02671 006231

CG39:
SETUP 100
JSR @ENTIN
100
STACK CG39B
JSR @ISTK
CG39B
LDA 0,CG39A
STA 0,ASC
JSR CG39C
ERROR
JMP CG39D
@,*1
CG39A:
@,*1
CG39D
CG39D
CG39H:
JMP CG39D
CG39C:
SYC 0,ASC
ERROR
LOOP
JSR @ENTLO

CG40:
SETUP 100
JSR @ENTIN
100
STACK CG40A
JSR @ISTK
CG40A
JMP CG40B
STA 3,ASC
RAND
JSR @ENTRA
INC 0,1
INC 1,2
SYC 1,1
ERROR
CG40A:
ADC# 0,1,SMR
CG40B:
ADC# 1,2,STR
ERROR
LOOP
JSR @ENTLO

10046 FCL23
01
02
03 02672 006230
04 02673 000100
05
06 02674 006221
07 02675 002705
08 02676 008402
09 02677 000412
10 02700 054002
11
12 02701 006234
13 02702 115425
14 02703 101020
15 02704 123510
16
17 02711 116015
18 02712 101002
19
20
21 02717 006231
22
23
24 02720 006230
25 02721 000040
26
27 02722 006221
28 02723 002731
29 02724 004402
30 02725 000410
31 02726 054002
32 02727 101020
33 02730 127510
34
35 02735 101003
36
37
38 02742 006231

CG41:
SETUP 100
JSR @ENTIN
100
STACK CG41A
JSR @ISTK
CG41A
JMP CG41B
STA 3,ASC
RAND
JSR @ENTRA
INC 0,3,SMR
MOV 0,0
SYC 1,0
ERROR
CG41A:
ADC# 0,3,SMR
CG41B:
MOV 0,0,ASC
ERROR
LOOP
JSR @ENTLO

CG42:
SETUP 40
JSR @ENTII
40
STACK CG42A
JSR @ISTK
CG42A
JMP CG42B
STA 3,ASC
MOV 0,0
SYC 1,1
ERROR
CG42A:
MOV 0,0,ASC
ERROR
LOOP
JSR @ENTLO

:INITIALIZE TEST.....
:INITIALIZE STACK
:FAULT ADDRESS IS CG41A
:TEST THAT AC0-3 ARE
:NOT CHANGED VIA "SCL".
:FC(AC0)=RANDOM# #
:OV OR FAIL TO CALL SYSTEM.
:FC(AC3) SHOULD BE (1)
:GREATER THEN C(AC0).
:ITERATE TEST ROUTINE.....

:INITIALIZE TEST.....
:INITIALIZE STACK
:FAULT ADDRESS IS CG42A
:TEST THAT C(CARRY)
:IS NOT CHANGED VIA SCL.
:OV OR FAIL TO CALL SYSTEM.
:ITERATE TEST ROUTINE.....

```

```

10047 ECL23
01
02
03 02763 006230
04 02744 000100
05
06 02745 006221
07 02746 002761
08 02747 020040
09 02750 040042
10 02751 004402
11 02752 004403
12 02753 054002
13 02754 135510
14
15
16 02761 006231
17
18
19 02762 006230
20 02763 000100
21
22 02764 006221
23 02765 003003
24 02766 020040
25 02767 040042
26 02770 004402
27 02771 004406
28 02772 054002
29
30 02773 006234
31 02774 105400
32 02775 131400
33 02776 137510
34
35 03003 106015
36 03004 132014
37
38
39 03011 006231
40
41
42

CG43:
SETUP 100
JSR @ENTIN
100
STACK CG43B
JSR @1STK
CG43B
LDA 0,SP
STA 0,SL
JSR *+2
JMP CG43A
STA 3,SC
SYC 1,2
ERRR
LOOP

CG43A:
ADC# 0,1,SNR
ADCM 1,2,SZR
ERRR
LOOP

CG43B:
JSR @ENTLO

CG44:
SETUP 100
JSR @ENTIN
100
STACK CG44B
JSR @1STK
CG44B
LDA 0,SP
STA 0,SL
JSR *+2
JMP CG44A
STA 3,SC
RAND
JSR @ENTRA
INC 0,1
INC 1,2
SYC 1,5
ERRR
ADCM 0,1,SNR
ADCM 1,2,SZR
ERRR
LOOP

CG44A:
ADC# 0,1,SNR
ADCM 1,2,SZR
ERRR
LOOP

CG45:
SETUP 100
JSR @ENTIN
100
STACK CG45B
JSR @1STK
CG45B
LDA 0,SP
STA 0,SL
JSR *+2
JMP CG44A
STA 3,SC
RAND
JSR @ENTRA
INC 0,3,SNR
MOVZ 0,0
SYC 2,0
ERRR
ADCM 0,3,SNR
MOV# 0,0,SZC
ERRR
LOOP
JSR @ENTLO

CG45A:
ADC# 0,3,SNR
MOV# 0,0,SZC
ERRR
LOOP

CG46:
SETUP 100
JSR @ENTIN
100
STACK CG46A
JSR @1STK
CG46A
LDA 0,SP
STA 0,SL
JSR *+2
JMP *+3
MOV0 0,0
SYC 2,1
ERRR
MOV 0,0,SNC
ERRR
LOOP
JSR @ENTLO

CG46A:
ADC# 0,1,SNR
ADCM 1,2,SZR
ERRR
LOOP
ITERATE TEST ROUTINE.....

INITIALIZE TEST.....
INITIALIZE TEST.....
INITIALIZE TEST.....
INITIALIZE TEST.....
INITIALIZE TEST.....
INITIALIZE TEST.....
ITERATE TEST ROUTINE.....
ITERATE TEST ROUTINE.....
ITERATE TEST ROUTINE.....
ITERATE TEST ROUTINE.....

```



```

10049 ECL23
01 03046 030226 CG47: LDA 2,HBEG
02 03047 024030- LDA 1,=10.
03 03070 147000- ADD 2,1
04 03071 044240 STA 1,TEM
05 03072 155400 I=C 2,3
06 03073 024227 LDA 1,BEND
07 03074 146400 SUB 2,1
08 03075 020027- LDA 0,=JSR 0,3
09 03076 041000 STA 0,0,2
10 03077 133710 RLM
11
12 03100 006230 CG47A: SETUP 4
13 03101 000004 JSR @ENTLO
14
15 03102 006221 STACK CG47A
16 03103 003100 JSR @LISTK
17 03104 020840 CG47A
18 03105 040042 LDA 0,SP
19 03106 020240 STA 0,SL
20 03107 040043 LDA 0,TF
21 03110 004402 JSR +2
22 03111 000402 JMP +2
23 03112 153510 SYNC 2,2
24 03113 115014 CG47B: ADC# 0,3,SZR
25 ERROR
26 LOOP
27 03120 006231 JSR @ENTLO
28 03121 010240 ISZ TEM
29 03122 020240 LDA 0,TEM
30 03123 024227 LDA 1,BEND
31 03124 105014 ADC# 0,1,SZR
32 03125 000753 JMP CG47A

10050 ECL23
01
02
03 03126 006230 CG4A: SETUP 100
04 03127 000100 JSR @ENTLO
05
06 03130 006221 STACK @CG49A
07 03131 103145 JSR @LISTK
08 03132 020040 @CG48A
09 03133 040042 LDA 0,SP
10 03134 004402 STA 0,SL
11 03135 000403 JMP +2
12 03136 054002 STA 3,SC
13 03137 157510 SYNC 2,3
14 ERROR
15 03140 000404 JMP CG48B
16 03145 103146 CG49A: +,1
17 03146 103147 +,1
18 03147 003150 LOOP
19
20 03150 006231 CG44B: JSR @ENTLO
21
22
23 03151 006230 CG49: SETUP 100
24 03152 000100 JSR @ENTLO
25
26 03153 006221 STACK CG49A
27 03154 003161 JSR @LISTK
28 03155 004402 CG49A
29 03156 000410 JMP CG49B
30 03157 054002 STA 3,SC
31 03160 153510 SYNC 3,0
32 ERROR
33 03165 000411 CG49A: JMP CG49C
34 03166 020226 CG49H: LDA 0,HBEG
35 03167 150010 ADI 0,0
36 03170 024040 LDA 1,SP
37 03171 106414 SUB# 0,1,SZR
38 ERROR
39
40 03176 006231 CG49C: LOOP
JSR @ENTLO

:TEST SYSTEM STACK
:FAULT TO EACH LOCATION
:IN THE BUFFER.
:FILL BUFR WITH
: (JSR 0,3) INSTRUCTIONS.
:INITIALIZE TEST....
:INITIALIZE STACK
:FAULT ADDRESS IS CG47A
:SET FAULT LOCATION TO
:POINT INTO BUFFER.
:WILL OVERFLOW HERE.....
:ITERATE TEST ROUTINE.....
:INITIALIZE TEST....
:INITIALIZE STACK
:FAULT ADDRESS IS CG49A
:TEST UPDATE OF
:(SP) BY 5 ITEMS.
:FAIL TO SYSTEM CALL.
:ITERATE TEST ROUTINE.....

```

```

10051 ECL23
01
02 03177 020226
03 03200 040240
04
05 03201 006230
06 03202 000005
07
08 03203 006221
09 03204 003515
10 03205 020240
11 03206 040040
12 03207 160010
13 03210 100010
14 03211 004402
15 03212 000410
16 03213 054002
17 03214 167510
18
19 03221 000407
20 03222 024040
21 03223 106414
22
23
24 03230 006231
25 03231 010240
26 03232 020240
27 03233 024042
28 03234 164110
29 03235 164110
30 03236 105414
31 03237 000742

LDA 0,ABEG
STA 0,TEM
SETUP 5
JSR @ENTIN
S
STACK CG50A
JSR @ISTK
CG50A
LDA 0,TEM
STA 0,SP
ADI 4,0
ADI 1,0
JSR *2
JMP CG50B
STA 3,SC
SYC 3,1
ERROR
JMP CG50C
CG50A:
CG50B:
CG50C:
JSR @ENTLO
ISZ TEM
LDA 0,TEM
LDA 1,SL
SBI 4,1
SBI 4,1
SUB# 0,1,SZR
JMP CG50

:INITIALIZE TEST....
:INITIALIZE STACK
:FAULT ADDRESS IS CG50A
:SET THE SYSTEM STACK
:POINTER AT EACH BUFFER
:LOCATION.
:OVERFLOW??
:(AC0)=CORRECT STACK POINTER.
:(AC1)=ACTUAL STACK POINTER
:ITERATE TEST ROUTINE....
:ADVANCE TO NEXT
:BUFFER LOCATION.

10052 ECL23
01
02
03 03240 006230
04 03241 000100
05
06 03242 006221
07 03243 003522
08 03244 004402
09 03245 000412
10 03246 054002
11
12 03247 006234
13 03250 105400
14 03251 173510
15
16 03256 006412
17 03257 034226
18 03260 031000
19 03261 035401
20 03262 112415
21 03263 136414
22
23
24 03270 006231
25
26
27 03271 006230
28 03272 000100
29
30 03273 006221
31 03274 003503
32 03275 004402
33 03276 000412
34 03277 054002
35
36 03300 006234
37 03301 115400
38 03302 177510
39
40 03307 000412
41 03310 030040
42 03311 025374
43 03312 031377
44 03313 106415
45 03314 156414
46
47
48 03321 006231

CG51:
CG51A:
CG51B:
CG51C:
CG52:
CG52A:
CG52B:
CG52C:

SETUP 100
JSR @ENTIN
100
STACK CG51A
JSR @ISTK
CG51A
:INITIALIZE TEST....
:FAULT ADDRESS IS CG51A
:TEST STORAGE ON
:STACK OF AC0-1.
:INITIALIZE STACK
:FAULT ADDRESS IS CG52A
:TEST STORAGE ON STACK
:OF AC0 AND AC3
:ITERATE TEST ROUTINE.....

JMP CG51A
JMP CG51B
STA 3,SC
RAND
JSR @ENTRA
IMC 0,1
SYC 3,2
ERROR
JMP CG51C
LDA 3,ABEG
LDA 2,0,3
LDA 3,1,3
SUB# 0,2,SZR
SUB# 1,3,SZR
ERROR
LOOP
JSR @ENTLO

SETUP 100
JSR @ENTIN
100
STACK CG52A
JSR @ISTK
CG52A
JSR *2
JMP CG52B
STA 3,SC
RAND
JSR @ENTRA
INC 0,3
SYC 3,3
ERROR
JMP CG52C
LDA 2,SP
LDA 1,-4,2
LDA 2,-1,2
SUB# 0,1,SZR
SUB# 2,3,SZR
ERROR
LOOP
JSR @ENTLO

:INITIALIZE TEST....
:INITIALIZE STACK
:FAULT ADDRESS IS CG52A
:TEST STORAGE ON STACK
:OF AC0 AND AC3
:ITERATE TEST ROUTINE.....

:INITIALIZE TEST....
:INITIALIZE STACK
:FAULT ADDRESS IS CG52A
:TEST STORAGE ON STACK
:OF AC0 AND AC3
:ITERATE TEST ROUTINE.....

```

```
10053 ECL23
01      SETUP 100
02      JSR @ENTIN
03      CG53:  CG53:  006230
04      03322 006230
05      03323 000100
06      STACK CG53A
07      JSR @1STK
08      CG53A:  CG53A:  006221
09      JSR +2
10      JMP CG53B
11      STA 3,SC
12      RAND
13      JSR @ENTRA
14      MOVZ 0,2
15      SYS 1,0
16      CG53A:  CG53A:  123310
17      JMP CG53
18      LDA 3,SP
19      CG53H:  CG53H:  000762
20      LDA 1,-2,3
21      CG53H:  CG53H:  03440
22      MOVZ# 0,0,SNR
23      SUB# 1,2,SZR
24      ERROR
25      LOOP
26      JSR @ENTLO
27      SETUP 100
28      JSR @ENTIN
29      CG54:  CG54:  006230
30      03353 006230
31      03354 000100
32      STACK CG54A
33      JSR @1STK
34      CG54A:  CG54A:  03355
35      JSR +2
36      JMP CG54X
37      STA 3,SC
38      MOVZ 0,0,SKP
39      @CG54A
40      LDA 0,-1
41      SYS 2,0
42      ERROR
43      CG54A:  CG54A:  000407
44      JMP CG54B
45      CG54X:  CG54X:  03373
46      SUB# 0,1,SZR
47      ERROR
48      CG54B:  CG54B:  03374
49      LOOP
50      JSR @ENTLO

10054 ECL23
01      EXECUTE A SYSTEM
02      CALL FROM EVERY
03      LOCATION IN THE BUFFER.
04      TEST FOR PROPER
05      PC STORAGE ON THE
06      STACK.
07      LDA 2,0BEG
08      LDA 1,-10,
09      ADD 2,1
10      STA 1,TEM
11      TMC 2,3
12      LDA 1,BEND
13      SUB 2,1
14      LDA 0,-SYN 3,0
15      STA 0,0,2
16      PLM
17      SETUP 5
18      JSR @ENTIN
19      CG55A:  CG55A:  006230
20      03402 006230
21      03403 024030-
22      03404 147000
23      03405 042240
24      03406 155000
25      03407 024227
26      03408 146400
27      03409 029026-
28      03410 041000
29      03412 041000
30      PLM
31      CG55A:  CG55A:  133710
32      03414 006230
33      03415 000005
34      03416 005231
35      03417 003402
36      03420 004402
37      03421 000404
38      03422 054002
39      03423 101020
40      03424 022240
41      03425 020240
42      03426 026040
43      03427 106014
44      03434 000405
45      03441 006231
46      03442 020240
47      03443 020240
48      03444 024042
49      03445 106414
50      03446 000746

:INITIALIZE TEST....
:INITIALIZE STACK
:FAULT ADDRESS IS CG53A
:CHECK STATE PUSH
:ON SCL INSTRUCTION.
:(AC0)=RANDOM #
:(AC0)=STORED ON
:THE STACK, ALSO
:CHECK FOR ZERO CARRY
:STORED ON STACK
:(AC3)=SYS STACK POINTER.
:(AC0)=CARRY
:(AC1)=C(AC2) FROM STACK.
:ITERATE TEST ROUTINE....

:INITIALIZE TEST....
:INITIALIZE STACK
:FAULT ADDRESS IS CG54A
:TEST PC STORED ON
:STACK VIA SCL INSTRUCTION.
:(AC0)=CORRECT
:(AC1)=PC VIA STACK.
:ITERATE TEST ROUTINE....

10055 ECL23
01      EXECUTE A SYSTEM
02      CALL FROM EVERY
03      LOCATION IN THE BUFFER.
04      TEST FOR PROPER
05      PC STORAGE ON THE
06      STACK.
07      LDA 2,0BEG
08      LDA 1,-10,
09      ADD 2,1
10      STA 1,TEM
11      TMC 2,3
12      LDA 1,BEND
13      SUB 2,1
14      LDA 0,-SYN 3,0
15      STA 0,0,2
16      PLM
17      SETUP 5
18      JSR @ENTIN
19      CG55A:  CG55A:  006230
20      03402 006230
21      03403 024030-
22      03404 147000
23      03405 042240
24      03406 155000
25      03407 024227
26      03408 146400
27      03409 029026-
28      03410 041000
29      03412 041000
30      PLM
31      CG55A:  CG55A:  133710
32      03414 006230
33      03415 000005
34      03416 005231
35      03417 003402
36      03420 004402
37      03421 000404
38      03422 054002
39      03423 101020
40      03424 022240
41      03425 020240
42      03426 026040
43      03427 106014
44      03434 000405
45      03441 006231
46      03442 020240
47      03443 020240
48      03444 024042
49      03445 106414
50      03446 000746

:EXECUTE A SYSTEM
:CALL FROM EVERY
:LOCATION IN THE BUFFER.
:TEST FOR PROPER
:PC STORAGE ON THE
:STACK.
:INITIALIZE TEST....
:INITIALIZE STACK
:FAULT ADDRESS IS CG55
:RETURN FROM BUFFER
:AT THIS POINT.
:EXIT TO BUFFER....
:SCL INSTRUCTION
:(AC0)+=CORRECT
:(AC1)=PC STORED VIA
:SCL STACK STUFF.
:OVERFLOW?
:ITERATE TEST ROUTINE....
```



1057 ECL23

```
01  
02  
03 03564 006230  
04 03565 000100  
05  
06 03566 006221  
07 03567 003605  
08 03570 004402  
09 03571 009406  
10 03572 054002  
11 03573 020040  
12 03574 040042  
13 03575 101040  
14 03576 167510  
15 03603 009415  
16 03604 103577  
17 03605 034226  
18 03606 021404  
19 03607 025411  
20 03610 030774  
21 03611 034025  
22 03612 112415  
23 03613 136414  
24  
25  
26  
27 03620 006231  
CG599: SETUP 100  
JSR @ENTIN  
100  
STACK CG599  
JSR @ISTK  
CG598  
JSR *+2  
CG599: JMP CG59A  
STA 3,SC  
LDA 0,SP  
STA 0,SL  
MOV 0,0  
SYC 3,1  
CG59A: ERROR  
JMP CG59C  
@CG59A  
CG598: LDA 3,AREG  
LDA 0,4,3  
LDA 1,1,1,3  
LDA 2,CG598-1  
LDA 3,@CG599  
SUB# 0,2,SNR  
SUB# 1,3,SR  
ERROR  
CG59C: LOOP  
JSR @ENTLO
```

1059 FCL23

```
01  
02  
03 03621 006230  
04 03622 000100  
05  
06 03623 006221  
07 03624 003642  
08 03625 004402  
09 03626 009407  
10 03627 054002  
11 03630 020040  
12 03631 040042  
13 03632 101021  
14 03633 003635  
15 03634 113510  
16  
17 03641 000410  
18 03642 020024  
19 03643 026040  
20 03644 106414  
21  
22  
23 03651 006231  
24  
25  
26 03652 006230  
27 03653 000100  
28  
29 03654 006221  
30 03655 003664  
31 03656 020040  
32 03657 040042  
33 03660 004402  
34 03661 009410  
35 03662 054002  
36 03663 103510  
37  
38 03670 000407  
39 03671 024040  
40 03672 106414  
41  
42  
43 03677 006231  
44  
CG600: SETUP 100  
JSR @ENTIN  
100  
STACK CG600  
JSR @ISTK  
CG600: JMP CG60A  
STA 3,SC  
LDA 0,SP  
STA 0,SL  
MOV 0,0,SKP  
*+2  
SYC 0,2  
ERROR  
CG60A: JMP CG60C  
LDA 0,@CG600  
LDA 1,@SP  
SUB# 0,1,SR  
ERROR  
CG60C: LOOP  
JSR @ENTLO  
CG61: SETUP 100  
JSR @ENTIN  
100  
STACK CG61A  
JSR @ISTK  
CG61A  
LDA 0,SP  
STA 0,SL  
JSR *+2  
JMP CG61B  
STA 3,SC  
SYC 0,0  
ERROR  
CG61A: JMP CG61C  
LDA 1,SP  
CG61B: LDA 1,SP  
SUB# 0,1,SR  
ERROR  
CG61C: LOOP  
JSR @ENTLO
```

: INITIALIZE TEST.....

: INITIALIZE STACK  
: FAULT ADDRESS IS CG60H

: TEST PC STORED ON OVERFLOW.

: FAIL TO OVERFLOW.

: C(AC0)=CORRECT PC  
: C(AC1)=PC STORED VIA OV

: ITERATE TEST ROUTINE.....

: INITIALIZE TEST.....

: INITIALIZE STACK  
: FAULT ADDRESS IS CG61A  
: IMG OVERFLO WHEN ACO  
: IS S & D IN SCL INSTR.

: OVERFLO, OR NO SYSTEM CALL.

: CONTENTS ACO MUST NOT  
: CHG IF ACO IS S & D OF SCL.

: ACO(C) DIFFERENT.

: ITERATE TEST ROUTINE.....

10059 ECL23

```

01
02
03
04
05 03700 006023-VC0:
06 03701 000324 VP
07 03702 003716 VCOA
08 03703 003723 VCOB
09 03704 000000
10
11 03705 006230 VC00:
12 03706 000100 JSR @ENTIN
13 03707 061777 VCT VCTAB
14
15 03715 000406
16 03722 000401
17 03723 103724 VC04:
18 03724 000401 VC08:
19
20
21
22 03725 006231 JSR @ENTLO
23

```

10060 ECL23

```

01
02 03726 006230 VC1:
03 03727 000100 JSR @ENTIN
04
05 03730 006023- VC1A:
06 03731 000324 VECTOR VC1B,VC1A
07 03732 003737 JSR @VEC
08 03733 003744 VP
09 03734 000000 VCTAB
10 03735 061777
11
12 03743 000410 VC1A:
13 03744 103747 VC1B:
14
15 03745 000401
16 03746 000405
17
18 03753 006231 VC1C:
19
20
21
22 03754 006230 VC2:
23 03755 000100 JSR @ENTIN
24
25 03756 006023- VC2A:
26 03757 000324 VECTOR VC2B,VC2A
27 03758 003766
28 03759 003773
29 03762 000000
30 03763 060277 INTDS
31 03764 061777 VCT VCTAB
32
33
34 03772 000406 VC2A:
35 03773 063577 VC2B:
36
37
38 04000 006231 VC2C:
39

```

```

:INITIALIZE TEST.....
:TEST ENTRY 0 DISPATCH
:ON "VCT" INSTRUCTION.
:ERROR ADDRESS
:CORRECT ADDRESS
:DEVICE 0
:INITIALIZE TEST.....
:FAIL TO DISPATCH ON VCT.
:VCT-INTA LOGIC DISPATCH
:TO WRONG LOCATION.
:ITERATE TEST ROUTINE.....
:INITIALIZE TEST.....
:DISPATCH TABLE AT LOCATION
:VCTAB... ENTRY 0 SET
:TO ADDRESS VC1A.
:OTHERS
:ITERATE TEST ROUTINE.....
:INITIALIZE TEST.....
:DISPATCH TABLE AT LOCATION
:VCTAB... ENTRY 0 SET
:TO ADDRESS VC2A.
:OTHERS
:ITERATE TEST ROUTINE.....
:VCT-INTA DISPATCH ERROR.
:WITH BIT 0(0) OF VCT
:TABLE THE I/O FLOP
:SHOULD NOT SET.
:ITERATE TEST ROUTINE.....

```

```

10061 FCL23
01
02
03 04001 030226
04 04002 155400
05 04003 024227
06 04004 146400
07 04005 020325
08 04006 041000
09 04007 133710
10 04010 020226
11 04011 101400
12 04012 040240
13 04013 062677
14
15 04014 006230
16 04015 000002
17 04016 020240
18 04017 040410
19 04020 101400
20 04021 042240
21 04022 102000
22 04023 062077
23 04024 004402
24 04025 000403
25 04026 061777
26 04030 000000
27 04030 020240
28 04031 116254
29
30
31 04036 006231
32
33 04037 010240
34 04040 024227
35 04041 020022-
36 04042 106400
37 04043 020240
38 04044 106014
39 04045 030747

VC4:
:FCLL JSR3
LDA 2,8BREG
INC 2,3
LDA 1,8END
SUB 2,1
LDA 0,JSR3
LDA 0,0,2
RLM
LDA 0,8BREG
INC 0,0
STA 0,TEM
TORST
SETUP 2
JSR 8ENTIN
2
LDA 0,TEM
STA 0,VC8C-1
INC 0,0
STA 0,8TEM,
MSKO 0
JSR +2
JMP VC8C
VCT 0
LDA 0,TEM
ADCR# 0,1,SZR
ERROR
LOOP
JSR 8ENTLO
AGAIN VC8A
ISZ TEM
-LDA 1,8END
LDA 0,-110
SUB 0,1
LDA 0,TEM
ADCR# 0,1,SZR
JMP VC8A

VC9:
:FCLL JSR3
LDA 2,8BREG
INC 2,3
LDA 1,8END
SUB 2,1
LDA 0,JSR3
LDA 0,0,2
RLM
LDA 0,8BREG
STA 0,TEM
SETUP 2
JSR 8ENTIN
2
LDA 0,TEM
STA 0,VC9C-1
JSR +2
JMP VC9C
STA 3,8TEM
JSR +2
VCT 0
ERROR
LOOP
JSR 8ENTLO
AGAIN VC9A
ISZ TEM
LDA 1,8END
LDA 0,-110
SUB 0,1
LDA 0,TEM
ADCR# 0,1,SZR
JMP VC9A

VC9A:
:FCLL JSR3
LDA 2,8BREG
INC 2,3
LDA 1,8END
SUB 2,1
LDA 0,JSR3
LDA 0,0,2
RLM
LDA 0,8BREG
STA 0,TEM
SETUP 2
JSR 8ENTIN
2
LDA 0,TEM
STA 0,VC9C-1
JSR +2
JMP VC9C
STA 3,8TEM
JSR +2
VCT 0
ERROR
LOOP
JSR 8ENTLO
AGAIN VC9A
ISZ TEM
LDA 1,8END
LDA 0,-110
SUB 0,1
LDA 0,TEM
ADCR# 0,1,SZR
JMP VC9A

VC9C:
:FCLL JSR3
LDA 2,8BREG
INC 2,3
LDA 1,8END
SUB 2,1
LDA 0,JSR3
LDA 0,0,2
RLM
LDA 0,8BREG
STA 0,TEM
SETUP 2
JSR 8ENTIN
2
LDA 0,TEM
STA 0,VC9C-1
JSR +2
JMP VC9C
STA 3,8TEM
JSR +2
VCT 0
ERROR
LOOP
JSR 8ENTLO
AGAIN VC9A
ISZ TEM
LDA 1,8END
LDA 0,-110
SUB 0,1
LDA 0,TEM
ADCR# 0,1,SZR
JMP VC9A

:ITERATE TEST ROUTINE.....
:ADVANCE TO NEXT BUFFER
:LOCATION AND TEST FOR
:END OF BUFFER-110, IF NOT
:END OF BUFFER GO TO VC9A.

```

```

10062 FCL23
01
02
03 04005 030226
04 04007 155400
05 04050 024227
06 04051 146400
07 04052 020325
08 04053 041000
09 04054 133710
10 04055 020226
11 04056 040240
12
13 04057 006230
14 04060 000002
15 04061 020240
16 04062 040407
17 04063 004402
18 04064 000412
19 04065 055240
20 04066 004402
21 04067 004072
22 04070 061777
23 04070 000000
24
25 04076 006231
26 04077 010240
27 04100 024227
28 04101 020022-
29 04102 106400
30 04103 020240
31 04104 106014
32 04105 030747
33
34 04105 000752

:VCT ORIGIN IS FIRST WORD
:OF THE BUFFER, THIS WORD
:POINTS TO EVERY OTHER WORD
:INITIALIZE TEST.....
:AFTER VCT DISPATCHES TO
:BUFFER, A JSR RETURNS
:TO MAIN PROGRAM.
:PC FROM THIS "JSR" IN
:THE BUFFER, IS CHECKED WITH
:THE VALUE IN DISPATCH
:TABLE OF THE VCT INSTRUCTION.
:VCT SHOULD GO TO ADDRESS
:IN C(CAC0).
:C(CAC3)-1 = ACTUAL LOC EXECUTED.
:ITERATE TEST ROUTINE.....
:ADVANCE TO NEXT BUFFER
:LOCATION AND TEST FOR
:END OF BUFFER-110, IF NOT
:END OF BUFFER GO TO VC9A.

```

```

10060 ECL23
01
02 SETUP 100
03 JSR @ENTIN :INITIALIZE TEST....
04 100
05 VECTOR @VC12R,VC12A
06 JSR @VEEC :DISPATCH TABLE AT LOCATION
07 VP :VCTAB... ENTRY 0 SET
08 VC12A :TO ADDRESS @VC12R. OTHERS
09 @VC12B :TO ADDRESS VC12A..
10 0
11 STACK VC12A
12 JSR @ISTK :INITIALIZE STACK
13 VC12A :FAULT ADDRESS IS VC12A
14 RAND
15 04201 096234 :C(AC0)=RANDOM #
16 04202 115400 :TEST THAT AC1-3 AND
17 04203 105040 :CARRY ARE NOT CHANGED
18 04204 061777 :BY THE VCT INSTRUCTION.
19 006653 :VCT DISPATCH OR OVERFLOW ERR.
20 ERROR
21 JMP VC12C
22 *+1
23 ADC# 1,3,SNR :C(AC1-3) SHOULD BE IN
24 MOV 0,0,SNC :ASSEMBLING ORDER WITH
25 ERROR :C(CARRY)=1.
26 LOOP
27 JSR @ENTLO :ITERATE TEST ROUTINE....
28
29 SETUP 100
30 JSR @ENTIN :INITIALIZE TEST....
31 100
32 VECTOR @VC13R,VC13A
33 JSR @VEEC :DISPATCH TABLE AT LOCATION
34 VP :VCTAB... ENTRY 0 SET
35 VC13A :TO ADDRESS @VC13R. OTHERS
36 @VC13B :TO ADDRESS VC13A.
37 0
38 STACK VC13A
39 JSR @ISTK :INITIALIZE STACK
40 VC13A :FAULT ADDRESS IS VC13A
41 RAND
42 JSR @ENTRA :C(AC0)=RANDOM #
43 INC 0,3 :TEST THAT AC1-3 AND
44 MOV 0,1 :CARRY ARE NOT CHANGED
45 VCT VCTAB :BY THE "VCT" INSTRUCTION.
46 :IS GENERATED.
47
48 ERROR :VCT DISPATCH OR OVERFLOW?
49 JMP VC13C
50 *+1
51 ADC# 1,3,SNR :C(AC1-3) SHOULD BE IN
52 MOV 0,0,SNC :ASSEMBLING ORDER.
53 ERROR :C(CARRY)=1
54 LOOP
55 JSR @ENTLO :ITERATE TEST ROUTINE....
56

```

```

10063 ECL23
01
02 SETUP 100
03 JSR @ENTIN :INITIALIZE TEST....
04 100
05 VECTOR @VC10C,VC10B
06 JSR @VEEC :DISPATCH TABLE AT LOCATION
07 VP :VCTAB... ENTRY 0 SET
08 VC10B :TO ADDRESS @VC10C. OTHERS
09 @VC10C :TO ADDRESS VC10B.
10 0
11 STACK VC10A
12 JSR @ISTK :INITIALIZE STACK
13 VC10A :FAULT ADDRESS IS VC10A
14 VCT VCTAB :FIRST TIME BIT 0(1) IN
15 006653
16
17 VC10A: ERROR
18 04125 000407 :VECTOR TABLE.
19 JMP VC100 :STACK OVERFLOW ON VCT..
20 ERROR :INCORRECT DISPATCH?
21 JMP VC100
22 *+1
23 ADC# 1,3,SNR :C(AC1-3) SHOULD BE IN
24 MOV 0,0,SNC :ASSEMBLING ORDER WITH
25 ERROR :C(CARRY)=1.
26 LOOP
27 JSR @ENTLO :ITERATE TEST ROUTINE....
28
29 SETUP 100
30 JSR @ENTIN :INITIALIZE TEST....
31 100
32 VECTOR @VC11R,VC11A
33 JSR @VEEC :DISPATCH TABLE AT LOCATION
34 VP :VCTAB... ENTRY 0 SET
35 VC11A :TO ADDRESS @VC11R. OTHERS
36 @VC11B :TO ADDRESS VC11A.
37 0
38 STACK VC11A
39 JSR @ISTK :INITIALIZE STACK
40 VC11A :FAULT ADDRESS IS VC11A
41 RAND
42 JSR @ENTRA :C(AC0)=RANDOM #
43 INC 0,3 :TEST THAT AC1-3 AND
44 MOV 0,1 :CARRY ARE NOT CHANGED
45 VCT VCTAB :BY THE "VCT" INSTRUCTION.
46 :IS GENERATED.
47
48 ERROR :VCT DISPATCH OR OVERFLOW?
49 JMP VC11C
50 *+1
51 ADC# 1,3,SNR :C(AC1-3) SHOULD BE ONE
52 MOV 0,0,SNC :GREATER THEN C(AC1).
53 ERROR :C(CARRY) SHOULD BE (0).
54 LOOP
55 JSR @ENTLO :ITERATE TEST ROUTINE....
56

```



```

10065 ECL23
01 01 04250 006230
02 02 04251 000100
03 03 04252 000100
04 04 04253 000100
05 05 04254 000100
06 06 04255 000100
07 07 04256 000100
08 08 04257 000100
09 09 04258 000100
10 10 04259 000100
11 11 04260 000100
12 12 04261 000100
13 13 04262 000100
14 14 04263 000100
15 15 04264 000100
16 16 04265 000100
17 17 04266 000100
18 18 04267 000100
19 19 04268 000100
20 20 04269 000100
21 21 04270 000100
22 22 04271 000100
23 23 04272 000100
24 24 04273 000100
25 25 04274 000100
26 26 04275 000100
27 27 04276 000100
28 28 04277 000100
29 29 04278 000100
30 30 04279 000100
31 31 04280 000100
32 32 04281 000100
33 33 04282 000100
34 34 04283 000100
35 35 04284 000100
36 36 04285 000100
37 37 04286 000100
38 38 04287 000100
39 39 04288 000100
40 40 04289 000100
41 41 04290 000100
42 42 04291 000100
43 43 04292 000100
44 44 04293 000100
45 45 04294 000100
46 46 04295 000100
47 47 04296 000100
48 48 04297 000100
49 49 04298 000100
50 50 04299 000100
51 51 04300 000100
52 52 04301 000100

:0066 ECL23
01 01 04301 006230
02 02 04302 000100
03 03 04303 000100
04 04 04304 000100
05 05 04305 006023-
06 06 04306 000324
07 07 04307 000324
08 08 04308 000355
09 09 04309 000363
10 10 04310 000000
11 11 04311 006221
12 12 04312 004355
13 13 04313 004277
14 14 04314 060277
15 15 04315 061777
16 16 04316 006653
17 17 04317 000407
18 18 04318 000407
19 19 04319 000407
20 20 04320 004363
21 21 04321 063577
22 22 04322 063577
23 23 04323 063577
24 24 04324 006231
25 25 04325 006231
26 26 04326 006230
27 27 04327 000100
28 28 04328 000100
29 29 04329 006023-
30 30 04330 000324
31 31 04331 004405
32 32 04332 004405
33 33 04333 104412
34 34 04334 104412
35 35 04335 000000
36 36 04336 006221
37 37 04337 004405
38 38 04338 024040
39 39 04339 061777
40 40 04340 006653
41 41 04341 000410
42 42 04342 000410
43 43 04343 004413
44 44 04344 030040
45 45 04345 132014
46 46 04346 132014
47 47 04347 006231
48 48 04348 006231
49 49 04349 006231
50 50 04350 006231

:INITIALIZE TEST.....
:DISPATCH TABLE AT LOCATION
:VCTAB... ENTRY 0 SET
:TO ADDRESS VC14B+. OTHERS
:TO ADDRESS VC14A.
:INITIALIZE STACK
:FAULT ADDRESS IS VC14A
:RANDOM #
:LIKE PREVIOUS TEST
:EXCEPT C(CARRY)=0
:TEST THAT AC1-3 AND
:C(CARRY) IS NOT CHANGED.
:AC3)=C(AC1)+1
:C(CARRY) SHOULD BE =(0).
:ITERATE TEST ROUTINE....
:INITIALIZE TEST.....
:DISPATCH TABLE AT LOCATION
:VCTAB... ENTRY 0 SET
:TO ADDRESS VC15A.
:INITIALIZE STACK
:FAULT ADDRESS IS VC15A
:TEST THAT "VCT"
:INSTRUCTION WILL TURN
:ON THE INTERRUPT SYSTEM
:WHEN BIT 0(1) IN TABLE
:OVERFLOW OR DISPATCH FAULT?
:VCT FAILED TO SET
:ION FLOP....
:ITERATE TEST ROUTINE.....
:INITIALIZE TEST.....
:DISPATCH TABLE AT LOCATION
:VCTAB... ENTRY 0 SET
:TO ADDRESS VC17A.
:INITIALIZE STACK
:FAULT ADDRESS IS VC17A
:CHECK THAT C(SSP)
:IS INCREMENTED BY 1
:WHEN THE MASK IS PUSHED.
:OVERFLOW OR VCT DISPATCH?
:JMP VC17C
:LD 2,SP
:AD 1,2,S7R
:ERROR
:LOOP
:JSR #ENTLO
:ITERATE TEST ROUTINE.....

```

```

10067 ECL23
01
02 04422 020226 VC18: LDA 0,8REG
03 04423 040240 STA 0,ITEM
04 VECTOR @VC18C,VC18B
05 JSR @VEVC
06 04424 06023- *DISPATCH TABLE AT LOCATION
07 04425 00324 *VCTAB... ENTRY 0 SET
08 04426 00441 *VC18B
09 04427 10446 *VC18C
10 04430 00000 *TO ADDRESS VC18B.
11 04431 06230 VC18A: SETUP 2
12 04432 00002 JSR @ENTIN
13 STACK VC18
14 04433 06221 JSR @ISTK
15 04434 00422 VC18
16 04435 02420 LDA 1,ITEM
17 04436 04040 STA 1,SP
18 04437 06177 VCT VCTAB
19 04437 006653
20
21 04445 000410 VC18B: ERROR
22 04446 00447 JMP VC18D
23 04447 030040 *+1
24 04450 132014 LDA 2,SP
25
26 04451 156414 ADC# 1,2,SZR
27 ERROR
28 04455 06231 LOOP
29 JSR @ENTLO
30 AGAIN VC18A
31 ISZ TEM
32 04457 02427 LDA 1,BEND
33 04460 020022- *LOCATION AND TEST FOR
34 04461 106400 LDA 0,I
35 04462 020240 SUB 0,I
36 04463 106014 ADC# 0,I,SZR
37 04464 000745 JMP VC18A
38
10068 ECL23
01
02 SETUP 100
03 JSR @ENTIN
04 100
05 VECTOR @VC19A,VC19A
06 JSR @VEVC
07 04467 066023- *DISPATCH TABLE AT LOCATION
08 04470 00324 *VCTAB... ENTRY 0 SET
09 04471 004501 VC19A
10 04472 104506 @VC19A
11 04473 000000 *TO ADDRESS VC19A.
12 STACK VC19A
13 JSR @ISTK
14 04474 06221 JSR @VEVC
15 04475 004501 VC19A
16 04476 024040 LDA 1,SP
17 04477 061777 VCT VCTAB
18 04477 006653
19
20 04505 000412 VC19A: ERROR
21 04506 106507 JMP VC19C
22 *+1
23 04507 030040 LDA 2,SP
24 04510 034021- *C(AC1)=ORIGINAL SYS STACK
25 04511 137000 ADD 1,I
26 04512 156414 SUB# 2,3,SZR
27 ERROR
28 04517 06231 LOOP
29 JSR @ENTLO
30
31
32
33
34
35
36
37
38

```



```

10069 ECL23
01
02 LDA 0,8REG
03 STA 0,TEM
04 VECTOR @VC20C,VC20B
05 :DISPATCH TABLE AT LOCATION
06 :VCTAB... ENTRY 0 SET
07 :TO ADDRESS @VC20C. OTHERS
08 :TO ADDRESS VC20B.
09 @VC20C
10 0
11 SETUP 2
12 JSR @ENTIN
13 2
14 STACK VC20B
15 JSR @ISTK
16 VC20B
17 LDA 1,TEM
18 STA 1,SP
19 RAND
20 JSR @ENTRA
21 INC 0,1
22 INC 1,2
23 VCT VCTAB
24 0
25 ERROR
26 04546 000414
27 JMP VC200
28 @+1
29 LDA 3,TEM
30 LDA 0,1,3
31 LDA 1,2,3
32 LDA 2,3,3
33 ADC# 0,1,SNR
34 ADC# 1,2,SNR
35 ERROR
36 LOOP
37 JSR @ENTLO
38 AGAIN VC20A
39 ISZ TEM
40 LDA 1,BEND
41 LDA 0,=110
42 SUB 0,1
43 LDA 0,TEM
44 ADC# 0,1,SNR
45 JMP VC20A

10070 ECL23
01
02 SETUP 40
03 JSR @ENTIN
04 40
05 VECTOR @VC21B,VC21A
06 :DISPATCH TABLE AT LOCATION
07 :VCTAB... ENTRY 0 SET
08 :TO ADDRESS @VC21B. OTHERS
09 :TO ADDRESS VC21A.
10 0
11 STACK VC21A
12 JSR @ISTK
13 VC21A
14 MOV0 0,0
15 VCT VCTAB
16 0
17 ERROR
18 JMP VC21C
19 @+1
20 LDA 3,SP
21 LDA 0,-1,3
22 MOV# 0,0,SNR
23 ERROR
24 LOOP
25 JSR @ENTLO
26 04623 046231

:INITIALIZE TEST....
:INITIALIZE STACK
:FAULT ADDRESS IS VC21A
:CHECK THAT C(CARRY)
:IS STORED AS A (1)
:ON THE SYSTEM STACK.
:ITERATE TEST ROUTINE.....

VC20:
VC20A:
VC20B:
VC20C:
VC20D:
VC20E:
VC20F:
VC20G:
VC20H:
VC20I:
VC20J:
VC20K:
VC20L:
VC20M:
VC20N:
VC20O:
VC20P:
VC20Q:
VC20R:
VC20S:
VC20T:
VC20U:
VC20V:
VC20W:
VC20X:
VC20Y:
VC20Z:
VC21:
VC21A:
VC21B:
VC21C:
VC21D:
VC21E:
VC21F:
VC21G:
VC21H:
VC21I:
VC21J:
VC21K:
VC21L:
VC21M:
VC21N:
VC21O:
VC21P:
VC21Q:
VC21R:
VC21S:
VC21T:
VC21U:
VC21V:
VC21W:
VC21X:
VC21Y:
VC21Z:
VC22:
VC22A:
VC22B:
VC22C:
VC22D:
VC22E:
VC22F:
VC22G:
VC22H:
VC22I:
VC22J:
VC22K:
VC22L:
VC22M:
VC22N:
VC22O:
VC22P:
VC22Q:
VC22R:
VC22S:
VC22T:
VC22U:
VC22V:
VC22W:
VC22X:
VC22Y:
VC22Z:
VC23:
VC23A:
VC23B:
VC23C:
VC23D:
VC23E:
VC23F:
VC23G:
VC23H:
VC23I:
VC23J:
VC23K:
VC23L:
VC23M:
VC23N:
VC23O:
VC23P:
VC23Q:
VC23R:
VC23S:
VC23T:
VC23U:
VC23V:
VC23W:
VC23X:
VC23Y:
VC23Z:
VC24:
VC24A:
VC24B:
VC24C:
VC24D:
VC24E:
VC24F:
VC24G:
VC24H:
VC24I:
VC24J:
VC24K:
VC24L:
VC24M:
VC24N:
VC24O:
VC24P:
VC24Q:
VC24R:
VC24S:
VC24T:
VC24U:
VC24V:
VC24W:
VC24X:
VC24Y:
VC24Z:
VC25:
VC25A:
VC25B:
VC25C:
VC25D:
VC25E:
VC25F:
VC25G:
VC25H:
VC25I:
VC25J:
VC25K:
VC25L:
VC25M:
VC25N:
VC25O:
VC25P:
VC25Q:
VC25R:
VC25S:
VC25T:
VC25U:
VC25V:
VC25W:
VC25X:
VC25Y:
VC25Z:
VC26:
VC26A:
VC26B:
VC26C:
VC26D:
VC26E:
VC26F:
VC26G:
VC26H:
VC26I:
VC26J:
VC26K:
VC26L:
VC26M:
VC26N:
VC26O:
VC26P:
VC26Q:
VC26R:
VC26S:
VC26T:
VC26U:
VC26V:
VC26W:
VC26X:
VC26Y:
VC26Z:
VC27:
VC27A:
VC27B:
VC27C:
VC27D:
VC27E:
VC27F:
VC27G:
VC27H:
VC27I:
VC27J:
VC27K:
VC27L:
VC27M:
VC27N:
VC27O:
VC27P:
VC27Q:
VC27R:
VC27S:
VC27T:
VC27U:
VC27V:
VC27W:
VC27X:
VC27Y:
VC27Z:
VC28:
VC28A:
VC28B:
VC28C:
VC28D:
VC28E:
VC28F:
VC28G:
VC28H:
VC28I:
VC28J:
VC28K:
VC28L:
VC28M:
VC28N:
VC28O:
VC28P:
VC28Q:
VC28R:
VC28S:
VC28T:
VC28U:
VC28V:
VC28W:
VC28X:
VC28Y:
VC28Z:
VC29:
VC29A:
VC29B:
VC29C:
VC29D:
VC29E:
VC29F:
VC29G:
VC29H:
VC29I:
VC29J:
VC29K:
VC29L:
VC29M:
VC29N:
VC29O:
VC29P:
VC29Q:
VC29R:
VC29S:
VC29T:
VC29U:
VC29V:
VC29W:
VC29X:
VC29Y:
VC29Z:
VC30:
VC30A:
VC30B:
VC30C:
VC30D:
VC30E:
VC30F:
VC30G:
VC30H:
VC30I:
VC30J:
VC30K:
VC30L:
VC30M:
VC30N:
VC30O:
VC30P:
VC30Q:
VC30R:
VC30S:
VC30T:
VC30U:
VC30V:
VC30W:
VC30X:
VC30Y:
VC30Z:
VC31:
VC31A:
VC31B:
VC31C:
VC31D:
VC31E:
VC31F:
VC31G:
VC31H:
VC31I:
VC31J:
VC31K:
VC31L:
VC31M:
VC31N:
VC31O:
VC31P:
VC31Q:
VC31R:
VC31S:
VC31T:
VC31U:
VC31V:
VC31W:
VC31X:
VC31Y:
VC31Z:
VC32:
VC32A:
VC32B:
VC32C:
VC32D:
VC32E:
VC32F:
VC32G:
VC32H:
VC32I:
VC32J:
VC32K:
VC32L:
VC32M:
VC32N:
VC32O:
VC32P:
VC32Q:
VC32R:
VC32S:
VC32T:
VC32U:
VC32V:
VC32W:
VC32X:
VC32Y:
VC32Z:
VC33:
VC33A:
VC33B:
VC33C:
VC33D:
VC33E:
VC33F:
VC33G:
VC33H:
VC33I:
VC33J:
VC33K:
VC33L:
VC33M:
VC33N:
VC33O:
VC33P:
VC33Q:
VC33R:
VC33S:
VC33T:
VC33U:
VC33V:
VC33W:
VC33X:
VC33Y:
VC33Z:
VC34:
VC34A:
VC34B:
VC34C:
VC34D:
VC34E:
VC34F:
VC34G:
VC34H:
VC34I:
VC34J:
VC34K:
VC34L:
VC34M:
VC34N:
VC34O:
VC34P:
VC34Q:
VC34R:
VC34S:
VC34T:
VC34U:
VC34V:
VC34W:
VC34X:
VC34Y:
VC34Z:
VC35:
VC35A:
VC35B:
VC35C:
VC35D:
VC35E:
VC35F:
VC35G:
VC35H:
VC35I:
VC35J:
VC35K:
VC35L:
VC35M:
VC35N:
VC35O:
VC35P:
VC35Q:
VC35R:
VC35S:
VC35T:
VC35U:
VC35V:
VC35W:
VC35X:
VC35Y:
VC35Z:
VC36:
VC36A:
VC36B:
VC36C:
VC36D:
VC36E:
VC36F:
VC36G:
VC36H:
VC36I:
VC36J:
VC36K:
VC36L:
VC36M:
VC36N:
VC36O:
VC36P:
VC36Q:
VC36R:
VC36S:
VC36T:
VC36U:
VC36V:
VC36W:
VC36X:
VC36Y:
VC36Z:
VC37:
VC37A:
VC37B:
VC37C:
VC37D:
VC37E:
VC37F:
VC37G:
VC37H:
VC37I:
VC37J:
VC37K:
VC37L:
VC37M:
VC37N:
VC37O:
VC37P:
VC37Q:
VC37R:
VC37S:
VC37T:
VC37U:
VC37V:
VC37W:
VC37X:
VC37Y:
VC37Z:
VC38:
VC38A:
VC38B:
VC38C:
VC38D:
VC38E:
VC38F:
VC38G:
VC38H:
VC38I:
VC38J:
VC38K:
VC38L:
VC38M:
VC38N:
VC38O:
VC38P:
VC38Q:
VC38R:
VC38S:
VC38T:
VC38U:
VC38V:
VC38W:
VC38X:
VC38Y:
VC38Z:
VC39:
VC39A:
VC39B:
VC39C:
VC39D:
VC39E:
VC39F:
VC39G:
VC39H:
VC39I:
VC39J:
VC39K:
VC39L:
VC39M:
VC39N:
VC39O:
VC39P:
VC39Q:
VC39R:
VC39S:
VC39T:
VC39U:
VC39V:
VC39W:
VC39X:
VC39Y:
VC39Z:
VC40:
VC40A:
VC40B:
VC40C:
VC40D:
VC40E:
VC40F:
VC40G:
VC40H:
VC40I:
VC40J:
VC40K:
VC40L:
VC40M:
VC40N:
VC40O:
VC40P:
VC40Q:
VC40R:
VC40S:
VC40T:
VC40U:
VC40V:
VC40W:
VC40X:
VC40Y:
VC40Z:
VC41:
VC41A:
VC41B:
VC41C:
VC41D:
VC41E:
VC41F:
VC41G:
VC41H:
VC41I:
VC41J:
VC41K:
VC41L:
VC41M:
VC41N:
VC41O:
VC41P:
VC41Q:
VC41R:
VC41S:
VC41T:
VC41U:
VC41V:
VC41W:
VC41X:
VC41Y:
VC41Z:
VC42:
VC42A:
VC42B:
VC42C:
VC42D:
VC42E:
VC42F:
VC42G:
VC42H:
VC42I:
VC42J:
VC42K:
VC42L:
VC42M:
VC42N:
VC42O:
VC42P:
VC42Q:
VC42R:
VC42S:
VC42T:
VC42U:
VC42V:
VC42W:
VC42X:
VC42Y:
VC42Z:
VC43:
VC43A:
VC43B:
VC43C:
VC43D:
VC43E:
VC43F:
VC43G:
VC43H:
VC43I:
VC43J:
VC43K:
VC43L:
VC43M:
VC43N:
VC43O:
VC43P:
VC43Q:
VC43R:
VC43S:
VC43T:
VC43U:
VC43V:
VC43W:
VC43X:
VC43Y:
VC43Z:
VC44:
VC44A:
VC44B:
VC44C:
VC44D:
VC44E:
VC44F:
VC44G:
VC44H:
VC44I:
VC44J:
VC44K:
VC44L:
VC44M:
VC44N:
VC44O:
VC44P:
VC44Q:
VC44R:
VC44S:
VC44T:
VC44U:
VC44V:
VC44W:
VC44X:
VC44Y:
VC44Z:
VC45:
VC45A:
VC45B:
VC45C:
VC45D:
VC45E:
VC45F:
VC45G:
VC45H:
VC45I:
VC45J:
VC45K:
VC45L:
VC45M:
VC45N:
VC45O:
VC45P:
VC45Q:
VC45R:
VC45S:
VC45T:
VC45U:
VC45V:
VC45W:
VC45X:
VC45Y:
VC45Z:

```

```

10071 ECL23
01
02 04624 020226 VC22: LDA 0,8865
03 04625 040840 STA 0,TEM
04
05 04626 006023- VECTOR @VC22C,VC22B
06 04627 000324 :DISPATCH TABLE AT LOCATION
07 04630 004647 VP :VCTAB... ENTRY 0 SET
08 04631 104654 VC22B :TO ADDRESS @VC22C. OTHERS
09 04632 000000 @VC22C :TO ADDRESS VC22B.
10
11 04633 006230 VC22A: SETUP 1
12 04634 000001 JSR @ENTIN
13
14 04635 006231 STACK VC22B
15 04636 004647 JSR @ISTK
16 04637 024240 VC22B :INITIALIZE STACK
17 04640 044040 LDA 1,TEM :FAULT ADDRESS IS VC22B
18 04641 006234 STA 0,0 :TEST THE ABILITY OF
19 04642 040900 MOVZL 0,3 :THE VCT INSTRUCTION TO
20 04643 115120 MOVPR 3,3 :STORE AC3 AND LOC 0.
21 04644 175220 VCT VCTAB :C(AC0)=RANDOM #
22 04645 061777 :THESE
23 04646 066653 :ITEMS ARE MADE RANDOM
24 :AND STORED AT EACH
25 :LOCATION OF THE BUFFER.
26 04653 000416 VC228: ERROR
27 04654 104655 @+1 JUMP VC22D
28 04655 030240 LDA 2,TEM :C(AC2)=ORIGINAL C(SP)
29 04656 025004 LDA 1,4,2 :C(AC1)=AC3 ON THE STACK.
30 04657 035005 LDA 3,5,2 :C(AC3)=LOC 0 ON THE STACK.
31 04660 020940 LDA 0,SP :C(AC0)=CURRENT STACK
32 04661 160110 SBI 4,0 :POINTER -6.
33 04662 120110 SUB# 0,2,SNR
34 04663 112415 SUB# 3,1,SZR
35 04664 166414 ERROR
36
37
38 04671 006231 VC22D: JSR @ENTLO
39 AGAIN VC22A
40
41 04672 010240 ISZ TEM
42 04673 024227 LDA 1,BEND :ADVANCE TO NEXT BUFFER
43 04674 020022- LDA 0,=110 :LOCATION AND TEST FOR
44 04675 106400 SUB 0,1 :PEND OF BUFFER-110. IF NOT
45 04676 020240 LDA 0,TEM :FEND OF BUFFER GO TO VC22A.
46 04677 106014 ADC# 0,1,SZR
47 04700 000733 JMP VC22A

```

```

10072 FCL23

```

```

01
02 04701 006230 VC23: SETUP 100
03 04702 000100 JSR @ENTIN
04
05 04703 006023- VECTOR @VC23B,VC23A
06 04704 000324 :DISPATCH TABLE AT LOCATION
07 04705 004717 VP :VCTAB... ENTRY 0 SET
08 04706 104724 VC23A :TO ADDRESS @VC23B. OTHERS
09 04707 000000 @VC23B :TO ADDRESS VC23A.
10
11 04710 006221 STACK VC23A
12 04711 004717 JSR @ISTK
13
14 04712 006234 VC23A :INITIALIZE STACK
15 04713 040005 JSR @ENTRA :FAULT ADDRESS IS VC23A
16 04714 105000 STA 0,PK :C(AC0)=RANDOM #
17 04715 061777 MOV 0,1 :STORE A RANDOM NUMBER IN
18 :THE VCT INSTRUCTION TO :C(MK) AND C(AC1) THE MASK
19 :STORE AC3 AND LOC 0. :REGISTER SHOULD BE PLACED
20 04723 000410 VC23A: ERROR
21 04724 004725 JUMP VC23C
22 04725 032000 LDA 2,@SP
23 04726 146414 SUB# 2,1,SZR
24
25
26
27 04733 006231 VC23C: LOOP
28 JSR @ENTLO
29

```

```

:ITERATE TEST ROUTINE....

```

```

:ITERATE TEST ROUTINE....
:ADVANCE TO NEXT BUFFER
:LOCATION AND TEST FOR
:FEND OF BUFFER-110. IF NOT
:FEND OF BUFFER GO TO VC22A.

```

10073 FCL23

```

01
02 04734 020226 VC24: LDA 0, RBEG
03 04735 040240 STA 0, TEM
04 VECTOR @VC248, VC24A
05 JSR @VEC
06 :DISPATCH TABLE AT LOCATION
07 :VCTAB, ENTRY 0 SET
08 :TO ADDRESS @VC24B, OTHERS
09 :TO ADDRESS VC24A.
10
11 VC244: SETUP 1
12 JSR @ENTLN
13
14 STACK VC24A
15 JSR @ISTK
16 VC24A
17 LDA 0, TEM
18 STA 0, SP
19 RAND
20 JSR @ENTRA
21 STA 0, MK
22 MOV 0, 1
23 VCT VCTAB
24
25 VC24A: ERROR
26 JMP VC24C
27
28 VC24B: LDA 3, TEM
29 LDA 2, @SP
30 SUB# 2, 1, SZR
31 ERROR
32 LOOP
33 JSR @ENTLO
34 AGAIN VC24A
35 ISZ TEM
36 LDA 1, @ENO
37 LDA 0, @I10
38 SUB 0, 1
39 LDA 0, TEM
40 ADC# 0, 1, SZR
41 JMP VC244

```

10074 ECL25

```

01
02 SETUP 100
03 JSR @ENTLN
04
05 VC25: VECTOR @VC258, VC25A
06 JSR @VEC
07 :DISPATCH TABLE AT LOCATION
08 :VCTAB, ENTRY 0 SET
09 :TO ADDRESS @VC25B, OTHERS
10 :TO ADDRESS VC25A.
11
12 STACK VC25A
13 JSR @ISTK
14 VC25A
15 RAND
16 :C(CAC0)=RANDOM #
17 :RANDOM # TO MASK
18 :LOCATION, THIS NUMBER
19 :SHOULD BE INCLUSIVE
20
21 :LOADED WITH 0, AND LOADED TO ACO.
22 :OVERFLOW OR DISPATCH ERROR?
23
24 VC25A: ERROR
25 JMP VC25C
26 @+2
27 SUB# 0, 1, SZR
28 ERROR
29 LOOP
30 JSR @ENTLO
31
32 VC26: VECTOR @VC268, VC26A
33 JSR @VEC
34 VCTAB... ENTRY 0 SET
35 @VC26B
36
37 SETUP 400
38 JSR @ENTLN
39
40 STACK VC26A
41 JSR @ISTK
42 VC26A
43 RAND
44 :C(CAC0)=RANDOM #
45 :TEST VCT INSTRUCTION TO
46 :LOAD ACO WITH NEW
47 :MASK.
48
49 :OVERFLOW OR DISPATCH?
50 :C(CAC1)=NEW MASK BITS
51 :C(CAC1)=SWAPPED= OLD MASK
52 :C(CAC3)=CORRECT INCLUSIVE
53 :OR OLD/NEW.
54 :C(CAC0)=NEW MASK VIA VCT.
55
56 ERROR
57 JMP VC26C
58 @+2
59 SUB# 0, 3, SZR
60 ERROR
61 LOOP
62 JSR @ENTLO
63
64 VC26C: ITERATE TEST ROUTINE.....

```

10075 FCL23

```
01
02 05074 030226 VC27: LDA 2,88EG ;VCT TABLE ENTRY 0 POINTS
03 05075 15500 INC 2,3 ;TO EACH LOCATION OF THE
04 05076 054240 STA 3,TEM ;BUFFER, THE WORD AT THIS POINT
05 05077 024227 LDA 1,8END ;WILL RETURN CNTL TO THE
06 05100 146001 SUB 2,1,SKP ;PROGRAM AT LOCATION VC27C.
07 05101 005154 VC27E ;THE NEXT BUFFER LOCATION
08 05102 020777 LDA 0,-1 ;CONTAINS A RANDOM MASK VALUE.
09 05103 041000 STA 0,0,2 ;OLD MASK IN C(MK) AND
10 05104 133710 BLM ;NEW MASK ARE CHECKED IN AC0.
11 VECTOR VC278,VC278
12 05105 006023- ;DISPATCH TABLE AT LOCATION
13 05106 009324 VP ;VCTAB... ENTRY 0 SET
14 05107 005134 VC278 ;TO ADDRESS VC278. OTHERS
15 05110 005134 VC278 0 ;TO ADDRESS VC278.
16 05111 000000
17 05112 006230 VC27A: SETUP 2
18 05113 000002 JSR @ENTIN
19 05114 020240 LDA 0,TEM
20 05115 103260 ADDR 0,0
21 05116 042020- STA 0,@VCTAB
22 STACK VC278
23 JSR @ISTK
24 05117 006221 VC278
25 05120 005134 RANO
26 JSR @ENTRA
27 05121 006234 JSR *2
28 05122 004402 JMP VC27D
29 05123 000416 VC27C: LDA 2,TEM
30 05124 030240 STA 3,0,2
31 05125 055000 STA 0,1,2
32 05126 041001 MOV8,0,1
33 05127 105300 STA 1,MK
34 05130 044005 MOV 0,3
35 05131 115000 VCT VCTAB
36 05132 061777
37 05133 006453
38
39 ERROR
40 05140 000420 VC278: JMP VC27F
41 05141 030240 VC27D: LDA 2,TEM
42 05142 134410 IOR 1,3
43 05143 116414 SUB# 0,3,SZR
44 ERROR
45 05150 030240 LDA 2,TEM
46 05151 021002 LDA 0,2,2
47 05152 041000 STA 0,0,2
48 05153 000405 JMP VC27E
49 ERROR
50 VC27F: LOOP
51 05160 006231 JSR @ENTLO
52 AGAIN VC27A
53 05161 010240 ISZ TEM
54 05162 024227 LDA 1,8END
55 05163 020022- LDA 0,=110
56 05164 106400 SUB 0,1
57 05165 020240 LDA 0,TEM
58 05166 106014 ADC# 0,1,SZR
59 05167 000723 JMP VC27A
```

10074 FCL23

```
01
02 SETUP 100 ;INITIALIZE TEST....
03 JSR @ENTIN
100 VECTOR @VC328,VC32A
04 05170 006230 ;DISPATCH TABLE AT LOCATION
05 JSR @VEC ;VCTAB... ENTRY 0 SET
06 05171 000100 ;TO ADDRESS @VC328. OTHERS
07 05172 006023- ;TO ADDRESS VC32A.
08 05173 000324 VP
09 05174 005203 VC32A
10 05175 105210 @VC32B
11 0 ;STACK VC32A
12 JSR @ISTK ;INITIALIZE STACK
13 05200 005203 ;FAULT ADDRESS IS VC32A
14 05201 061777 VCT VCTAR
15 05202 006453 ;OVERFLOW?
16 ERROR
17 05207 000411 VC32A: JMP VC32C
18 05210 005211 ;CHECK THE PC STORED
19 05211 024777 *+1 ;IN C(AC3) VIA "VCT".
20 05212 104110 SBI 1,1 ;C(AC1)=CORRECT
21 05213 132414 SUB# 1,2,SZR ;C(AC3)=VIA VCT INSTRUCTION.
22 ERROR
23 LOOP
24 05220 006231 VC32C: LOOP
25 JSR @ENTLO ;ITERATE TEST ROUTINE....
26
27 SETUP 100 ;INITIALIZE TEST....
28 JSR @ENTIN
100 VECTOR @VC338,VC33A
29 JSR @VEC ;DISPATCH TABLE AT LOCATION
30 05223 006023- ;VCTAB... ENTRY 0 SET
31 05224 000324 VC33A ;TO ADDRESS @VC338. OTHERS
32 05225 005234 @VC33B ;TO ADDRESS VC33A.
33 05226 105234 0
34 05227 000000 ;STACK VC33A
35 JSR @ISTK ;INITIALIZE STACK
36 05230 006221 ;FAULT ADDRESS IS VC33A
37 05231 005234 VC33A
38 05232 061777 VCT VCTAR
39 ERROR ;OVERFLOW?
40 05240 000411 VC33A: JMP VC33C
41 05241 105242 ;CHECK PC STORED IN
42 05242 024777 LDA 1,-1 ;C(AC2) VIA "VCT".
43 05243 104110 SBI 1,1 ;LIKE PREVIOUS TEST
44 05244 132534 SUBZL# 1,2,SZR ;EXCEPT BIT 0(1) AT VC33B.
45 ERROR
46 LOOP
47 05251 006231 VC33C: LOOP
48 JSR @ENTLO ;ITERATE TEST ROUTINE....
49
```

```

10077 ECL23
01
02 05252 030226 VC34:          LDA 2, RBEG
03 05253 155400              INC 2, 3
04 05254 054240              STA 3, TEM
05          SETUP 2
06 05255 006230              JSR @ENTIN
07 05256 000002              2
08          STACK VC34D
09 05277 006221              JSR @ISTK
10 05260 005273              VC34D
11 05261 020240              LDA 0, TEM
12 05262 103240              ADDR 0, 0
13 05263 042017-            STA 0, @=VCTAR
14 05264 004402              JSR *+2
15 05265 000404              JMP VC34C
16 05266 056240              STA 3, @TEM
17 05267 061777              VCT VCTAR
18
19
20 05271 024240 VC34C:          LDA 1, TEM
21 05272 132414              SUB# 1, 2, SZR
22          ERROR
23          LOOP
24 05277 006231              JSR @ENTLO
25          AGAIN VC34A
26 05300 010240              ISZ TEM
27 05301 024227              LDA 1, @BEND
28 05302 020022-            LDA 0, =110
29 05303 106400              SUB 0, 1
30 05304 020240              LDA 0, TEM
31 05305 106014              ADC# 0, 1, SZR
32 05306 000747              JMP VC34A
33
10078 ECL23
01
02          AFILL JSR3
03          LDA 2, @BEG
04          INC 2, 3
05          LDA 1, @MEMO
06          SUB 2, 1
07          LDA 0, JSR3
08          STA 0, 0, 2
09          RLM
10          LDA 2, @BEG
11          INC 2, 3
12          STA 3, TEM
13          SETUP 2
14          JSR @ENTIN
15          2
16          STACK VC35C
17          JSR @ISTK
18          VC35C
19          LDA 0, TEM
20          ADDR 0, 0
21          STA 0, @=VCTAR
22          JSR *+2
23          JMP VC35D
24          LDA 2, TEM
25          ADI 1, 2
26          STA 2, -1, 2
27          MOV 3, 2
28          VCT VCTAH
29
30          ERROR
31          LDA 2, TEM
32          ADI 1, 2
33          ADC# 2, 3, SZR
34          ERROR
35          LDA 0, =JSR 0, 2
36          STA 0, @TEM
37          LOOP VC35E:
38          JSR @ENTLO
39          AGAIN VC35A
40          ISZ TEM
41          LDA 1, @BEND
42          LDA 0, =110
43          SUB 0, 1
44          LDA 0, TEM
45          ADC# 0, 1, SZR
46          JMP VC35A

```

```

: FILL THE BUFFER WITH
: JSR3.

```

```

: CONTENTS OF VCTARLE
: ENTRY POINTS TO A BUFFER
: LOCATION. THIS BUFFER
: LOCATION POINTS TO THE NEXT.
: INITIALIZE TEST....

```

```

: INITIALIZE STACK
: FAULT ADDRESS IS VC35C
: WITH BIT 0(1). POINT VCT
: TABLE TO BUFFER.

```

```

: SET BUFFER ENTRY TO
: CURRENT LOCATION+1.
: THIS IS A JSR 0,3 INST.

```

```

: C(AC2)*1=CURRENT PC
: C(AC3)=PC VIA JSR IN BUFFER.
: RESTORE BUFFER
: ITERATE TEST ROUTINE....

```

```

: ADVANCE TO NEXT BUFFER
: LOCATION AND TEST FOR
: END OF BUFFER-110. IF NOT
: END OF BUFFER GO TO VC35A.

```

```

10078 ECL23

```

```

01
02
03 05307 030226 VC35:          LDA 2, RBEG
04 05310 155400              INC 2, 3
05 05311 024227              LDA 1, @MEMO
06 05312 146400              SUB 2, 1
07 05313 020325              LDA 0, JSR3
08 05314 041900              STA 0, 0, 2
09 05315 133710              RLM
10 05316 030226              LDA 2, @BEG
11 05317 155400              INC 2, 3
12 05320 054240              STA 3, TEM
13          SETUP 2
14 05321 006230              JSR @ENTIN
15 05322 000002              2
16          STACK VC35C
17 05323 006221              JSR @ISTK
18 05324 005340              VC35C
19 05325 020240              LDA 0, TEM
20 05326 103240              ADDR 0, 0
21 05327 042016-            STA 0, @=VCTAR
22 05330 004402              JSR *+2
23 05331 004413              JMP VC35D
24 05332 030240              LDA 2, TEM
25 05333 110010              ADI 1, 2
26 05334 051377              STA 2, -1, 2
27 05335 171000              MOV 3, 2
28 05336 061777              VCT VCTAH
29
30          ERROR
31 05344 030240 VC35D:          LDA 2, TEM
32 05345 110010              ADI 1, 2
33 05346 156014              ADC# 2, 3, SZR
34          ERROR
35 05353 020015-            LDA 0, =JSR 0, 2
36 05354 042240              STA 0, @TEM
37          LOOP VC35E:
38 05355 006231              JSR @ENTLO
39          AGAIN VC35A
40 05356 010240              ISZ TEM
41 05357 024227              LDA 1, @BEND
42 05360 020022-            LDA 0, =110
43 05361 106400              SUB 0, 1
44 05362 020240              LDA 0, TEM
45 05363 106014              ADC# 0, 1, SZR
46 05364 000735              JMP VC35A

```

```

: VCT TABLE ENTRY 0
: POINTS TO EACH WORD
: OF THE BUFFER.

```

```

: INITIALIZE TEST....

```

```

: INITIALIZE STACK
: FAULT ADDRESS IS VC34D
: SETUP VCT TABLE
: WITH BIT 0(1). POINT
: TO BUFFER.

```

```

: SET BUFFER FOR A
: RETURN TO THE TEST

```

```

: PROGRAM.
: C(AC1)=CORRECT PC
: C(AC2)=PC VIA VCT INST.

```

```

: ITERATE TEST ROUTINE....

```

```

: ADVANCE TO NEXT BUFFER
: LOCATION AND TEST FOR
: END OF BUFFER-110. IF NOT
: END OF BUFFER GO TO VC34A.

```

```

10077 ECL23

```

```

01
02 05252 030226 VC34:          LDA 2, RBEG
03 05253 155400              INC 2, 3
04 05254 054240              STA 3, TEM
05          SETUP 2
06 05255 006230              JSR @ENTIN
07 05256 000002              2
08          STACK VC34D
09 05277 006221              JSR @ISTK
10 05260 005273              VC34D
11 05261 020240              LDA 0, TEM
12 05262 103240              ADDR 0, 0
13 05263 042017-            STA 0, @=VCTAR
14 05264 004402              JSR *+2
15 05265 000404              JMP VC34C
16 05266 056240              STA 3, @TEM
17 05267 061777              VCT VCTAR
18
19
20 05271 024240 VC34C:          LDA 1, TEM
21 05272 132414              SUB# 1, 2, SZR
22          ERROR
23          LOOP
24 05277 006231              JSR @ENTLO
25          AGAIN VC34A
26 05300 010240              ISZ TEM
27 05301 024227              LDA 1, @BEND
28 05302 020022-            LDA 0, =110
29 05303 106400              SUB 0, 1
30 05304 020240              LDA 0, TEM
31 05305 106014              ADC# 0, 1, SZR
32 05306 000747              JMP VC34A
33

```

```

10079 ECL23
01
02 05365 020226 LDC 0,VBREG
03 05366 160010 ADI 4,0
04 05367 160010 ADI 4,0
05 05370 040240 STA 0,1EM
06 B1LL J8R3
07 05371 050226 LDA 2,VBREG
08 05372 155400 INC 2,3
09 05373 024227 LDC 1,BEND
10 05374 146400 SUB 2,1
11 05375 020325 LDA 0,JSR3
12 05376 041000 STA 0,0,2
13 05377 133710 BLM
14 SETUP 2
15 05400 06230 VC37:
16 05401 000002 JSR 0,ENTIN
17 VECTOR 0VC37C,VC37B
18 05402 06023- JSR 0,VEC
19 05403 000324 VP
20 05404 095417 VC37R
21 05405 105424 0VC37C
22 05406 000000 0
23 STACK VC37B
24 JSR 0,ISTK
25 05410 05417 VC37B
26 05411 020240 LDA 0,TEM
27 05412 040412 STA 0,VC37C
28 05413 044402 JSR *2
29 05414 009411 VC37A: JMP VC37D
30 05415 061777 VCT VCTAB
31 05416 061777 0
32 05423 000410 VC37B: ERROR
33 05424 000000 JMP VC37E
34 05425 020240 VC37C: 0
35 05426 115014 LDA 0,TEM
36 05427 020240 ADC# 0,3,SZR
37 ERROR
38 LOOP
39 05433 006231 VC37E:
40 AGAIN VC37
41 ISZ TEM
42 05434 010200 LDA 1,BEND
43 05435 024227 LDC 0,=110
44 05436 020022- SUB 0,1
45 05440 020240 LDA 0,TEM
46 05441 106014 ADC# 0,1,SZR
47 05442 000736 JMP VC37
48

:0080 ECL23
01
02 SETUP 100
03 JSR 0,ENTIN
04 100
05 VECTOR 0VC38A,VC38B
06 JSR 0,VEC
07 05445 06023- VP
08 05446 000324 VC38B
09 05447 105465 0VC38A
10 05450 000000 0
11 STACK VC38C
12 JSR 0,ISTK
13 05452 004221 VC38C
14 05453 005466 VC38C
15 05454 020000 LDA 0,SP
16 05455 040042 STA 0,SL
17 05456 061777 VCT VCTAB
18 ERROR
19 05464 000402 VC38B:
20 05465 005460 VC38A:
21 05466 006231 VC38C:
22 05466 006231 JSR 0,ENTLO
23
24 SETUP 100
25 JSR 0,ENTIN
26 100
27 VECTOR 0VC39A,0VC39A
28 JSR 0,VEC
29 05471 06023- VP
30 05472 000324 0VC39A
31 05473 105504 0VC39A
32 05474 105504 0
33 STACK VC39B
34 JSR 0,ISTK
35 05476 06221 VC39B
36 05477 005511 LDA 0,SP
37 05500 020040 STA 0,SL
38 05501 061777 VCT VCTAB
39 @.*1
40 05502 105505 VC39A:
41 ERROR
42 LOOP
43 JSR 0,ENTLO
44 05511 006231 VC39B:
45
46
47
48

:INITIALIZE TEST.....
:DISPATCH TABLE AT LOCATION
:VC38B... ENTRY 0 SET
:TO ADDRESS 0VC38A, OTHERS
:TO ADDRESS VC38B.

:INITIALIZE TEST.....
:DISPATCH TABLE AT LOCATION
:VC39A... ENTRY 0 SET
:TO ADDRESS 0VC39A, OTHERS
:TO ADDRESS 0VC39A.

:INITIALIZE TEST.....
:FAULT ADDRESS IS VC39B
:TEST VCT OVERFLOW
:ON VCT PUSH OF 6 WORDS.

:ITERATE TEST ROUTINE.....

:INITIALIZE TEST.....
:DISPATCH TABLE AT LOCATION
:VC38B... ENTRY 0 SET
:TO ADDRESS 0VC37C, OTHERS
:TO ADDRESS VC37B.

:INITIALIZE TEST.....
:FAULT ADDRESS IS VC37B
:SETUP FOR VCT DISPATCH
:TO BUFFER.

:INITIALIZE TEST.....
:FAULT ADDRESS IS VC37B
:SETUP FOR VCT DISPATCH
:TO BUFFER.

:ADVANCE TO NEXT BUFFER
:LOCATION AND TEST FOR
:END OF BUFFER=110. IF NOT
:END OF BUFFER GO TO VC37.

:ITERATE TEST ROUTINE.....

```



100R1 FCL23

```

01 02
03 04 05512 006230
05 05513 000100
06 07 05514 006023-
08 05515 000324
09 05516 005327
10 05517 105334
11 05520 000000
12 13 05521 006221
14 05522 005327
15 05533 102400
16 05534 000043
17 05525 061777
18 106653
19
20 VTOA:
21 05533 000410
22 05534 005335
23 05535 020043
24 05536 101005
25
26 VT0C:
27 05533 006231
28
29
30 05544 006230
31 05545 000100
32
33 05546 006023-
34 05547 000324
35 05550 005327
36 05551 105367
37 05552 000000
38
39 05553 006221
40 05554 005562
41
42 05555 006234
43 05556 105400
44 05557 115020
45 05558 061777
46 106653
47
48 05566 000410
49 05567 005570
50 05570 166015
51 05571 101002
52
53
54 05576 006231
55

```

100R2 FCL23

```

01 02
03 04 05577 004230
05 05500 000100
06
07 05601 006023-
08 05602 000324
09 05603 005615
10 05604 105622
11 05605 000000
12
13 05606 006231
14 05607 005615
15 05610 024004
16 05611 164010
17 05612 125440
18 05613 061777
19 106653
20
21 05621 000411
22 05622 005623
23 05623 030040
24 05624 132415
25 05625 101003
26
27
28 05632 006231
29

```

```

:INITIALIZE TEST.....
:DISPATCH TABLE AT LOCATION
:VCTAB... ENTRY 0 SET
:TO ADDRESS VT0B, OTHERS
:TO ADDRESS VTOA.
:INITIALIZE STACK
:FAULT ADDRESS IS VTOA
:VECTOR SHOULD LOAD
:C(SF) WITH C(1SF) BECAUSE
:OF INDIRECT BIT IN VCT
:INSTRUCTION
:OVERFLOW?
:ITERATE TEST ROUTINE.....
:INITIALIZE TEST.....
:DISPATCH TABLE AT LOCATION
:VCTAB... ENTRY 0 SET
:TO ADDRESS VT1B, OTHERS
:TO ADDRESS VT1A.
:INITIALIZE STACK
:FAULT ADDRESS IS VT1A
:RANDOM #
:CHECK THAT VCT INSTRUCTION
:WILL NOT CHANGE TIME
:STATE OF AC1-3 OR CARRY.
:OVERFLOW?
:C(AC1) SHOULD EQUAL C(AC3)+1
:C(CARRY) SHOULD BE ZERO.
:ITERATE TEST ROUTINE.....

```

```

VT0:
VT0A:
VT0B:
VT0C:
VT1:
VT1A:
VT1B:
VT1C:
VT2:
VT2A:
VT2B:
VT2C:

```

```

SETUP 100
JSR @ENTIN
100
VECTOR @VT0B, VTOA
JSR @VEEC
VP
VT0A
@VT0B
0
STACK VTOA
JSR @ISTK
VTOA
SUB 0,0
STA 0,SF
VCT 180+VCTAB
ERROR
JMP VT0C
+1
LDA 0,SF
MOV 0,0,SNR
ERROR
JSR @ENTLO
SETUP 100
JSR @ENTIN
100
VECTOR @VT1B, VT1A
JSR @VEEC
VT1A
@VT1B
0
STACK VT1A
JSR @ISTK
VT1A
RAND
JSR @ENTRA
INC 0,1
MOVZ 0,3
VCT 180+VCTAB
ERROR
JMP VT1C
+1
ADCA 3,1,SNR
MOV 0,0,SZC
ERROR
JSR @ENTLO

```

```

VT2:
VT2A:
VT2B:
VT2C:

```

```

:INITIALIZE TEST.....
:DISPATCH TABLE AT LOCATION
:VCTAB... ENTRY 0 SET
:TO ADDRESS @VT2B, OTHERS
:TO ADDRESS VT2A.
:INITIALIZE STACK
:FAULT ADDRESS IS VT2A
:TEST ABILITY OF VCT INST-
:RUCTION TO UPDATE C(SF)
:BY 5 ITEMS, C(40)-C(43)
:AND MASK NDR0.
:ERROR ON OVERFLOW?
:ITERATE TEST ROUTINE.....

```

```

VT2:
VT2A:
VT2B:
VT2C:

```

```

SETUP 100
JSR @ENTIN
100
VECTOR @VT2B, VT2A
JSR @VEEC
VP
VT2A
@VT2B
0
STACK VT2A
JSR @ISTK
VT2A
LDA 1,ISP
ADI 4,1
INCO 1,1
VCT 180+VCTAB
ERROR
JMP VT2C
+1
LDA 2,SP
SUB# 1,2,SNR
MOV 0,0,SNR
ERROR
JSR @ENTLO

```

```

VT2:
VT2A:
VT2B:
VT2C:

```

```

:INITIALIZE TEST.....
:DISPATCH TABLE AT LOCATION
:VCTAB... ENTRY 0 SET
:TO ADDRESS @VT2B, OTHERS
:TO ADDRESS VT2A.
:INITIALIZE STACK
:FAULT ADDRESS IS VT2A
:TEST ABILITY OF VCT INST-
:RUCTION TO UPDATE C(SF)
:BY 5 ITEMS, C(40)-C(43)
:AND MASK NDR0.
:ERROR ON OVERFLOW?
:ITERATE TEST ROUTINE.....

```

```

VT2:
VT2A:
VT2B:
VT2C:

```

10083 ECL23

```

01
02
03
04 05633 006230
05 05634 000100
06 05635 006023-
07 05636 000324
08 05637 005654
09 05640 105661
10 05641 000000
11 05642 006221
12 05643 005654
13 05644 006234
14 05645 105400
15 05646 131400
16 05647 040040
17 05650 044041
18 05651 050042
19 05652 061777
20 05653 106653
21 05660 000414
22 05661 005662
23 05662 034226
24 05663 021400
25 05664 025401
26 05665 031402
27 05666 106015
28 05667 132014
29
30
31
32
33
34
35 05674 006231
36

```

VT3:                   :INITIALIZE TEST.....

VT3A:                   :DISPATCH TABLE AT LOCATION

VT3B:                   :VCTAB... ENTRY 0 SET

VT3C:                   :TO ADDRESS @VT3B. OTHERS

VT3D:                   :TO ADDRESS VT3A.

VT3E:                   :INITIALIZE STACK

VT3F:                   :FAULT ADDRESS IS VT3A

VT3G:                   :RAND

VT3H:                   :J(RAND)=RANDOM #

VT3I:                   :SET C(00-02) TO ASCENDING

VT3J:                   :RANDOM NUMBERS. THESE

VT3K:                   :NUMBERS SHOULD BE PLACED

VT3L:                   :ON THE STACK BECAUSE

VT3M:                   :OF

VT3N:                   :OVERFLOW?

VT3O:                   :LD 3,8BREG

VT3P:                   :LD 0,0,3

VT3Q:                   :LD 1,1,3

VT3R:                   :LD 2,2,3

VT3S:                   :ADCM 0,1,SNR

VT3T:                   :ADCM 1,2,SZR

VT3U:                   :ERROR

VT3V:                   :LOOP

VT3W:                   :J(RANDTLO)

VT3X:                   :ITERATE TEST ROUTINE.....

10084 ECL23

```

01
02
03
04 05675 006230
05 05676 000100
06 05677 006023-
07 05700 000324
08 05701 005713
09 05702 105720
10 05703 000000
11 05704 006221
12 05705 005713
13 05706 006234
14 05707 040043
15 05710 105000
16 05711 061777
17 05712 106653
18 05717 000414
19 05720 005721
20 05721 034226
21 05722 021403
22 05723 030040
23 05724 174010
24 05725 156015
25 05726 106414
26
27
28
29
30
31
32
33 05733 006231
34

```

VT4:                   :INITIALIZE TEST.....

VT4A:                   :VECTOR @VT4B,VT4A

VT4B:                   :J(RAND)=RANDOM #

VT4C:                   :DISPATCH TABLE AT LOCATION

VT4D:                   :VCTAB... ENTRY 0 SET

VT4E:                   :TO ADDRESS @VT4B. OTHERS

VT4F:                   :TO ADDRESS VT4A.

VT4G:                   :INITIALIZE STACK

VT4H:                   :FAULT ADDRESS IS VT4A

VT4I:                   :RAND

VT4J:                   :J(RAND)=RANDOM #

VT4K:                   :TEST THAT LOC 43

VT4L:                   :IS PROPERLY STORED

VT4M:                   :ON THE STACK

VT4N:                   :ERROR

VT4O:                   :JMP VT4C

VT4P:                   :LD 3,8BREG

VT4Q:                   :LD 0,0,3

VT4R:                   :LD 1,1,3

VT4S:                   :LD 2,2,3

VT4T:                   :LD 4,3

VT4U:                   :SUB# 2,3,SNR

VT4V:                   :SUB# 0,1,SZR

VT4W:                   :ERROR

VT4X:                   :LOOP

VT4Y:                   :J(RANDTLO)

VT4Z:                   :ITERATE TEST ROUTINE.....

```

100% ECL23
01
02
03
04 05784 006230
05 05735 000100
06
07 05736 006023-
08 05737 000324
09 05740 005753
10 05741 105760
11 05742 000000
12
13 05743 006221
14 05744 005753
15 05745 020004
16 05746 024005
17 05747 030006
18 05750 034007
19 05751 061777
20
21
22 05757 000432
23 05760 105761
24 05761 034040
25 05762 021773
26 05763 024004
27 05764 031774
28 05765 034005
29 05766 156415
30 05767 106414
31 05770 000415
32 05771 034040
33 05772 020006
34 05773 025775
35 05774 030007
36 05775 035776
37 05776 156415
38 05777 106414
39
40 06004 101001
41
42
43 06011 006231
44

```

```

:INITIALIZE TEST.....
SETUP 100
JSR @ENTIN
100
VECTOR @VT6A,VT6R
JSP @VEEC
VP
VT6R
@VT6A
0
STACK VT6R
JSR @ISTK
VT6R
LDA 0,4
LDA 1,5
LDA 2,6
LDA 3,7
VCT 180+VCTAB
ERROR
JMP VT5F
@+1
LOA 3,8P
LOA 0,-5,3
LOA 1,4
LOA 2,-4,3
LOA 3,5
SUB# 2,3,SNR
SUB# 0,1,SZR
JMP VT5E
LOA 3,8P
LOA 0,6
LOA 1,-3,3
LOA 2,7
LOA 3,-2,3
SUB# 2,3,SNR
SUB# 0,1,SZR
MOV 0,0,SKP
ERROR
LOOP
JSR @ENTLO
:ITERATE TEST ROUTINE.....

```

```

VT5:
VT5A:
VT5B:
VT5C:
VT5D:
VT5E:
VT5F:
VT5G:
VT5H:
VT5I:
VT5J:
VT5K:
VT5L:
VT5M:
VT5N:
VT5O:
VT5P:
VT5Q:
VT5R:
VT5S:
VT5T:
VT5U:
VT5V:
VT5W:
VT5X:
VT5Y:
VT5Z:

```

```

:DISPATCH TABLE AT LOCATION
:VCTAB... ENTRY 0 SET
:TO ADDRESS @VT6A. OTHERS
:TO ADDRESS VT6R.
:INITIALIZE STACK
:FAULT ADDRESS IS VT6R
:TEST STACK OVERFLOW
:ON INTERRUPT STACK.
:FAIL TO OVERFLOW.
:ITERATE TEST ROUTINE.....

```

```

10087 ECL23
01
02
03
04 06035 006230
05 06036 000100
06
07 06037 006023-
08 06040 000324
09 06041 06054
10 06042 106054
11 06043 000000
12
13 06044 006221
14 06045 006063
15 06046 034004
16 06047 054006
17 06050 020072-
18 06051 040005
19 06052 061777
20 06054 106056
21 06055 177400
22
23
24 06062 000411
25 06063 021413
26 06064 025416
27 06065 136415
28 06066 100014
29
30
31 06073 006231
32
33
34
35
36

```

```

:INITIALIZE TEST.....
:DISPATCH TABLE AT LOCATION
:VCTAR... ENTRY 0 SET
:TO ADDRESS @VT7B. OTHERS
:TO ADDRESS VT7B.
:INITIALIZE STACK
:FAULT ADDRESS IS VT7C
:TEST FOR AC0-AC3
:STORAGE ON THE STACK
:AFTER AN OVERFLOW
:VIA THE INTERRUPT STACK.
:FAIL TO OVERFLOW.
:AC0) = AC0 VIA OV ON STACK
:AC1) = AC3 VIA OV ON STACK
:AC2) = PC VIA VCT INSTRUCTION
:CURRENT C(AC0) = 177777
:ITERATE TEST ROUTINE.....

```

```

10088 ECL23
01
02
03
04 06074 006230
05 06075 000100
06
07 06076 006023-
08 06077 000324
09 06104 006115
10 06101 106122
11 06102 000000
12
13 06103 006221
14 06104 006115
15
16 06105 006234
17 06106 105400
18 06107 131400
19 06110 004802
20 06111 000415
21 06112 054000
22 06113 061777
23
24
25 06121 000413
26 06122 106123
27 06123 165710
28 06124 014040
29 06125 167710
30 06126 106015
31 06127 132014
32
33
34
35 06134 006231
36

```

```

:INITIALIZE TEST.....
:DISPATCH TABLE AT LOCATION
:VCTAR... ENTRY 0 SET
:TO ADDRESS @JR0B. OTHERS
:TO ADDRESS JR0A.
:INITIALIZE STACK
:FAULT ADDRESS IS JR0A
:RANDOM
:RANDOM #
:TEST JMPB ABILITY TO
:RESTORE AC0-1-2.
:PLACE AC'S IN ASCENDING
:ORDER.
:OVERFLOW?
:VECTOR ENTERS HERE
:DESTROY AC1-2
:ADJUST STACK POINTER
:FOR MASK. RESTORE...
:JMPB FAILED TO
:RESTORE AC'S PROPERLY
:ITERATE TEST ROUTINE.....

```

:0009 FCL23

```

01
02
03
04 06135 006230
05 06136 000100
06
07 06137 006233-
08 06140 000324
09 06141 006156
10 06142 006163
11 06143 000000
12
13 06144 006221
14 06145 006156
15 06146 004402
16 06147 000420
17 06150 054000
18
19 06151 006234
20 06152 115400
21 06153 101040
22 06154 061177
23 106653
24
25 06162 000413
26 06163 106164
27 06164 014040
28 06165 101020
29 06166 167110
30 06167 116615
31 06170 110003
32
33
34 06175 006231
35

```

```

JBI:
SETUP 100
JSR @ENTIN
100
VECTOR @JB18,JB1A
JSR @VEEC
VP
JB1A
@JB18
0
STACK JB1A
JSR @ISTK
JB1A
JSR +2
JMP JB1C
STA 3,0
RAND
JSR @ENTRA
INC 0,3
MOV 0,0
VCT 180+VCTAR
JB1A:
JMP JB1D
@+1
DSZ SP
MOVZ 0,0
RSTR
ADCB 0,3,SNR
MOV 0,0,SNR
ERROR
LOOP
JSR @ENTLO
JB10:

```

:0090 FCL23

```

01
02
03
04 0617A 006230
05 06177 000100
06
07 06200 006233-
08 06201 000324
09 06202 006222
10 06203 106227
11 06204 000000
12
13 06205 006221
14 06206 006222
15 06207 004402
16 06210 000422
17 06211 054000
18
19 06212 006234
20 06213 105400
21 06214 131400
22 06215 040040
23 06216 044041
24 06217 050042
25 06220 061177
26 106653
27 06226 000706
28 06227 106230
29 06230 014040
30 06231 167110
31 06232 020040
32 06233 024041
33 06234 036042
34 06235 106015
35 06236 132014
36
37
38
39 06243 006231
40

```

```

JMP:
SETUP 100
JSR @ENTIN
100
VECTOR @JB28,JB2A
JSR @VEEC
VP
JB2A
@JB28
0
STACK JB2A
JSR @ISTK
JB2A
JSR +2
JMP JB2C
STA 3,0
RAND
JSR @ENTRA
INC 0,1
JMPR TO RESTORE
:C(LDC 40-41-42).
:ASCENDING RANDOM NUMBERS
:TO LOCATIONS 40-42.
:OVERFLOW
JMP JB80
@+1
DSZ SP
RSTR
LDA 0,40
LDA 1,41
LDA 2,42
ADCB 0,1,SNR
ERROR
LOOP
JSR @ENTLO
JB20:

```

```

:INITIALIZE TEST....
:DISPATCH TABLE AT LOCATION
:VCTAR... ENTRY 0 SET
:TO ADDRESS @JB18, OTHERS
:TO ADDRESS JB1A.
:INITIALIZE STACK
:FAULT ADDRESS IS JB1A
:C(AC0)=RANDOM #
:TEST THE ABILITY OF
:JMPR TO RESTORE
:C(LDC 40-41-42).
:ASCENDING RANDOM NUMBERS
:TO LOCATIONS 40-42.
:OVERFLOW
:ADJUST C(SP) FOR MASK.
:RESTORE STATE...
:TEST THE VALUES IN
:40-42 FOR ASCENDING
:ORDER.
:ITERATE TEST ROUTINE....

```

```

10092 FCL23
01 06312 020226
02 06313 040240
03
04 06314 006233-
05 06315 000324
06 06316 006344
07 06317 106351
08 06320 000000
09
10 06321 006230
11 06322 000002
12
13 06323 006231
14 06324 006344
15 06325 020240
16 06326 040004
17 06327 004402
18 06330 000426
19 06331 054000
20
21 06332 006234
22 06333 105400
23 06334 131400
24 06335 155400
25 06336 040040
26 06337 044041
27 06340 050042
28 06341 054043
29 06342 061777
30
31 06350 000430
32 06351 106352
33 06352 014000
34 06353 151710
35 06354 171710
36 06355 167710
37 06356 106015
38 06357 156014
39 06360 000414
40 06361 020040
41 06362 024041
42 06363 030042
43 06364 034043
44 06365 034043
45 06366 106015
46 06366 156014
47
48 06373 101001
49
50
51 06400 006231
52
53 06401 010240
54 06402 028227
55 06403 020022-
56 06404 106400
57 06405 020240
58 06406 106014
59 06407 000712

```

```

10091 FCL23
01
02
03
04 06244 006230
05 06245 000100
06
07 06246 006233-
08 06247 000324
09 06250 006266
10 06251 106273
11 06252 000900
12
13 06253 006221
14 06254 006266
15 06255 004402
16 06256 000421
17 06257 054000
18
19 06260 006234
20 06261 040043
21 06262 105260
22 06263 044044
23 06264 061777
24
25
26 06272 000417
27 06273 106274
28 06274 014040
29 06275 101040
30
31 06276 167710
32 06277 030043
33 06300 034044
34 06301 101002
35 06302 000403
36 06303 112415
37 06304 136414
38
39
40 06311 006231
41

```

```

:DISPATCH TABLE AT LOCATION
:VCTAB... ENTRY 0 SET
:TO ADDRESS @JB40. OTHERS
:TO ADDRESS JB4C.

:INITIALIZE TEST.....

:INITIALIZE STACK
:FAULT ADDRESS IS JB4C
:CHECK JMPB AT EACH LOC
:OP THE STACK.

:(AC0)=RANDOM #
:PLACE C(40-43) IN ASCENDING
:ORDER. ALSO PLACE AC'S
:IN THIS ORDER.

:OVERFLOW?

:VCT ENTRY...
:ADJUST STACK POINTER AND
:DESTROY C(AC0-3).

:RESTORE AC'S, LOC 40-43, ETC.
:FALL AC'S SHOULD BE IN
:ASCENDING DATA ORDER.
:ERROR ON AC RESTORE

:(LOC 40-43) SHOULD BE
:RESTORED TO ORIGINAL
:DATA.

:ITERATE TEST ROUTINE....

:ADVANCE TO NEXT BUFFER
:LOCATION AND TEST FOR
:END OF BUFFER-110. IF NOT
:END OF BUFFER GO TO JB4.

```

```

LDA 0, R0EG
STA 0, TEM
VECTOR @JB40, JB4C
JSR @VEEC
VP
JB4C
@JB40
0
SETUP 2
JSR @ENTIN
2
STACK JB4C
JSR @ISTK
JB4C
LDA 0, TEM
STA 0, ISP
JSR +2
JMP JB4E
STA 3, 0
RAND
JSR @ENTRA
INC 0, 1
INC 1, 2
INC 2, 3
STA 0, 40
STA 1, 41
STA 2, 42
STA 3, 43
VCT 180+VCTAR
ERROR
JMP JB4H
@+1
DSZ SP
DHR 4, 0
DHR 4, 2
RSTR
ADC# 0, 1, SNR
ADC# 2, 3, SZR
JMP JB4G
LDA 0, 40
LDA 1, 41
LDA 2, 42
LDA 3, 43
ADC# 0, 1, SNR
ADC# 2, 3, SZR
ERROR
MOV 0, 0, SKP
ERROR
LOOP
JSR @ENTLO
AGAIN JB4
ISZ TEM
LDA 1, REND
LOR 0, -110
SUB 0, 1
LDA 0, TEM
ADC# 0, 1, SZR
JMP JB4

```

```

10093 ECL23
01
02 06410 002401
03 06411 006600
04
05
06 06412 054310 PMUL:
07 06413 034014-
08 06414 125203
09 06415 101201
10 06416 143220
11 06417 175404
12 06420 000774
13 06421 125260
14 06422 040304
15 06423 044305
16 06424 050306
17 06425 002310
18
19 06426 054311 MDCK:
20 06427 034304
21 06430 162414
22 06431 006406
23 06432 036305
24 06433 164414
25 06434 006403
26 06435 036306
27 06436 172414
28 06437 010311 MDCK1:
29 06440 002311
30
31 06441 054312 PDIV:
32 06442 142432
33 06443 000411
34 06444 034014-
35 06445 125120
36 06446 101100 PDIV1:
37 06447 142412
38 06450 142400
39 06451 125100
40 06452 175404
41 06453 000773
42 06454 040304 PDIV2:
43 06455 044305
44 06456 050306
45 06457 002312
46
47 06460 054311 $MDCK:
48 06461 034271
49 06462 101003
50 06463 175400
51 06464 175213
52 06465 000752
53 06466 101003
54 06467 000740
55 06470 002311
56

10094 ECL23
01
02
03 06471 040304 EDIV:
04 06472 102400
05 06473 125112
06 06474 102000
07 06475 101001
08 06476 040304 PSDIV:
09 06477 054307
10 06500 050306
11 06501 044305
12 06502 155102
13 06503 150400
14 06504 176560
15 06505 175120
16 06506 101113
17 06507 004405
18 06510 175400
19 06511 124404
20 06512 100441
21 06513 100440
22 06514 142432 PSDIV1:
23 06515 000415
24 06516 054013-PSDIV2:
25 06517 034014-
26 06520 125120
27 06521 101100
28 06522 142412
29 06523 142400
30 06524 125109
31 06525 175404
32 06526 000773
33 06527 030306 PSDIV3:
34 06530 125113
35 06531 000404
36 06532 176520 PSDIV4:
37 06533 054271
38 06534 002307
39 06535 034013-
40 06536 175203
41 06537 174001
42 06540 100400
43 06541 175203
44 06542 124400
45 06543 176540
46 06544 054271
47 06545 040304
48 06546 044305
49 06547 050306
50 06550 002307
51

:SIGN EXTEND DIVIDE
:SOFTWARE UNSIGNED MULTIPLY
:RESULT TO C(AC0-1)
:CHECK PRESENT C(AC0-2)
:WITH C(OK0-2)
:RETURN +1 IF NO ERROR,
:+2 IF ERROR.
:SOFTWARE DIVIDE
:(AC0-1)/C(AC2)
:(AC0)REMAINDER
:(AC1)=QUOTIENT
:STORE RESULT IN OK
:BLOCK
:CHECK SIGNED DIVIDE.
:(CARRY) IS WRONG
:NO 0-V. CHECK RESULT.
:0-V. DON'T CHECK.

:STA 0,OK0
:SUBC 0,+0
:MOVL# 1,1,SZC
:ADC 0,0
:MOV 0,0,SKP
:STA 0,(OK)
:STA 2,OK2
:STA 1,OK1
:MOVL 2,+3,SZC
:NEG 2,2
:SUBCL 3,+3
:MOVL# 0,0,SNC
:JMP PSDIV1
:INC 3,+3
:NEG 1,1,SZR
:COMO 0,0,SKP
:NEGO 0,0
:SUBR# 2,+0,SZC
:JMP PSDIV0
:STA 3,=0
:MOVL 1,1
:MOVL 0,0
:SUBR# 2,0,SZC
:SUB 2,0
:MOVL 1,1
:INC 3,+3,SZR
:JMP PSDIV2+3
:LOA 2,OK2
:MOVL# 1,1,SNC
:JMP +4
:SUBZL 3,+3
:STA 3,PSCRY
:JMP #OK3
:LOA 3,=0
:MOVR 3,+3,SNC
:COM 3,+3,SKP
:NEG 0,0
:MOVR 3,+3,SNC
:NEG 1,1
:SUBOL 3,+3
:STA 3,PSCRY
:STA 0,OK0
:STA 1,OK1
:STA 2,OK2
:JMP #OK3

:SOFTWARE SIGNED DIVIDE.
:SAVE ORIG NUMBERS.
:CHECK DIVISOR SIGN
:FORM ABS VALUE
:SAVE SIGN IN C(AC3)
:BIT 14
:DIVIDENT POSITIVE.SAVE
:IN RIT 15
:FORM DIVIDENT ABS
:VALUE
:DIVIDE OVERFLOW...
:FLAG WORD FOR SIGNS
:UNSIGNED DIVIDE SUBROUTINE
:(CARRY)=1, DIVIDE ERROR
:GET FLAG WORD
:TEST REMAINDER SIGN
:POSITIVE
:NEGATIVE
:TEST QUOTIENT SIGN
:NEGATIVE
:ZERO C(CARRY).
:STORE RESULTS IN
:C(OK0-2)

```

10095 ECL23

```

01
02
03 06551 050307 PSMUL: STA 3,0K3
04 06552 040257 STA 0,MTEM
05 06553 102460 SUBC 0,0
06 06554 176400 SUB 3,3
07 06555 125112 MOVL# 1,1,SZC
08 06556 157000 ADD 2,3
09 06557 151112 MOVL# 2,2,SZC
10 06560 137000 ADD 1,3
11 06561 054013- STA 3,=0
12 06562 004630 JSR PHUL
13 06563 034013- LDA 3,=0
14 06564 162400 SUB 3,0
15 06565 034257 LDA 3,MTEM
16 06566 175113 MOVL# 3,3,SNC
17 06567 152401 SUB 2,2,SKP
18 06570 152000 ADC 2,2
19 06571 167022 ADD 3,1,SZC
20 06572 151400 INC 2,2
21 06573 143000 ADD 2,0
22 06574 036304 LDA 2,OK2
23 06575 040304 STA 0,OK0
24 06576 044305 STA 1,OK1
25 06577 002307 JMP 0OK3

```

:SOFTWARE SIGNED MULTIPLY

:UNSIGNED MULTIPLY

10096 FCL23

```

01
02 06600 014204 END: DSZ
03 06601 000437 JMP
04 06602 010203 ISZ
05 06603 101001 MOV
06 06604 063077 HALT
07 06605 020205 LDA
08 06606 040204 STA
09
10 06607 060477 READS 0
11 06610 101112 MOVL# 0,0,SZC
12 06611 000403 JMP *3
13 06612 122470 ELDA 0,SMREG
14
15 06614 143770 ANDI 104,0
16
17 06616 101004 MOV 0,0,SZR
18 06617 000406 JMP PCK1
19
20 06620 006220 JSR @IMSS
21 06621 001333 PASMES
22 06622 125020 MOVZ 1,1
23 06623 024203 LDA 1,PASS
24 06624 006217 JSR @IPDEC
25
26 06625 034010--PCK1: LDA 3,EGGS
27 06626 021400 LDA 0,0,3
28 06627 101005 MOV 0,0,SNR
29 06630 000410 JMP DOMOR
30 06631 015403 DSZ 3,3
31 06632 000406 JMP DOMOR
32 06633 062677 IORST
33 06634 021403 LDA 0,3,3
34 06635 035404 LDA 3,4,3
35 06636 041774 STA 0,2,3
36 06637 001400 JMP 0,3
37 06640 010407 DOMOR: ISZ CATSW
38 06641 002403 JMP 2,4,3
39 06642 002401 JMP @+1
40 06643 000562 START-1
41 06644 000563 START
42
43
44 : *** END OF TEST ROUTINES ***

```

```

:INTERNAL COUNT DONE?
:NOPE, LOOP BACK
:YEP, RUMP PASS COUNT
:PASS CNT > 65K
:RESTORE INTERNAL COUNT
:CHECK SWITCH 4
:IF SET, 00 NGI
:PRINT PASS COUNT
:PRINT PASS COUNT
:STANDARD DTOS RETURN
:AUTO MODE?
:NO, RE-00 TEST
:YES, PASS COUNT ZERO?
:NO, BACK TO TEST
:YES, BACK TO DTOS
:END OF TEST
:RESTART TEST...
:FOR START CAT THEN TEST

```



10097 ECL23

```

01 ; *****EGGS & DIRT DATA BLOCKS*****
02 ;
03 ;
04 ;
05 EGGS:
06 06645 000000 AUTO: 0 ;DTOS AUTO MODE SWITCH
07 06646 000000 DEV: 0 ;PRIMARY DEVICE CODE TO BE TESTED
08 06647 000000 CATSW: 0 ;CAT SWITCH, SET IF CAT LOADED
09 06650 000000 PCNT: 0 ;PASS COUNT, # OF TIMES TO RUN
10 06651 000000 RTRN: 0 ;RETURN POINT TO RESTORE DTOS
11 06652 000000 SWREG: 0 ;DEFAULT SWITCH REGISTER
12 06653 000100 VCTAB: .BLK 100
13
14 06753 006753 PRGEND: PRGEND
15
16 06754 047503 .TXT /COPYRIGHT(C)DGC,1974,76
17 054520
18 044522
19 044107
20 024124
21 024503
22 043504
23 026103
24 034461
25 032067
26 033454
27 06767 040466 ALL RIGHTS RESERVED/
28 046114
29 051040
30 053511
31 052110
32 020123
33 042522
34 042523
35 053122
36 042105
37 000000

```

0098 ECL23

```

01 006653
02 000006
03 000110
04 000665
05 003626
06 103571
07 163510
08 005400
09 000012
10 002520
11 000005
12 002467
13 002435
14 002411
15 002357
16 002324
17 002300
18 002253
19 000327
20 022002
21 117000
22 036402
23 173000
24 032402
25 147000
26 026402
27 123000
28 000775
29 125252
30 000340
31 177770
32 000212
33 104400
34 022000
35 000060
36 000011
37 033031
38 000144
39 000100
40 177740
41 000010
42 000140
43 000377
44 000200
45 000400
46 001777

```

07002 141705 DIRTY -TXTE IECLIPSE2321

DTOS8 175772

DTOSB

```

54
55
56 00013-000000
57 177760
58 005000
59 006653
60 006653

```

0099 FCL23

| AC0   | 000245 | 13/37 | 20/08 | 20/26 | 21/21 | 21/34 | 22/05 | 22/16 |
|-------|--------|-------|-------|-------|-------|-------|-------|-------|
| AC1   | 000246 | 22/41 | 23/35 | 21/35 | 22/28 | 22/43 | 23/23 |       |
| AC2   | 000247 | 13/38 | 21/22 | 21/36 | 22/29 | 22/45 | 23/24 |       |
| AC3   | 000250 | 13/40 | 22/47 | 23/36 | 36/08 | 37/22 | 37/43 | 38/20 |
|       |        | 38/41 | 39/23 | 39/45 | 40/15 | 40/41 | 41/19 | 41/27 |
|       |        | 41/51 | 42/18 | 43/26 | 43/50 | 44/14 | 44/16 | 44/40 |
|       |        | 44/44 | 45/12 | 45/17 | 45/20 | 45/38 | 45/41 | 46/17 |
|       |        | 46/20 | 46/35 | 46/37 | 47/15 | 47/35 | 47/38 | 48/19 |
|       |        | 48/22 | 48/38 | 48/40 | 49/26 | 50/15 | 50/35 | 50/39 |
|       |        | 51/19 | 51/23 | 52/16 | 52/23 | 52/40 | 52/47 | 53/16 |
|       |        | 53/25 | 53/40 | 53/44 | 54/26 | 54/28 | 55/18 | 56/18 |
|       |        | 56/25 | 56/45 | 56/51 | 57/16 | 57/26 | 58/17 | 58/22 |
|       |        | 58/38 | 58/42 | 59/16 | 59/18 | 60/13 | 60/18 | 60/34 |
|       |        | 60/37 | 61/30 | 62/25 | 63/18 | 63/20 | 63/44 | 63/49 |
|       |        | 64/31 | 64/36 | 64/49 | 64/54 | 65/21 | 65/26 | 65/46 |
|       |        | 65/50 | 66/19 | 66/23 | 66/43 | 67/22 | 67/22 | 67/27 |
|       |        | 68/19 | 68/26 | 68/26 | 69/35 | 70/19 | 70/25 | 71/26 |
|       |        | 71/37 | 72/21 | 72/26 | 73/24 | 73/32 | 74/22 | 74/27 |
|       |        | 74/52 | 74/57 | 75/40 | 75/45 | 75/50 | 76/17 | 76/23 |
|       |        | 76/41 | 76/47 | 77/23 | 78/31 | 78/35 | 79/33 | 79/38 |
|       |        | 80/19 | 80/42 | 81/21 | 81/26 | 81/48 | 81/53 | 82/21 |
|       |        | 82/27 | 83/25 | 83/34 | 84/22 | 84/32 | 85/22 | 85/40 |
|       |        | 85/42 | 86/21 | 87/24 | 87/30 | 88/26 | 88/34 | 89/25 |
|       |        | 89/33 | 90/28 | 90/38 | 91/27 | 91/39 | 92/32 | 92/48 |
|       |        | 92/50 |       |       |       |       |       |       |
| AGAIN | 000435 | 10/20 | 61/32 | 62/27 | 67/29 | 69/37 | 71/39 | 73/34 |
|       |        | 75/32 | 77/25 | 78/34 | 79/40 | 82/52 |       |       |

AGAIN 000435 MC

AUTO 006645  
80 000241  
817 000242  
R20 000244  
R40 000240  
BAC1 000321  
BAC2 000322  
BAC3 000323

MC

BAMER 000160  
BAMH1 001404  
BAMH2 001436  
BAMH3 001447  
BAMH4 001457  
BAMH5 001472  
BAMH7 001532  
BAMH8 001531  
BAMRE 000286  
BAMSI 000313  
BAMSU 000265  
BAMT0 000315  
BAMT1 000314  
BAMXR 000270  
BAMX2 000267  
BBEG 000226

0100 FCL23

| REGG  | 000576 | 17/25 | 17/53 | 17/54 | 17/59 | 17/59 | 17/59 | 18/07 | 35/02 | 18/37 | 43/04 | 44/48 |
|-------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| BE63  | 000605 | 13/17 | 13/17 | 13/19 | 13/19 | 13/19 | 13/19 | 14/06 | 49/30 | 54/06 | 61/34 | 62/29 |
| BEGIN | 000214 | 67/31 | 67/31 | 69/39 | 71/41 | 73/36 | 75/05 | 79/42 | 79/09 | 79/42 | 92/54 | 77/27 |
| BEVD  | 000227 | 78/05 | 78/41 | 79/09 | 79/09 | 79/42 | 92/54 |       |       |       |       |       |
|       |        | 10/10 | 61/02 | 62/02 | 78/02 |       |       |       |       |       |       |       |
|       |        | 12/45 | 12/47 |       |       |       |       |       |       |       |       |       |
|       |        | 32/54 |       |       |       |       |       |       |       |       |       |       |
|       |        | 13/34 |       |       |       |       |       |       |       |       |       |       |
|       |        | 13/50 |       |       |       |       |       |       |       |       |       |       |
|       |        | 13/51 |       |       |       |       |       |       |       |       |       |       |
|       |        | 13/52 |       |       |       |       |       |       |       |       |       |       |
|       |        | 13/53 |       |       |       |       |       |       |       |       |       |       |
|       |        | 7/19  |       |       |       |       |       |       |       |       |       |       |
|       |        | 29/15 |       |       |       |       |       |       |       |       |       |       |
|       |        | 29/29 |       |       |       |       |       |       |       |       |       |       |
|       |        | 12/31 |       |       |       |       |       |       |       |       |       |       |
|       |        | 35/07 |       |       |       |       |       |       |       |       |       |       |
|       |        | 37/04 |       |       |       |       |       |       |       |       |       |       |
|       |        | 37/25 |       |       |       |       |       |       |       |       |       |       |
|       |        | 38/02 |       |       |       |       |       |       |       |       |       |       |
|       |        | 38/23 |       |       |       |       |       |       |       |       |       |       |
|       |        | 39/02 |       |       |       |       |       |       |       |       |       |       |
|       |        | 39/08 |       |       |       |       |       |       |       |       |       |       |
|       |        | 39/18 |       |       |       |       |       |       |       |       |       |       |
|       |        | 39/26 |       |       |       |       |       |       |       |       |       |       |
|       |        | 39/34 |       |       |       |       |       |       |       |       |       |       |
|       |        | 40/01 |       |       |       |       |       |       |       |       |       |       |
|       |        | 40/09 |       |       |       |       |       |       |       |       |       |       |
|       |        | 40/16 |       |       |       |       |       |       |       |       |       |       |
|       |        | 40/26 |       |       |       |       |       |       |       |       |       |       |
|       |        | 41/02 |       |       |       |       |       |       |       |       |       |       |
|       |        | 41/10 |       |       |       |       |       |       |       |       |       |       |
|       |        | 41/19 |       |       |       |       |       |       |       |       |       |       |
|       |        | 41/38 |       |       |       |       |       |       |       |       |       |       |
|       |        | 41/46 |       |       |       |       |       |       |       |       |       |       |
|       |        | 42/15 |       |       |       |       |       |       |       |       |       |       |
|       |        | 43/02 |       |       |       |       |       |       |       |       |       |       |
|       |        | 43/08 |       |       |       |       |       |       |       |       |       |       |
|       |        | 43/16 |       |       |       |       |       |       |       |       |       |       |
|       |        | 43/23 |       |       |       |       |       |       |       |       |       |       |
|       |        | 43/34 |       |       |       |       |       |       |       |       |       |       |
|       |        | 43/42 |       |       |       |       |       |       |       |       |       |       |
|       |        | 44/03 |       |       |       |       |       |       |       |       |       |       |
|       |        | 44/08 |       |       |       |       |       |       |       |       |       |       |
|       |        | 44/10 |       |       |       |       |       |       |       |       |       |       |
|       |        | 44/16 |       |       |       |       |       |       |       |       |       |       |
|       |        | 44/19 |       |       |       |       |       |       |       |       |       |       |
|       |        | 44/38 |       |       |       |       |       |       |       |       |       |       |
|       |        | 44/41 |       |       |       |       |       |       |       |       |       |       |
|       |        | 44/49 |       |       |       |       |       |       |       |       |       |       |
|       |        | 44/50 |       |       |       |       |       |       |       |       |       |       |
|       |        | 44/59 |       |       |       |       |       |       |       |       |       |       |
|       |        | 44/40 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/02 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/08 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/13 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/16 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/17 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/18 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/12 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/15 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/23 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/28 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/37 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/41 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/13 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/16 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/18 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/12 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/15 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/20 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/28 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/37 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/41 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/13 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/16 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/18 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/12 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/15 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/20 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/28 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/37 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/41 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/13 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/16 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/18 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/12 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/15 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/20 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/28 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/37 |       |       |       |       |       |       |       |       |       |       |
|       |        | 45/41 |       |       |       |       |       |       |       |       |       |       |

BEG1 000563

0101 FCL23

|              |       |       |       |
|--------------|-------|-------|-------|
| C640R 002663 | 45/30 | 45/38 |       |
| C641 002672  | 46/02 | 46/16 |       |
| C641A 002705 | 46/07 | 46/17 |       |
| C641B 002711 | 46/09 | 46/23 |       |
| C642A 002720 | 46/28 | 46/34 |       |
| C642B 002731 | 46/30 | 46/35 |       |
| C643 002743  | 47/02 | 47/14 |       |
| C643A 002755 | 47/11 | 47/15 |       |
| C643B 002761 | 47/07 | 47/15 |       |
| C644 002762  | 47/18 |       | 48/11 |
| C644A 002777 | 47/27 | 47/34 |       |
| C644B 003003 | 47/23 | 47/35 |       |
| C645 003012  | 48/02 |       |       |
| C645A 003027 | 48/18 |       |       |
| C645B 003033 | 48/07 | 48/19 |       |
| C646 003042  | 48/25 |       |       |
| C646A 003060 | 48/30 | 48/38 |       |
| C647 003066  | 49/01 |       | 49/32 |
| C647A 003100 | 49/11 | 49/16 |       |
| C647B 003113 | 49/24 |       |       |
| C648 003126  | 50/02 |       |       |
| C648A 003145 | 50/07 | 50/16 |       |
| C648B 003150 | 50/15 | 50/19 |       |
| C649 003151  | 50/22 |       |       |
| C649A 003161 | 50/27 | 50/32 |       |
| C649B 003166 | 50/29 | 50/34 |       |
| C649C 003176 | 50/33 | 50/39 |       |
| C650 003201  | 51/04 | 51/31 |       |
| C650A 003215 | 51/09 | 51/18 |       |
| C650B 003222 | 51/15 | 51/20 |       |
| C650C 003230 | 51/19 | 51/23 |       |
| C651 003240  | 52/02 |       |       |
| C651A 003252 | 52/07 | 52/15 |       |
| C651B 003257 | 52/09 | 52/17 |       |
| C651C 003270 | 52/16 | 52/23 |       |
| C652 003271  | 52/26 |       | 53/39 |
| C652A 003303 | 52/31 | 52/39 |       |
| C652B 003310 | 52/33 | 52/41 |       |
| C652C 003321 | 52/40 | 52/47 |       |
| C653 003322  | 53/02 | 53/16 |       |
| C653A 003334 | 53/07 | 53/15 |       |
| C653B 003341 | 53/09 | 53/17 |       |
| C654 003353  | 53/26 |       |       |
| C654A 003366 | 53/31 | 53/36 |       |
| C654B 003401 | 53/40 | 53/44 |       |
| C654X 003373 | 53/33 | 53/41 |       |
| C655 003402  | 54/01 | 54/16 |       |
| C655A 003414 | 54/11 | 54/34 |       |
| C655B 003425 | 54/18 | 54/22 |       |
| C655C 003435 | 54/27 |       |       |
| C655D 003441 | 54/26 | 54/28 |       |
| C656 003447  | 55/02 |       |       |
| C656A 003451 | 55/04 | 55/24 |       |
| C656B 003470 | 55/09 | 55/18 |       |
| C657 003476  | 56/02 |       |       |
| C657A 003512 | 56/09 | 56/17 |       |
| C657B 003517 | 56/07 | 56/19 |       |

0102 FCL23

|              |       |       |       |
|--------------|-------|-------|-------|
| C657C 003530 | 56/18 | 56/25 |       |
| C65A 003531  | 56/28 |       |       |
| C65A 003546  | 56/35 | 56/44 |       |
| C65B 003553  | 56/45 | 56/51 |       |
| C65C 003563  | 57/02 | 57/22 |       |
| C65D 003571  | 57/09 | 57/15 | 57/21 |
| C65E 003577  | 57/07 | 57/18 |       |
| C65F 003605  | 57/16 | 57/26 |       |
| C65G 003620  | 58/02 |       |       |
| C65H 003626  | 58/09 | 58/18 |       |
| C65I 003635  | 58/09 | 58/16 |       |
| C65J 003642  | 58/07 | 58/18 |       |
| C65K 003651  | 58/17 | 58/22 |       |
| C65L 003652  | 58/25 |       |       |
| C65M 003664  | 58/30 | 58/37 |       |
| C65N 003671  | 58/34 | 58/39 |       |
| C65O 003677  | 58/38 | 58/42 |       |
| C65P 003684  | 58/42 | 58/46 |       |
| C65Q 003691  | 25/42 | 25/57 | 27/03 |
| C65R 003697  | 25/42 | 25/47 | 26/01 |
| C65S 003704  | 27/30 | 27/39 |       |
| C65T 003711  | 27/33 | 27/46 |       |
| C65U 003718  | 27/40 | 27/57 |       |
| C65V 003725  | 27/42 | 27/57 |       |
| C65W 003732  | 13/44 | 27/04 |       |
| C65X 003739  | 13/44 | 27/30 |       |
| C65Y 003746  | 13/45 | 27/50 | 27/57 |
| C65Z 003753  | 27/05 | 27/52 | 28/03 |
| C65AA 003760 | 27/05 | 27/52 | 28/03 |
| C65AB 003767 | 13/41 | 22/07 | 22/18 |
| C65AC 003774 | 97/06 |       |       |
| C65AD 003781 | 12/05 | 97/39 |       |
| C65AE 003788 | 31/60 | 96/31 | 96/37 |
| C65AF 003795 | 96/03 | 96/29 |       |
| C65AG 003802 | 12/45 | 97/46 |       |
| C65AH 003809 | 94/03 |       |       |
| C65AI 003816 | 30/56 |       |       |
| C65AJ 003823 | 12/16 | 12/22 |       |
| C65AK 003830 | 93/03 | 96/02 |       |
| C65AL 003837 | 13/24 |       |       |
| C65AM 003844 | 13/23 | 37/22 | 38/20 |
| C65AN 003851 | 39/45 | 41/19 | 41/51 |
| C65AO 003858 | 43/26 | 44/14 | 44/18 |
| C65AP 003865 | 45/17 | 44/16 | 44/20 |
| C65AQ 003872 | 46/37 | 45/38 | 44/24 |
| C65AR 003879 | 48/40 | 46/20 | 45/12 |
| C65AS 003886 | 49/26 | 47/38 | 46/35 |
| C65AT 003893 | 50/15 | 48/19 | 48/38 |
| C65AU 003900 | 52/16 | 50/39 | 51/19 |
| C65AV 003907 | 52/23 | 52/47 | 51/23 |
| C65AW 003914 | 53/44 | 53/16 | 53/40 |
| C65AX 003921 | 54/26 | 55/18 | 53/45 |
| C65AY 003928 | 56/51 | 56/18 | 56/45 |
| C65AZ 003935 | 57/16 | 57/26 | 58/42 |
| C65BA 003942 | 59/16 | 59/18 | 58/42 |
| C65BB 003949 | 63/20 | 60/37 | 60/37 |
| C65BC 003956 | 62/25 | 60/13 | 61/30 |
| C65BD 003963 | 64/49 | 63/44 | 64/26 |
| C65BE 003970 | 66/23 | 65/26 | 65/46 |
| C65BF 003977 | 66/43 | 67/22 | 66/19 |
| C65BG 003984 | 70/19 | 70/25 | 68/26 |
| C65BH 003991 | 69/35 | 71/26 | 71/37 |
| C65BI 003998 | 73/29 | 74/22 | 72/21 |
| C65BJ 004005 | 72/26 | 74/27 | 74/52 |
| C65BK 004012 | 75/40 | 76/23 | 76/47 |
| C65BL 004019 | 77/23 | 79/33 | 78/47 |
| C65BM 004026 | 81/21 | 81/48 | 80/19 |
| C65BN 004033 | 83/34 | 81/53 | 82/27 |
| C65BO 004040 | 84/22 | 85/22 | 82/27 |
| C65BP 004047 | 88/26 | 89/33 | 86/21 |
| C65BQ 004054 | 87/30 | 88/26 | 86/21 |
| C65BR 004061 |       | 89/33 | 90/28 |







0109 FCL23

0110 FCL25

|              |       |       |       |       |              |       |       |       |  |  |  |  |  |  |  |  |
|--------------|-------|-------|-------|-------|--------------|-------|-------|-------|--|--|--|--|--|--|--|--|
| VC1C 003747  | 60/14 | 60/17 | 60/16 | 60/18 | VC35A 005321 | 78/13 | 78/46 |       |  |  |  |  |  |  |  |  |
| VC1D 003753  | 60/13 | 60/16 | 60/18 |       | VC35B 005331 | 78/23 |       |       |  |  |  |  |  |  |  |  |
| VC2 003754   | 50/21 |       |       |       | VC35C 005340 | 78/18 | 78/30 |       |  |  |  |  |  |  |  |  |
| VC20A 004520 | 69/10 |       |       |       | VC35D 005344 | 78/23 | 78/31 |       |  |  |  |  |  |  |  |  |
| VC20B 004542 | 69/07 |       |       | 69/25 | VC35E 005355 | 78/57 |       |       |  |  |  |  |  |  |  |  |
| VC20C 004507 | 69/08 |       |       |       | VC37 005400  | 79/14 | 79/47 |       |  |  |  |  |  |  |  |  |
| VC20D 004542 | 69/26 |       |       |       | VC37A 005414 | 79/29 |       |       |  |  |  |  |  |  |  |  |
| VC21 004572  | 70/02 |       |       |       | VC37B 005417 | 79/20 | 79/25 | 79/32 |  |  |  |  |  |  |  |  |
| VC21A 004606 | 70/08 |       |       | 70/16 | VC37C 005424 | 79/21 | 79/27 | 79/34 |  |  |  |  |  |  |  |  |
| VC21B 004613 | 70/09 |       |       |       | VC37D 005425 | 79/21 | 79/27 |       |  |  |  |  |  |  |  |  |
| VC21C 004623 | 70/19 |       |       |       | VC37E 005433 | 79/29 | 79/35 |       |  |  |  |  |  |  |  |  |
| VC22 004624  | 71/02 |       |       |       | VC38 005443  | 79/53 | 79/58 |       |  |  |  |  |  |  |  |  |
| VC22A 004633 | 71/10 |       |       |       | VC38A 005445 | 80/02 | 80/20 |       |  |  |  |  |  |  |  |  |
| VC22B 004647 | 71/07 |       |       | 71/25 | VC38B 005446 | 80/09 | 80/18 | 80/20 |  |  |  |  |  |  |  |  |
| VC22C 004654 | 71/08 |       |       |       | VC38C 005466 | 80/08 | 80/18 | 80/20 |  |  |  |  |  |  |  |  |
| VC22D 004671 | 71/26 |       |       |       | VC39 005467  | 80/13 | 80/19 | 80/21 |  |  |  |  |  |  |  |  |
| VC23 004701  | 72/01 |       |       |       | VC39A 005504 | 80/24 | 80/31 | 80/40 |  |  |  |  |  |  |  |  |
| VC23A 004717 | 72/07 |       |       | 72/20 | VC39B 005511 | 80/35 | 80/42 |       |  |  |  |  |  |  |  |  |
| VC23B 004724 | 72/08 |       |       |       | VCA 004001   | 81/02 |       |       |  |  |  |  |  |  |  |  |
| VC23C 004733 | 72/21 |       |       | 72/26 | VC6A 004018  | 61/14 | 61/39 |       |  |  |  |  |  |  |  |  |
| VC24 004734  | 73/10 |       |       |       | VC6C 004030  | 61/18 | 61/24 | 61/27 |  |  |  |  |  |  |  |  |
| VC24A 004743 | 73/10 |       |       | 73/41 | VC9 004046   | 62/02 |       |       |  |  |  |  |  |  |  |  |
| VC24B 004756 | 73/07 |       |       | 73/25 | VC9A 004057  | 62/12 |       |       |  |  |  |  |  |  |  |  |
| VC24C 004763 | 73/08 |       |       |       | VC9C 004072  | 62/16 |       |       |  |  |  |  |  |  |  |  |
| VC24D 004773 | 73/26 |       |       | 73/32 | VC90 004076  | 62/18 |       |       |  |  |  |  |  |  |  |  |
| VC25 005003  | 74/02 |       |       | 74/21 | VCTAR 006653 | 62/18 |       |       |  |  |  |  |  |  |  |  |
| VC25A 005021 | 74/08 |       |       |       |              | 62/21 |       |       |  |  |  |  |  |  |  |  |
| VC25B 005026 | 74/09 |       |       | 74/25 |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC25C 005035 | 74/22 |       |       |       |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC26 005036  | 74/30 |       |       | 74/51 |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC26A 005057 | 74/33 |       |       |       |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC26B 005064 | 74/34 |       |       | 74/53 |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC26C 005073 | 74/52 |       |       |       |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC27 005074  | 75/02 |       |       | 75/25 |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC27A 005134 | 75/14 |       |       |       |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC27B 005112 | 75/17 |       |       | 75/39 |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC27C 005123 | 75/29 |       |       |       |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC27D 005141 | 75/29 |       |       |       |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC27E 005154 | 75/07 |       |       | 75/50 |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC27F 005150 | 75/40 |       |       |       |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC28 005166  | 60/27 |       |       |       |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC2B 003773  | 60/28 |       |       |       |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC2C 004000  | 60/34 |       |       | 60/37 |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC3E 005170  | 76/02 |       |       | 76/16 |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC3E 005203  | 76/08 |       |       |       |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC3E 005210  | 76/09 |       |       |       |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC3E 005220  | 76/17 |       |       | 76/23 |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC33 005221  | 76/26 |       |       |       |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC33A 005234 | 76/32 |       |       |       |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC33B 005241 | 76/33 |       |       | 76/42 |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC33C 005251 | 76/41 |       |       | 76/47 |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC34 005252  | 77/02 |       |       |       |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC34A 005255 | 77/05 |       |       | 77/32 |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC34B 005265 | 77/15 |       |       |       |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC34C 005271 | 77/15 |       |       | 77/20 |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC34D 005273 | 77/10 |       |       | 77/22 |              | 62/25 |       |       |  |  |  |  |  |  |  |  |
| VC35 005307  | 78/02 |       |       |       |              | 62/25 |       |       |  |  |  |  |  |  |  |  |

0111 FCL23

|               |       |       |       |
|---------------|-------|-------|-------|
| VT1C 005576   | 81/48 | 81/53 |       |
| VT2 005577    | 82/03 | 82/10 | 82/20 |
| VT2A 005615   | 82/09 | 82/14 |       |
| VT2B 005622   | 82/10 | 82/22 |       |
| VT2C 005632   | 82/21 | 82/27 |       |
| VT3 005633    | 83/03 | 83/14 | 83/24 |
| VT3A 005654   | 83/09 | 83/11 |       |
| VT3B 005661   | 83/10 | 83/26 |       |
| VT3C 005674   | 83/25 | 83/34 |       |
| VT4 005675    | 84/03 | 84/14 | 84/21 |
| VT4A 005713   | 84/09 | 84/10 | 84/23 |
| VT4B 005720   | 84/10 | 84/22 | 84/32 |
| VT4C 005733   | 84/22 | 84/32 |       |
| VT5 005734    | 85/03 | 85/09 | 85/21 |
| VT5A 005753   | 85/09 | 85/14 |       |
| VT5B 005760   | 85/10 | 85/23 |       |
| VT5C 005770   | 85/31 |       |       |
| VT5D 006000   | 85/39 |       |       |
| VT5E 006005   | 85/31 | 85/41 |       |
| VT5F 006011   | 85/22 | 85/42 |       |
| VT6 006012    | 86/03 |       |       |
| VT6A 006027   | 86/10 | 86/19 |       |
| VT6B 006034   | 86/09 | 86/14 | 86/21 |
| VT7 006035    | 87/03 |       |       |
| VT7A 006052   | 87/19 |       |       |
| VT7B 006054   | 87/09 | 87/10 | 87/21 |
| VT7C 006063   | 87/14 | 87/25 |       |
| VT7D 006073   | 87/24 | 87/30 |       |
| ZERGA 000151  | MC    |       |       |
|               | 87/13 |       |       |
|               | 13/26 | 24/05 |       |
| .BRAN 001113  | 12/22 | 23/31 | 96/26 |
| .EGGS 000010- | 8/34  | 37/07 | 37/28 |
| .MXCT 000174  | MC    |       |       |
|               | 13/25 | 24/06 | 24/07 |
| .RAND 001115  | 13/14 |       |       |
| .RANS 001133  |       |       |       |

38/05 38/26







LISTING

096-000223-01

PROGRAM

EXERCISER FOR ECLIPSE  
PART 7

TAPE

095-000244-01

ABSTRACT

'ECLIPSE30' IS AN EXERCISER PROGRAM USED TO TEST THE RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF THE ECLIPSE COMPUTER. 'ECLIPSE30' EXERCISES THE DOUBLE WORD INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES OF ITS RELIABLE OPERATION.



0001 ECL30 MACRO REV 03.00 15:53:19 08/06/76  
01  
02  
03  
04  
05  
06  
07  
\*\*\*\*\*  
08 ; NAME: ECLIPSE30.SR PART NUMBER: 098-000643  
09 ;  
10 ;  
11 ;  
12 ; DESCRIPTION: ECLIPSE EXERCISER, PART 7  
13 ;  
14 ;  
15 ; REVISION HISTORY:  
16 ;  
17 ; REV. DATE  
18 ; 00 12/20/74  
19 ; 01 08/06/76  
20 ;  
21 ;  
22 ; COPYRIGHT (C) DATA GENERAL CORPORATION, 1974, 1976  
23 ; ALL RIGHTS RESERVED.  
24 ;  
25 ;\*\*\*\*\*

\*\*\*\*\*  
; TITLE ECL30  
; ECLIPSE30  
; ECLIPSE30 - CONTINUATION OF ECLIPSE23  
; PART 7 OF EXERCISER FOR ECLIPSE  
; \*\*\*\*\*

\*\*\*\*\*  
; NAME: ECLIPSE30.SR PART NUMBER: 098-000643  
;  
; DESCRIPTION: ECLIPSE EXERCISER, PART 7  
;  
; REVISION HISTORY:  
; REV. DATE  
; 00 12/20/74  
; 01 08/06/76  
;  
; COPYRIGHT (C) DATA GENERAL CORPORATION, 1974, 1976  
; ALL RIGHTS RESERVED.  
; \*\*\*\*\*

10003 ECL30

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47

EXERCISER FOR ECLIPSE: PART 7

PROGRAM NAME

ECLIPSE30  
GENERAL DESCRIPTION

ECLIPSE30' IS AN EXERCISER PROGRAM USED TO TEST THE RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF THE ECLIPSE COMPUTER. 'ECLIPSE30' EXERCISES THE DOUBLE WORD INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES OF ITS RELIABLE OPERATION.

THE INSTRUCTIONS EXERCISED ARE AS FOLLOWS:

ADDI,IORI,XORI,ANDI,HLV,HSP,SNB,CLM AND PSHJ

LOCATIONS 200 TO 205 IN PAGE 0 ARE FIXED FOR ECLIPSE30 PROGRAM.

LOCATION 203 KEEPS TRACK OF NUMBER OF PASSES RUN THROUGH ECLIPSE30 PROGRAM.

LOCATION 201 KEEPS TRACK OF THE TEST RUNNING AT PRESENT AND IS USEFUL FOR DEBUG WHEN LOOPING OCCURS IN THE PROGRAM.

LOCATION 202 CONTAINS THE STARTING ADDRESS OF ECLIPSE30 PROGRAM.

LOCATION 200 IS USED BY DTOS.

LOCATION 204 KEEPS TRACK OF INTERNAL PASS COUNT WHICH IS FIXED BY LOCATION 205.

FIRST PASS THROUGH ECLIPSE30 TEST WILL RUN SUPERFAST. NEXT PASSES WILL RUN SLOWER AS EACH TEST IS RUN SEVERAL TIMES TO RUN ALL RANDOM NUMBER COMBINATIONS.

MACHINE REQUIREMENTS

ECLIPSE PROCESSOR

4K READ-WRITE MEMORY

CONSOLE EQUIPMENT

10004 ECL30

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44

SWITCH SETTINGS

THIS PROGRAM USES DATA SWITCHES AS FOLLOWS

SW\*0" - USE CONTENTS OF "SWREG" IF 0

USE DATA SWITCHES IF 1

SW\*1" - LOOP ON FAILING TEST IF 0

PROCEED TO NEXT TEST IF 1

SW\*2" - OUTPUT TO TTY IF 0

INHIBIT PRINTING TO TTY IF 1

SW\*3" - DO NOT PRINT % ERRORS IF 0

PRINT FAILURE RATE IF 1

SW\*4" - PRINT PASS COUNT IF 0

INHIBIT PRINTING PASS COUNT IF 1

SW\*5" - INHIBIT OUTPUT TO LINE PRINTER IF 0

OUTPUT TO LINE PRINTER IF 1

EDTOS PRESUMES THAT ALL SWITCHES ARE 0. RETURN IS MADE TO THE OPER SYS UPON AN ERROR AFTER FIRST PRINTING THE FAILURE REPORT. IN STAND ALONE MODE, THE OPERATOR IS ASKED TO PRESET THESE SWITCHES (THE MACHINE HALTS FOR HIM TO DO SO) AND PRESS CONTINUE. THIS PRESET VALUE CAN BE CHANGED AT ANY TIME BY SETTING DATA SWITCH #0 TO A ONE. THE PROGRAM WILL THEN FOLLOW ACTUAL SETTINGS.

STAND ALONE STARTING ADDRESS = 200

IF 'CAT' OR 'KITTEN' WAS LOADED FROM DTOS AND RESTRI WAS NEEDED, THEN USE AS FOLLOWS:

STARTING ADDR = 170 (FOR START WITH NO 'CAT')

STARTING ADDR = 171 (FOR START WITH 'CAT')

MONITOR LOCATION 203 TO CHECK THE CURRENT PASS COUNT

MONITOR LOCATION X6000 TO MAKE SURE THAT 'CAT' OR 'KITTEN' IS RUNNING. IN CASES WHERE PROGRAM IS STARTED WITH 'CAT' OR 'KITTEN' LOCATION X6000 WILL SHOW A PATTERN CHANGING FROM ZEROS TO ALL ONES TO AN INC/SNAP PATTERN.

(K= THE NUMBER OF THE HIGHEST MEMORY MODULE IN THE SYSTEM AND MAY BE A VALUE 0 - 7)



```

10007 ECL30
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40

? MACROS USED IN THIS TEST

.MACRO ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUB 3,3
*
.MACRO SUBAL
SBI 1,0
SBI 1,1
SBI 1,2
SBI 1,3
*
.MACRO ADDAL
ADD 1,0
ADD 1,1
ADD 1,2
ADD 1,3
*
.MACRO ASSEND
INC 0,1
INC 1,2
INC 2,3
*
.MACRO ACD
ZEROAC
ADC 2,2
*1 -1,2
ADD# 2+1&3, 2+2&3, SNR
MOV# 2+3&3, 2+3&3, SZR
ERROR
*

? CLEAR ALL AC'S
? AND C(CARRY)

? SUBTRACT FROM ALL AC'S

? ADD TO ALL AC'S

?SET C(AC1) TO 1 GREATER
?THEN C(AC0), ETC.

?THE VALUE IN AC*2
?SHOULD NOT EFFECT
?OTHER AC'S. CHECK
?AC DESTINATION FOR
?THE *1 INSTRUCTION.

```

```

10008 ECL30
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21

```

```

.MACRO XORIACD
ZEROAC
ADC 1,1
XOR 0,1
ADD# 1+1&3, 1+2&3, SNR
MOV# 1+3&3, 1+3&3, SZR
ERROR
*
.MACRO ANDIACD
ZEROAC
ADC 1,1
STA 1,2
AND 0,1
ADD# 1+1&3, 1+2&3, SNR
MOV# 1+3&3, 1+3&3, SZR
ERROR
*

```

```

?TEST THAT XOR RESULT
?GOES ONLY TO C(ACD)

```



```

0010 ECL30
01 ; *****LOCAL ZREL*****
02 ; *****
03 00216 005747 MINLOC: PRGEND ;END OF PROGRAM
04 00217 000000 MAXLOC: 0 ;RELOCATION RANGE OF PROGRAM
05 ;
06 ; POINTERS TO NON-RELOCATING SUBROUTINES
07
08 00220 000767 ICAL: CAL ;RELOCATED CALL SUBROUTINE
09 00221 001127 IMESS: MESS ;MESSAGE PRINT SUBROUTINE
10 00222 001001 ISIZE: SIZE ;MEMORY SIZING SUBROUTINE
11 00223 001066 IPOCT: POCY ;PRINT OCTAL SUBROUTINE
12 00224 001054 IPODEC: PDEC ;PRINT DECIMAL SUBROUTINE
13
14 ; TEMPORARYS FOR TESTS AND SUBROUTINES
15
16 00225 000000 AC0: 0 ;TEMPORARY STORAGE FOR
17 00226 000000 AC1: 0 ;MACHINE STATE IN "LOOP"
18 00227 000000 AC2: 0 ;AND "ERROR"
19 00230 000000 AC3: 0 ;SUBROUTINES.
20
21 00231 000000 CRY: 0 ;
22 00232 000000 CHARET: 0 ;RETURN SAVE CHAR ROUT.
23 00233 000000 CHORZ: 0 ;CHARACTER LINE COUNT
24 00234 000000 PDERET: 0 ;RETURN SAVE FR PRINT ROUT.
25 00235 000000 MESRET: 0 ;RETURN FR MESSAGE ROUT.
26 00236 123456 RAN: 123456 ;RANDOM NUMBER
27 00237 000000 CAL0: 0 ;TEMPORARYS FR CALL ROUT.
28 00240 000000 CAL1: 0 ;
29 00241 000000 CAL2: 0 ;
30 00242 000283 ABUF: -ABUF ;
31 00243 000010 ABUF: -BLK 10 ;
32 00254 000000 SVPC: 0 ;
33
34 ; ROUTINE TO GET PC ON STACK FAULT
35
36 00255 103210 STFL: POP 0,0 ;GET TRAP PC
37 00256 040254 STA 0,SVPC
38 00257 100010 ADI 3,0
39 00260 103110 PSH 0,0
40 00261 107710 POPB
41
42

```

```

1009 ECL30
01 ; ***** DIAGNOSTIC PROGRAM PREAMBLE *****
02 ; *****
03
04 .LOC 0 ;POINTER TO DIRTY BLOCK
05 00000 005776
06
07 .LOC 40 ;STACK CONTROL LOCATIONS
08
09 00040 000000 SP: 0
10 00041 000000 SF: 0
11 00042 000000 USL: 0
12 00043 000255 USF: STFL
13 00044 000000
14
15 .LOC 45 ;POINTER TO EGGS BLOCK
16 00045 005741 EGGS
17
18 .ZREL ;8 LOCATIONS RESERVED
19 00000-000010 ;FOR DEBUG BREAKPOINTS
20
21 00010-005741 .EGGS: EGGS ;PERMANENT POINTER TO EGGS
22 00011-000000 MEMTOP: 0 ;TOP OF MEMORY FROM SIZE
23 00012-000000 ICAT: 0 ;START LOCATION OF CAT
24
25 ; SPACE RESERVED FOR CAT/NO CAT RESTART ENTRIES
26
27
28 000170 .LOC 170
29 00170 102441 OFF: SUBO 0,0,SKP ;START WITHOUT CAT
30 00171 102000 ON: ADC 0,0 ;START WITH CAT
31 00172 142470 ESTA 0,CATSW
32
33 00174 002175 JMP @+1
34 00175 000534 SWMESS
35
36 ; SPACE RESERVED FOR SPECIAL RESTART ENTRIES
37
38 000176 .LOC 176
39 00176 000002 .BLK 2
40
41 ; LOCATIONS 200 - 215 RESERVED FOR ECLIPSE TESTS
42
43
44 000200 .LOC 200
45 00200 002202 OT098: JMP @BGNADR ;PAGE ZERO STARTING LOC.
46 00201 000000 ITRET: 0 ;LAST TEST ENTERED
47 00202 000500 @GNADR: RSTRT ;POINTER TO TEST SETUP ROUTINES
48 00203 000000 PASS: 0 ;PASS COUNT
49 00204 000005 PASSIN: S. ;INTERNAL PASS COUNT
50 00205 000005 PASSVL: S. ;INITIAL VALUE, INT. PASS COUNT
51
52 00206 000000 ITR: 0 ;ITERATION VALUE FOR THIS TEST
53 00207 000000 ITRCT: 0 ;ITERATION COUNTER
54 00210 000000 ITRER: 0 ;ERROR SWITCH
55 00211 000000 ITRFC: 0 ;ERROR COUNTER
56 00212 000000 ERRET: 0 ;ERROR RETURN
57 00213 000000 LOPRET: 0 ;LAST PLACE LOOP EXECUTED
58 00214 000000 LOLOC: 0 ;RELOCATION VALUE
59 00215 000000 LISTNG: 0 ;LISTING ADDR OF FAILING TEST
60

```

```

0010 ECL30
01 ; *****LOCAL ZREL*****
02 ; *****
03 00216 005747 MINLOC: PRGEND ;END OF PROGRAM
04 00217 000000 MAXLOC: 0 ;RELOCATION RANGE OF PROGRAM
05 ;
06 ; POINTERS TO NON-RELOCATING SUBROUTINES
07
08 00220 000767 ICAL: CAL ;RELOCATED CALL SUBROUTINE
09 00221 001127 IMESS: MESS ;MESSAGE PRINT SUBROUTINE
10 00222 001001 ISIZE: SIZE ;MEMORY SIZING SUBROUTINE
11 00223 001066 IPOCT: POCY ;PRINT OCTAL SUBROUTINE
12 00224 001054 IPODEC: PDEC ;PRINT DECIMAL SUBROUTINE
13
14 ; TEMPORARYS FOR TESTS AND SUBROUTINES
15
16 00225 000000 AC0: 0 ;TEMPORARY STORAGE FOR
17 00226 000000 AC1: 0 ;MACHINE STATE IN "LOOP"
18 00227 000000 AC2: 0 ;AND "ERROR"
19 00230 000000 AC3: 0 ;SUBROUTINES.
20
21 00231 000000 CRY: 0 ;
22 00232 000000 CHARET: 0 ;RETURN SAVE CHAR ROUT.
23 00233 000000 CHORZ: 0 ;CHARACTER LINE COUNT
24 00234 000000 PDERET: 0 ;RETURN SAVE FR PRINT ROUT.
25 00235 000000 MESRET: 0 ;RETURN FR MESSAGE ROUT.
26 00236 123456 RAN: 123456 ;RANDOM NUMBER
27 00237 000000 CAL0: 0 ;TEMPORARYS FR CALL ROUT.
28 00240 000000 CAL1: 0 ;
29 00241 000000 CAL2: 0 ;
30 00242 000283 ABUF: -ABUF ;
31 00243 000010 ABUF: -BLK 10 ;
32 00254 000000 SVPC: 0 ;
33
34 ; ROUTINE TO GET PC ON STACK FAULT
35
36 00255 103210 STFL: POP 0,0 ;GET TRAP PC
37 00256 040254 STA 0,SVPC
38 00257 100010 ADI 3,0
39 00260 103110 PSH 0,0
40 00261 107710 POPB
41
42

```

```

1009 ECL30
01 ; ***** DIAGNOSTIC PROGRAM PREAMBLE *****
02 ; *****
03
04 .LOC 0 ;POINTER TO DIRTY BLOCK
05 00000 005776
06
07 .LOC 40 ;STACK CONTROL LOCATIONS
08
09 00040 000000 SP: 0
10 00041 000000 SF: 0
11 00042 000000 USL: 0
12 00043 000255 USF: STFL
13 00044 000000
14
15 .LOC 45 ;POINTER TO EGGS BLOCK
16 00045 005741 EGGS
17
18 .ZREL ;8 LOCATIONS RESERVED
19 00000-000010 ;FOR DEBUG BREAKPOINTS
20
21 00010-005741 .EGGS: EGGS ;PERMANENT POINTER TO EGGS
22 00011-000000 MEMTOP: 0 ;TOP OF MEMORY FROM SIZE
23 00012-000000 ICAT: 0 ;START LOCATION OF CAT
24
25 ; SPACE RESERVED FOR CAT/NO CAT RESTART ENTRIES
26
27
28 000170 .LOC 170
29 00170 102441 OFF: SUBO 0,0,SKP ;START WITHOUT CAT
30 00171 102000 ON: ADC 0,0 ;START WITH CAT
31 00172 142470 ESTA 0,CATSW
32
33 00174 002175 JMP @+1
34 00175 000534 SWMESS
35
36 ; SPACE RESERVED FOR SPECIAL RESTART ENTRIES
37
38 000176 .LOC 176
39 00176 000002 .BLK 2
40
41 ; LOCATIONS 200 - 215 RESERVED FOR ECLIPSE TESTS
42
43
44 000200 .LOC 200
45 00200 002202 OT098: JMP @BGNADR ;PAGE ZERO STARTING LOC.
46 00201 000000 ITRET: 0 ;LAST TEST ENTERED
47 00202 000500 @GNADR: RSTRT ;POINTER TO TEST SETUP ROUTINES
48 00203 000000 PASS: 0 ;PASS COUNT
49 00204 000005 PASSIN: S. ;INTERNAL PASS COUNT
50 00205 000005 PASSVL: S. ;INITIAL VALUE, INT. PASS COUNT
51
52 00206 000000 ITR: 0 ;ITERATION VALUE FOR THIS TEST
53 00207 000000 ITRCT: 0 ;ITERATION COUNTER
54 00210 000000 ITRER: 0 ;ERROR SWITCH
55 00211 000000 ITRFC: 0 ;ERROR COUNTER
56 00212 000000 ERRET: 0 ;ERROR RETURN
57 00213 000000 LOPRET: 0 ;LAST PLACE LOOP EXECUTED
58 00214 000000 LOLOC: 0 ;RELOCATION VALUE
59 00215 000000 LISTNG: 0 ;LISTING ADDR OF FAILING TEST
60

```

```

10011 ECL30
01
02
03
04 00262 113710 BAMBY: BAM ; ROUTINE TO RELOCATE TEST PROGRAM
05
06 00263 020214 LDA 0,RELOC ;BLOCK MOVE ON PROGRAM
07 00264 024047- LDA 1,=4
08 00265 030046- LDA 2,=BAMBLK
09 00266 034045- LDA 3,=BAMBLK+4
10 00267 113710 BAM ;ADJUST POINTERS TO
11 CALL ;RELOCATED SUBROUTINES
12 JSR @ICAL ;CALL SUBROUTINE
13 00270 006220
14 00271 001342 BEGIN
15
16 ; INITIAL LOCATIONS OF RELOCATING SUBROUTINES
17 00272 000566 BAMBLK: INIT
18 00273 000612 ;TEST INITIALIZER
19 00274 000666 ;TEST TERMINATOR
20 00275 000751 ERR ;ERROR ROUTINE
21 .RAND ;RANDOM NUMBER GEN.
22
23 ; CORRECTED POINTERS TO RELOCATED ROUTINES
24 00276 000566 ENTIN: INIT ;CORRECTED VALUE=
25 00277 000612 ENTLO: LOP ;INITIAL VALUE +
26 00300 000666 ENTER: ERR ;C(RELOC)
27 00301 000751 ENTRA: .RAND
28

```

```

10012 ECL30 .LOC 500
01 000500
02
03
04 ; *****SIZE SYSTEM & RESERVE MEMORY*****
05 00500 062677 NSTRT: IOMST @ISIZE
06 00501 066222 JSR 2,2 ;CALL SIZE ROUTINE
07 00502 150400 NEG 2,2 ;STORE ADDRESS OF LAST...
08 00503 150000 COM 2,2 ;LOCATION IN MEMTOP.
09 00504 050011- STA 2,MEMTOP
10
11 00505 125470 ELDA 0,CATSW ;TEST IF DTDS SET CATSW
12 005235 MOV 0,0,SMR ;IF 0 CAT WAS NOT LOADED.
13 00507 101005 JMP NOCAT
14 00510 000414
15
16 00511 122470 ELDA 0,PRGEND ;IF CAT WAS LOADED
17 005235
18 00513 024044- LDA 1,=1777 ;TEST FOR SUFFICIENT
19 00514 123000 ADD 1,0 ;MEMORY TO RUN CAT
20 00515 112035 AOCZ# 0,2,SRC
21 00516 000406 JMP NOCAT
22
23 00517 132400 SUB 1,2 ;TAKE 2K OFF THE TOP
24 00520 024043- LOA 1,=400 ;OF MEMORY FOR CAT
25 00521 147000 ADD 2,1
26 00522 044012- STA 1,ICAT ;SET UP CAT START ADDR.
27 00523 000406 JMP STLOC
28
29 00524 024042-NOCAT: LDA 1,=200 ;TAKE 200 OFF THE
30 00525 132400 SUB 1,2 ;TOP OF MEMORY FOR
31 00526 126400 SUB 1,1 ;MINI-MONITOR
32 00527 146470 ESTA 1,CATSW ;& CLEAR CATSW.
33 00528 005213
34
35 00531 024216 STLOC: LDA 1,MINLOC ;STORE PROGRAM
36 00532 132400 SUB 1,2 ;RELOCATION RANGE.
37 00533 050217 STA 2,MAXLOC

```

```

1:0013 ECL30
01
02
03
04 0534 06221 SWMES: JSR @IMES8
05 0535 01302 LDA 1,MEMTOP
06 0536 024011- MOV 0,0
07 0537 101040 JSR @IPOCT
08 0540 06223 JSR @IMES8
09 0541 06221 JSR @IMES8
10 0542 001316 KCRLF
11 0543 126400 SUB
12 0544 044203 STA 1,PASS
13
14 0545 126470 ELDA 1,AUTO
15 05173 MOV 1,1,SZR
16 0547 125004 JMP START
17 0550 000413
18
19 0551 006221 JSR @IMES8
20 0552 001330 SETSW
21 0553 06221 JSR @IMES8
22 0554 001316 KCRLF
23 0555 083077 HALT
24 0556 000401 JMP
25 0557 060477 READS
26 0560 142470 ESTA
27 0560 051165
28 0562 000401 JMP START
29
30 START:
31 0563 176400 SUB 3,3
32 0564 054214 STA 3,RELOC
33 0565 000263 JMP

```

```

: *****OUTPUT STRT MESSAGE & READ SWITCHES*****
:
: PRINT SIZE OF MEMORY
:
: INIT PASS COUNT
:
: RUNNING IN AUTO MODE?
:
: YES START PROGRAM.
:
: NO, PRINT SET SWITCHES MESS.
:
: READ NEW STATE OF SWITCHES
:
: SET RELOCATION CONSTANT TO 0
:
: AND START PROGRAM

```

```

1:0014 ECL30
01
02
03
04
05
06 00566 175400 INIT: INC 3,3
07 00567 054201 STA 3,ITRET
08 00570 040225 STA 0,ACO
09
10 00571 021777 LDA 0,-1,3
11 00572 040206 STA 0,ITR
12 00573 040207 STA 0,ITRCT
13
14 00574 020214 LDA 0,RELOC
15 00575 116400 SUB 0,3
16 00576 054215 STA 3,LISTING
17
18 00577 176400 SUB 3,3
19 00600 054210 STA 3,ITRER
20 00601 054211 STA 3,ITREC
21
22 00602 034203 LDA 3,PASS
23 00603 175004 MOV 3,3,SZR
24 00604 000404 JMP INIT1
25
26 00605 176520 SUB2L 3,3
27 00606 054206 STA 3,ITR
28 00607 054207 STA 3,ITRCT
29
30 00610 020225 INIT1: LOA 0,ACO
31 00611 002201 JMP @ITRET

```

```

: *****TEST UTILITY SUBROUTINES*****
:
: SUBROUTINE TO INITIALIZE A TEST LOOP
:
: TEST LOOP INITIALIZER
: SAVE RETURN LOCATION
: SAVE CONTENTS OF ACO
:
: GET # OF ITERATIONS
: SET ITER. VALUE
: SET ITER. COUNT
:
: COMPUTE AND SAVE
: THE LISTING ADDRESS
: OF THIS TEST.
:
: CLEAR ERROR SWITCH
: CLEAR ERROR COUNT
:
: TEST FOR FIRST PASS
:
: THIS IS 1'ST PASS
: SET ITERATIONS FOR
: 1 LOOP ONLY.
:
: RESTORE AC'S AND
: EXIT TO TEST

```

```

10015 ECL30
01
02
03
04 00612 054213 LOP: STA 3,LOPRET
05 00613 014207 ITRCT ITRCT
06 00614 000440 LOP3 LOP3
07 00615 034210 LDA 3,ITRER
08 00616 175005 MOV 3,3,SNR
09 00617 002213 JMP @LOPRET
10 00620 034206 LDA 3,ITR
11 00621 054207 STA 3,ITRCT
12
13 00622 074477 LOP1: READS 3
14 00623 175112 MOVL# 3,3,9ZC
15 00624 000403 *3
16 00625 136470 ELDA 3,3,SWREG
17 005120
18 00627 177100 ADDL
19 00630 177103 JMP 3,3,SNR
20 00631 000421 LOP2
21 00632 040225 STA 0,AC0
22 00633 044226 STA 1,AC1
23 00634 050227 STA 2,AC2
24 00635 006221 JSR @IMESS
25 00636 001234 PERCENT
26 00637 102400 SUB 0,0
27 00640 024211 LDA 1,ITREC
28 00641 040211 STA 0,ITREC
29 00642 030041 * MUL 2,100.
30 00643 143710 LDA 2,ITR
31 00644 030206 LDA 0,AC0
32 00645 153710 DIV JSR
33 00646 006224 * @IPDEC
34 00647 020225 LDA 0,AC0
35 00650 024226 LDA 1,AC1
36 00651 030227 LDA 2,AC2
37 00652 176400 LOP2: SUB 3,3
38 00653 054211 STA 3,ITREC
39
40 00654 034210 LOP3: LDA 3,ITRER
41 00655 175004 MOV 3,3,9ZR
42 00656 074477 READS 3,3,9ZC
43 00657 175112 MOVL# 3,3,9ZC
44 00660 000403 JMP *3
45 00661 136470 ELDA 3,3,SWREG
46 005064
47 00663 177113 ADDL# 3,3,SNR
48 00664 002201 JMP @ITRET
49 00665 002213 JMP @LOPRET

; SUBROUTINE TO TERMINATE A TEST LOOP
01
02
03
04 00666 054212 ERR: STA 3,ERRET
05 00667 040225 STA 0,AC0
06 00670 102560 SUBCL 0,0
07 00671 040231 STA 0,CRY
08 00672 010211 I&Z ITRC
09
10 00673 020210 LDA 0,ITRER
11 00674 116033 * 0,3,SNR
12 00675 000451 JMP ERR1
13
14 00676 054210 STA 3,ITRER
15 00677 044226 STA 1,AC1
16 00700 050227 STA 2,AC2
17 00701 006221 JSR @IMESS
18 00702 001241 ERM&G
19 00703 024203 LDA 1,PASS
20 00704 125420 IMCZ 1,1
21 00705 004224 JSR @IPDEC
22 00706 006221 JSR @IMESS
23 00707 001255 HEADER
24 00710 101020 MOVZ 0,0
25 00711 024231 LDA 1,CRY
26 00712 006224 JSR @IPDEC
27 00713 101040 MOV0 0,0
28 00714 024225 LDA 1,AC0
29 00715 006223 JSR @IPOCT
30 00716 024226 LDA 1,AC1
31 00717 006223 JSR @IPOCT
32 00720 024227 LDA 1,AC2
33 00721 006523 JSR @IPOCT
34 00722 024230 LDA 1,AC3
35 00723 006223 JSR @IPOCT
36
37 00724 024212 LDA 1,ERRET
38 00725 020214 LDA 0,RELOC
39 00726 106420 SUBZ 0,1
40 00727 006223 JSR @IPOCT
41 00730 024212 LDA 1,ERRET
42 00731 006523 JSR @IPOCT
43 00732 006521 JSR @IMESS
44 00733 001316 K&RLF
45
46 00734 024226 LDA 1,AC1
47 00735 030227 LDA 2,AC2
48 00736 136470 ELDA 3,AUTO
49 005002
50 00740 175005 MOV 3,3,SNR
51 00741 000405 JMP ERR1
52
53 00742 062677 IOR&T
54 00743 034010 * LDA 3,,EGGS
55 00744 035404 LDA 3,4,3
56 00745 001400 JMP 0,3
57
58 00746 020225 ERR1: LDA 0,AC0
59 00747 034230 LDA 3,AC3
60 00750 002212 JMP @ERRET

; ERROR SUBROUTINE: SETS ITRER FLAG, PRINTS MACHINE STATE.
10016 ECL30
01
02
03
04 00666 054212 ERR: STA 3,ERRET
05 00667 040225 STA 0,AC0
06 00670 102560 SUBCL 0,0
07 00671 040231 STA 0,CRY
08 00672 010211 I&Z ITRC
09
10 00673 020210 LDA 0,ITRER
11 00674 116033 * 0,3,SNR
12 00675 000451 JMP ERR1
13
14 00676 054210 STA 3,ITRER
15 00677 044226 STA 1,AC1
16 00700 050227 STA 2,AC2
17 00701 006221 JSR @IMESS
18 00702 001241 ERM&G
19 00703 024203 LDA 1,PASS
20 00704 125420 IMCZ 1,1
21 00705 004224 JSR @IPDEC
22 00706 006221 JSR @IMESS
23 00707 001255 HEADER
24 00710 101020 MOVZ 0,0
25 00711 024231 LDA 1,CRY
26 00712 006224 JSR @IPDEC
27 00713 101040 MOV0 0,0
28 00714 024225 LDA 1,AC0
29 00715 006223 JSR @IPOCT
30 00716 024226 LDA 1,AC1
31 00717 006223 JSR @IPOCT
32 00720 024227 LDA 1,AC2
33 00721 006523 JSR @IPOCT
34 00722 024230 LDA 1,AC3
35 00723 006223 JSR @IPOCT
36
37 00724 024212 LDA 1,ERRET
38 00725 020214 LDA 0,RELOC
39 00726 106420 SUBZ 0,1
40 00727 006223 JSR @IPOCT
41 00730 024212 LDA 1,ERRET
42 00731 006523 JSR @IPOCT
43 00732 006521 JSR @IMESS
44 00733 001316 K&RLF
45
46 00734 024226 LDA 1,AC1
47 00735 030227 LDA 2,AC2
48 00736 136470 ELDA 3,AUTO
49 005002
50 00740 175005 MOV 3,3,SNR
51 00741 000405 JMP ERR1
52
53 00742 062677 IOR&T
54 00743 034010 * LDA 3,,EGGS
55 00744 035404 LDA 3,4,3
56 00745 001400 JMP 0,3
57
58 00746 020225 ERR1: LDA 0,AC0
59 00747 034230 LDA 3,AC3
60 00750 002212 JMP @ERRET

;PRINT PC OF ERROR
;SAVE TEMPORARY
;AC0 AND CARRY
;BUMP ERROR COUNT
;NEW ERROR?
;NO, RETURN.
;YES,
;SAVE MACHINE STATE
;PRINT "ERROR PASS" MSG.
;SET LEADING ZERO SUPR.
;PRINT MACHINE STATE
;PRINT LOGICAL &
;(LISTING PC)
;(LOGICAL PC)
;FLUSH DEVICE BUFFERS
;RESTORE AC'S 1 & 2
;PROGRAM IN AUTO MODE?
;NOPE: GO BACK TO TEST
;AUTO MODE & SW6=0
;MAKE ERROR RETURN TO OTOS
;FINISH RESTORING AC'S
;GO BACK TO TEST

```

```

0017 ECL30
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

; RANDOM NUMBER GENERATOR SUBROUTINE
-RAND: LDA 0,RAN
1,ITRER
MOV 1,1,SZR
JMP 0,3
HXL 0,1
ADD 2,1
MOVZL 1,1
ADD 1,0
LDA 1,=35031
ADD 1,0
STA 0,RAN
JMP 0,3

; CALL ROUTINE TO REACH RELOCATED SUBROUTINES
CAL: STA 0,CAL0
1,CAL1
LDA 0,RELOC
LDA 1,0,3
ADD 1,0
STA 0,CAL2
INC 3,3
LDA 0,CAL0
1,CAL1
JMP 0,CAL2

; SIZING SUBROUTINE: RETURNS SIZE OF LOGICAL MEM IN AC2
SIZE: LDA 2,MINLOC
MOV# 2,2,SZC
JMP 0,6
LDA 0,0,2
STA 2,0,2
LDA 1,0,2
STA 0,0,2
SUB# 1,2,SZR
JMP 0,3
STA 2,MENTOP
JMP SIZE+1

; RELOCATE SUBROUTINE: ALLOCATES MEMORY FOR COPIES
; OF TEST PROGRAM, WORKS IN CONJUNCTION
; WITH BAW ROUTINE IN PAGE ZERO, WHICH
; ACTUALLY COPIES THE TEST PROGRAM TO
; THE NEW LOCATION.
EISZ CATSW
JMP *2
JSR @ICAT
RAND
JSR @ENTRA
LDA 1,MAXLOC
LDA 2,MINLOC
SUB# 2,1,SNC
JMP REL2
SUBZ 1,0,SZC
ADD 0,1
SUB# 2,1,SNC
JMP REL
LDA 0,RELOC
SUB# 1,0,SZC
JMP REL1
SUB 0,1
SUB# 1,2,SNC
JMP *+3
LDA 1,MAXLOC
SUB 0,0,SKP
ADD 0,1
STA 1,RELOC
SUB 3,3
XCH 0,3
XCH 2,1
XCH 2,3
JMP @BAMBY
CALL @ICAT
JSR @ICAT
BEGIN
REL:
REL0:
REL1:
REL2:

```

```

10018 ECL30
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40

; RELOCATE SUBROUTINE: ALLOCATES MEMORY FOR COPIES
; OF TEST PROGRAM, WORKS IN CONJUNCTION
; WITH BAW ROUTINE IN PAGE ZERO, WHICH
; ACTUALLY COPIES THE TEST PROGRAM TO
; THE NEW LOCATION.
EISZ CATSW
JMP *2
JSR @ICAT
RAND
JSR @ENTRA
LDA 1,MAXLOC
LDA 2,MINLOC
SUB# 2,1,SNC
JMP REL2
SUBZ 1,0,SZC
ADD 0,1
SUB# 2,1,SNC
JMP REL
LDA 0,RELOC
SUB# 1,0,SZC
JMP REL1
SUB 0,1
SUB# 1,2,SNC
JMP *+3
LDA 1,MAXLOC
SUB 0,0,SKP
ADD 0,1
STA 1,RELOC
SUB 3,3
XCH 0,3
XCH 2,1
XCH 2,3
JMP @BAMBY
CALL @ICAT
JSR @ICAT
BEGIN
REL:
REL0:
REL1:
REL2:

```

```

10019 ECL30
01 ;*****PRINT ROUTINES*****
02 ;
03
04 01054 175100 PDEC: MOVL 3,3 ;DECIMAL PRINT C(AC1).
05 01055 054234 STA 3,PDERET
06 01056 175200 MOVR 3,3
07 01057 004441 JSR PDEC3 ;RESET C(CARRY) FOR ZERO SUPPRESSION
08 01060 023420 10000. ;SET C(CARRY) IF NOT
09 01061 001750 1000.
10 01062 000144 100.
11 01063 000012 10.
12 01064 000001 1.
13 01065 000000 0.
14 01066 175100 POCT: MOVL 3,3 ;OCTAL PRINT C(AC1)
15 01067 054234 STA 3,PDERET
16 01070 175200 MOVR 3,3
17 01071 004427 JSR PDEC3 ;RESET C(CARRY) FOR ZERO SUPPRESSION
18 01072 100000 100000. ;SET C(CARRY) IF NOT
19 01073 010000 10000.
20 01074 001000 1000.
21 01075 000100 100.
22 01076 000010 10.
23 01077 000001 1.
24 01100 000000 0.
25
26 01101 020037-PDEC1: LDA 0,-11 ;PRECESS WITH TAB
27 01102 031377 LDA 2,-1,2
28 01103 151015 MOV# 2,2,SNR
29 01104 000415 JMP PDEC3+1
30 01105 102060 SUBC 0,0
31 01106 146452 SUBO# 2,1,SZC
32 01107 000404 JMP PDEC2
33 01110 146420 SUBZ 2,1
34 01111 101400 INC 0,0
35 01112 000774 JMP *-4
36 01113 151234 POEC2: MOV# 2,2,SZR
37 01114 152462 SUBC 2,2,SZC
38 01115 030036- LDA 2,-60
39 01116 143004 ADD 2,0,SZR
40 01117 171401 INC 3,2,SKP
41 01120 171401 INC 3,2,SKP
42 01121 004423 JSR CHAR
43 01122 155004 MOV 2,3,SZR
44 01123 000756 JMP PDEC1
45 01124 034234 LDA 3,PDERET
46 01125 175200 MOVR 3,3
47 01126 001400 JMP 0,3
48
49 01127 175400 MESS: INC 3,3 ;MESSAGE PRINTER
50 01130 054235 STA 3,MESS ;JSR (MESS)
51 01131 031777 LDA 2,-1,3 ;MESSAGE ADDRESS
52 01132 020035-MESS1: LDA 0,-377
53 01133 025000 LDA 1,0,2
54 01134 151420 INCZ 2,2
55 01135 123400 AND 1,0
56 01136 106700 SUBS 0,1
57 01137 004405 JSR CHAR
58 01140 121005 MOV 1,0,SNR
59 01141 062235 JMP 2,MESSRET
60 01142 004402 JSR CHAR

```

0022 ECL30  
 01 01225 000746  
 02  
 03 01226 000000 CHRSV: 0

JMP CHAR1

! LPT/TTO INTERFACE ROUTINE: CHARACTER PASSED IN AC3

!SAVE RETURN ADDR.  
 !SAVE AC2

!READ SWITCHES INTO AC3  
 !TEST SWITCH 0  
 ? SMO SET  
 ; SMO CLEAR, DEFAULT

!MASK SW2 A, SWS INTO  
 !AC2 FROM AC3  
 !LEFT JUSTIFY SW2  
 !COMPLEMENT SW2  
 !NO OUTPUT, RETURN

!MASK CHARACTER INTO L-BYTE  
 !DF AC3, CLEAR CARRY.  
 !IF NULL CHAR. RETURN

```

!0021 ECL30      ! LPT/TTO INTERFACE ROUTINE: CHARACTER PASSED IN AC3
01 01144 175100 CHAR: 3,3
02 01145 058232 STA 3,CHAR2
03 01146 050460 STA 3,CHRSV
04 01147 074477 READS 3
05 01150 175112 MOVL# 3,3,SZC
06 01151 000403 JMP *3
07 01152 136470 ELDA 3,SMRES
08 01154 030034 LDA 2,=22000
09 01155 175400 AND 3,2
10 01156 177260 ADDR 2,2
11 01157 153265 ADDCR 2,2,SNR
12 01160 000435 JMP REST
13 01161 034035 LDA 3,=377
14 01162 11725 ANDZS 0,3,SNR
15 01163 000432 JMP REST
16 01164 103004 ADD 0,0,SZR
17 01165 000777 JMP *-1
18 01166 177260 ADDCR 3,3
19 01167 020033 LDA 0,=211*400
20 01170 162445 SUBO 3,0,SNR
21 01171 000430 JMP CHAR4
22 01172 161340 MOVOS 3,0
23 01173 010233 CHAR1: ISZ CHORZ
24 01174 151135 MOVZL# 2,2,SNR
25 01175 000405 JMP CHAR2
26 01176 061117 DOAS 0,LPT
27 01177 063517 SKPBZ LPT
28 01200 000777 JMP *-1
29 01201 060217 NIOC LPT
30 01202 151133 CHAR2: MOVZL# 2,2,SNC
31 01203 000405 JMP CHAR3
32 01204 061111 DOAS 0,TTO
33 01205 063511 SKPBZ TTO
34 01206 000777 JMP *-1
35 01207 060211 NIOC TTO
36 01210 175403 CHAR3: INC 3,3,SNC
37 01211 000762 JMP CHAR1
38 01212 030032 LDA 2,=212
39 01213 142405 SUB 2,0,SNR
40 01214 040233 STA 0,CHORZ
41 01215 030411 REST: LDA 2,CHRSV
42 01216 034232 LDA 3,CHAR2
43 01217 175200 MOVR 3,3
44 01220 001400 JMP 0,3
45 01221 034233 CHAR4: LDA 3,CHORZ
46 01222 020031 LDA 0,=-8
47 01223 114410 TOR 0,3
48 01224 020030 LDA 0,=-240

```

!DTERMINE REQUIRED  
 !STATE OF PARITY BIT &  
 !INSERT IT  
 !TEST FOR TAB  
 !TRUE: SETUP TAB SIMULATION

!RESTORE CHARACTER TO R-BYTE AC0,  
 !SET CARRY, BUMP LINE COUNT.

!SEND TO LPT?  
 !NOPE, MUST BE TTO  
 !O.K. PRINT CHARACTER  
 !WAIT FOR DONE  
 !CLEAR DEVICE

!SEND TO TTO?  
 !O.K. SEND CHARACTER  
 !WAIT FOR DONE  
 !CLEAR DEVICE

!IF TABBING, AND NOT  
 !FINISHED, LOOP.  
 !TEST FOR CR/LF  
 !TRUE: RESET LINE COUNT  
 !OTHERWISE RETURN

!SET UP TO TAB  
 !C(AC3) IS TWO'S COMPLEMENT  
 !OF # OF SPACES NEEDED

```

0024 ECL30
01 01320 142523 SETSM: .TXTE ISET DATA SWITCHS AND PRESS CONTINUE!
02 120324
03 040304
04 040724
05 051640
06 144727
07 141724
08 051510
09 040640
10 042116
11 050240
12 142722
13 051523
14 141640
15 047317
16 144724
17 052516
18 000305

```

```

10023 ECL30
01
02
03 *****MESSAGE DATA BLOCK*****
04 01227 045215 PASHES: .TXTE !<15><12>PASS I
05 040520
06 051523
07 120240
08 000000
09 01234 052115 PERGEN: .TXTE !<15><12>% FAIL=1
10 120245
11 040706
12 146311
13 000275
14 01241 052115 ERMSG: .TXTE !<15><12><15><12>ERROR IN PASS: I
15 045215
16 151305
17 147722
18 120322
19 047311
20 050240
21 051501
22 035123
23 000240
24 01253 005215 HEADER: .TXTE !<15><12><15><12>
25 052115
26 01255 151303 CRY ACO AC1 AC2 AC3 LISTING LOGICAL<15><12>:
27 04331
28 141501
29 120060
30 040411
31 130703
32 004640
33 141501
34 120242
35 040411
36 031703
37 146011
38 051711
39 144724
40 043516
41 146011
42 043717
43 141711
44 146101
45 005215
46 000000
47 01302 005215 MESIZ: .TXTE !<15><12>LAST LOGICAL ADDRESS=I
48 040714
49 152123
50 146240
51 043717
52 141711
53 146101
54 040640
55 042104
56 142722
57 051523
58 000275
59 01316 005215 KCRLF: .TXTE !<15><12>I
60 000000

```



```

10025 ECL30
01 02 01342 101000 BEGIN: MOV 0,0
03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
*****FIRST TEST*****
; *** ADD IMMEDIATE ***
EX0:
09 01343 006276 ;INITIALIZE TEST.
10 01344 000100 JSR @ENTIN
11 01345 102420 SUBZ 0,0
12 01346 163770 ADDI 1,0
13 01350 000001 MOV 0,0,SBN
14 01350 101007 ERROR
15 01355 006277 JSR @ENTLO
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
EX1:
09 01356 006276 ;INITIALIZE TEST.
10 01357 000100 JSR @ENTIN
11 01358 102420 SUBZ 0,0
12 01360 105920 ADDZ 0,0
13 01361 163770 ADDI 1,0
14 01363 101002 MOV 0,0,SZC
15 01370 101004 ERROR
16 01375 006277 JSR @ENTLO
17 01376 102400 EX2:
18 01377 006276 JSR @ENTIN
19 01400 000010 LOOP
20 01401 105040 EX2.1:
21 01402 167770 ADDI 1,1
22 01404 106014 ADC# 0,1,SZR
23 01411 101003 ERROR
24 01416 034210 JMPER EX2.2
25 01417 175004 LDA 3,I,TRER
26 01420 000403 MOV 3,S,SZR
27 01421 101704 INCS 0,0,SZR
28 01422 000757 JMP EX2.1
29 01423 006277 JSR @ENTLO
30 01425 006277 EX2.2:
31 01426 102400 EX3:
32 01427 006276 JSR @ENTIN
33 01428 000010 LOOP
34 01429 105020 EX3.1:
35 01430 173770 ADDI 2,2
36 01432 151003 MOV 2,S,SNC
37 01433 000002 ERROR
38 01437 105400 INC 0,1
39 01440 125400 SUB# 1,2,SZR
40 01441 132414 JMPER EX3.2
41 01446 034210 LDA 3,I,TRER
42 01447 175004 MOV 3,S,SZR
43 01450 000403 JMP EX3.2
44 01451 101704 INCS 0,0,SZR
45 01452 000755 JMP EX3.1
46 01453 006277 JSR @ENTLO
47 01454 102400 EX4:
48 01455 006276 JSR @ENTIN
49 01457 115020 EX4.1:
50 01457 177770 ADDI 3,3
51 01462 105600 INCR 0,1
52 01463 125500 INCL 1,1
53 01464 171000 MOV 3,2
54 01465 132414 SUB# 1,2,SZR
55 01472 034210 JMPER EX4.2
56 01473 175004 LDA 3,I,TRER
57 01474 000403 MOV 3,S,SZR
58 01475 101704 INCS 0,0,SZR
59 01476 000761 JMP EX4.1
60 01477 006277 JSR @ENTLO
61 01500 126400 EX5:
62 01501 006276 JSR @ENTIN
63 01502 000010 LOOP
64 01503 121020 EX5.1:
65 01504 163770 ADDI 4,0
66 01506 131600 INCR 1,2
67 01507 151500 INCL 2,2
68 01510 151400 SUB# 2,0,SZR
69 01511 142414 JMPER EX5.2
70 01516 034210 LDA 3,I,TRER
71 01517 175004 MOV 3,S,SZR

```

```

10026 ECL30
01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
; *ADDI* TEST
;INITIALIZE TEST.
;SET C(AC2)=C(AC0)
;ADD #2 TO C(AC2)
;ADDI INSTRUCTION
;SHOULD NOT CHANGE C(CARRY)
;C(AC0)=ORIG
;C(AC1)=CORRECT
;C(AC2)=ADDI RESULT
;IF A ERROR PRESENT
;JMP TO EX3.2
;ITERATE TEST ROUTINE
; *ADDI* TEST
;INITIALIZE TEST.
;SET C(AC3)=C(AC0)
;C(AC3)=C(AC3)+5
;THE C(AC1) IS MADE
;3 GREATER THAN C(AC0).
;C(AC0)=ORIG C(AC3)
;C(AC1)=CORRECT
;C(AC2)=ADDI RESULT
;IF A ERROR PRESENT
;JMP TO EX4.2
;ITERATE TEST ROUTINE
; *ADDI* TEST
;INITIALIZE TEST.
;SET C(AC0)=C(AC1)
;SHOULD ADD (4) TO C(AC0)
;C(AC2) IS MADE 4 GREATER
;THEN C(AC1).
;C(AC2)=CORRECT
;C(AC1)=ORIG
;C(AC0)=ADDI RESULT
;IF A ERROR PRESENT

```



```

0029 ECL30
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

JMPER EX6.2
LDA 3,IYRER
MOV 3,SZRSZR
JMP EX6.2
INCS 1,I,SZR
JMP EX6.1
LOOP
JSR @ENTLO
EX6.2:
08 01605 006277
09

10030 ECL30
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

:TEST "ADDI" BY ADDITION
:INITIALIZE TEST.
:OF ALL VALUES TO ALL
:ACCUMULATORS
:C(AC1) IS 1 GREATER THAN
:C(AC0), C(AC2) IS 1
:GREATER THAN C(AC1), ETC.
:ADD TO ALL AC'S

:ITERATE TEST ROUTINE

SETUP 100
JSR @ENTIN
100
SUB 0,0
INC 0,1
INC 1,2
INC 2,3
ADDA 1
ADDI 1,0
ADDI 1,1
ADDI 1,2
ADDI 1,3
ADDA 2
ADDI 2,0
ADDI 2,1
ADDI 2,2
ADDI 2,3
ADDA 3
ADDI 3,0
ADDI 3,1
ADDI 3,2
ADDI 3,3
ADDA 4
ADDI 4,0
ADDI 4,1
ADDI 4,2
ADDI 4,3
ADDA# 0,1,SZR
ERROR
ADDA# 1,2,SZR
ERROR
MOV 3,0
ADDA# 2,0,SZR
ERROR
LOOP
JSR @ENTLO
EX7:
04 01606 006276
05 01607 000100
06 01610 102400
07 01611 105400
08 01612 131400
09 01613 155400
10 01614 163770
11 01614 000001
12 01616 167770
13 01616 000001
14 01620 173770
15 01620 000001
16 01622 177770
17 01622 000001
18 01624 163770
19 01624 000002
20 01626 167770
21 01626 000002
22 01630 173770
23 01630 000002
24 01632 177770
25 01632 000002
26 01634 163770
27 01634 000003
28 01636 167770
29 01636 000003
30 01640 173770
31 01640 000003
32 01642 177770
33 01642 000003
34 01644 163770
35 01644 000004
36 01646 167770
37 01646 000004
38 01650 173770
39 01650 000004
40 01652 177770
41 01652 000004
42 01654 106014
43 01654 132014
44 01661 132014
45 01666 161900
46 01667 142014
47 01667 006277

```

```
10031 ECL30
01
02 01675 102400 EXD: SUB 0,0
03 SETUP 10
04 01676 006276 JSR @ENTLO
05 01677 000010 10
06 01700 111020 EXD.1: MOVZ 0,2
07 000016 J00 16
08 01701 150110 SBI 3,2
09 01702 173770 ADDI 3,2
10 000003
11 01704 150110 SBI 3,2
12 01705 173770 ADDI 3,2
13 000003
14 01707 150110 SBI 3,2
15 01710 173770 ADDI 3,2
16 000003
17 01712 150110 SBI 3,2
18 01713 173770 ADDI 3,2
19 000003
20 01715 150110 SBI 3,2
21 01716 173770 ADDI 3,2
22 000003
23 01720 150110 SBI 3,2
24 01721 173770 ADDI 3,2
25 000003
26 01723 150110 SBI 3,2
27 01724 173770 ADDI 3,2
28 000003
29 01726 150110 SBI 3,2
30 01727 173770 ADDI 3,2
31 000003
32 01731 150110 SBI 3,2
33 01732 173770 ADDI 3,2
34 000003
35 01734 150110 SBI 3,2
36 01735 173770 ADDI 3,2
37 000003
38 01737 150110 SBI 3,2
39 01740 173770 ADDI 3,2
40 000003
41 01742 150110 SBI 3,2
42 01743 173770 ADDI 3,2
43 000003
44 01745 150110 SBI 3,2
45 01746 173770 ADDI 3,2
46 000003
47 01750 150110 SBI 3,2
48 01751 173770 ADDI 3,2
49 000003
50 01753 112414 SUBW 0,2,SZR
51 ERROR
52 01760 151002 MOV 2,2,SZC
53 ERROR
54 JIMPER EXD.2
55 01765 034210 LDA 3,1YR
56 01766 175004 MOV 3,3,SZR
57 01767 000403 JMP EXD.2
58 01770 101704 INCS 0,0,SZR
59 01771 000707 JMP EXD.1
60
```

:TEST "ADDI"

:INITIALIZE TEST.

:SUBTRACT 3

:ADD 3

:SUBTRACT 3

:ADD 3

:SUBTRACT 3

:ADD 3

:SUBTRACT 3

:ADD 3

:SUBTRACT 3

:ADD 3

:SUBTRACT 3

:ADD 3

:SUBTRACT 3

:ADD 3

:SUBTRACT 3

:ADD 3

:SUBTRACT 3

:ADD 3

:SUBTRACT 3

:ADD 3

:SUBTRACT 3

:ADD 3

:SUBTRACT 3

:ADD 3

:SUBTRACT 3

:ADD 3

:RESULT SHOULD NOT BE  
:CHANGED BY +-3  
:C(0)=ORIG, C(AC2)=ERROR  
:CARRY SHOULD NOT CHANGE.

:IF A ERROR PRESENT  
:JMP TO EXD.2

EXD.2: LOOP

```

10033 ECL30
01 02037 006276
02 02040 000100
03 01775 126440
04 01776 107770
05 01777 000000
06 02000 101003
07 02001 125004
08 02006 006277
09 02007 006276
10 02010 000100
11 02011 152420
12 02012 113770
13 02014 101002
14 02015 150014
15 02022 006277
16 02023 006276
17 02024 000100
18 02025 175040
19 02026 117770
20 02030 165002
21 02031 174014
22 02036 006277
23 02037 006276
24 02040 000100
25 02041 102040
26 02042 103770
27 02044 101002
28 02045 100014
29 02052 006277
30 02053 102400
31 02054 006276
32 02055 000005
33 02056 105400
34 02057 131400
35 02060 155400
36 02061 044402
37 02062 107770
38 02064 050402
39 02065 113770
40 02070 117770
41 02100 156014
42 02105 034210
43 02106 175004
44 02107 000403
45 02110 101704
46 02111 000745
47 02112 006277
48 02113 066276
49 02114 000010
50 02115 102400
51 02116 126400
52 02117 152400
53 02120 175440
54 02121 103770
55 02123 107770
56 02125 113770
57 02127 117770
58 02131 106414
59 02136 112415
60 02137 156414
61 02144 006277
62 02053 102400
63 02054 006276
64 02055 000005
65 02056 105400
66 02057 131400
67 02060 155400
68 02061 044402
69 02062 107770
70 02064 050402
71 02065 113770
72 02070 117770
73 02100 156014
74 02105 034210
75 02106 175004
76 02107 000403
77 02110 101704
78 02111 000745
79 02112 006277
80 02113 066276
81 02114 000010
82 02115 102400
83 02116 126400
84 02117 152400
85 02120 175440
86 02121 103770
87 02123 107770
88 02125 113770
89 02127 117770
90 02131 106414
91 02136 112415
92 02137 156414
93 02144 006277
94 02053 102400
95 02054 006276
96 02055 000005
97 02056 105400
98 02057 131400
99 02060 155400
100 02061 044402
101 02062 107770
102 02064 050402
103 02065 113770
104 02070 117770
105 02100 156014
106 02105 034210
107 02106 175004
108 02107 000403
109 02110 101704
110 02111 000745
111 02112 006277
112 02113 066276
113 02114 000010
114 02115 102400
115 02116 126400
116 02117 152400
117 02120 175440
118 02121 103770
119 02123 107770
120 02125 113770
121 02127 117770
122 02131 106414
123 02136 112415
124 02137 156414
125 02144 006277

```

```

:TEST ORI
:INITIALIZE TEST.
:SET C(AC1) TO 1 GREATER
:THEN C(AC0), ETC.
:INCLUSIVE OR OF A
:NUMBER WITH ITSELF
:SHOULD NOT CHANGE
:THE NUMBER.
:C(AC1) SHOULD=C(AC0)+1
:C(AC2) SHOULD=C(AC0)+2
:ETC.
:IF A ERROR PRESENT
:JMP TO EXJ.2
:512 ITERATE.
:ITERATE TEST ROUTINE
:TEST "IORI"
:INITIALIZE TEST.
:CLEAR ALL AC'S
:AND C(CARRY)
:C(AC0,1,2,3)=0
:INCLUSIVE OR FROM 0 TO 1
:ALL AC'S SHOULD=(-1)
:AFTER THE ORI INSTRUCTIONS.
:ITERATE TEST ROUTINE

```

```

SUB 0,0
SETUP 5
JSR @ENTIN
ASSEND
INC 0,1
INC 1,2
INC 2,3
STA 1,*,2
IORI 1,1
STA 2,*,2
IORI 2,2
STA 3,*,2
IORI 3,3
ADC# 0,1,SNR
ADC# 1,2,SZR
ERROR
ADC# 2,3,SZR
JMPER EXJ.2
LDA 3,ITRER
MOV 3,3,SZR
JMP EXJ.2
INCS 0,0,SZR
JMP EXJ.1
LOOP
JSR @ENTLO
SETUP 10
JSR @ENTIN
10
ZEROGAC
SUB 0,0
SUB 1,1
SUB 2,2
SUB 3,3
IORI -1,0
IORI -1,1
IORI -1,2
IORI -1,3
SUB# 0,1,SZR
ERROR
SUB# 0,2,SNR
SUB# 2,3,SZR
ERROR
LOOP
JSR @ENTLO

```

```

:INITIALIZE TEST.
:(AC1) AND CARRY ARE CLEARED.
:0 INCLUSIVE OR TO 0
:SHOULD GIVE RESULT OF 0.
:C(CARRY) SHOULD NOT CHANGE.
:0 OR 0
:ITERATE TEST ROUTINE
:INITIALIZE TEST.
:CLEAR C(AC2), SET C(CARRY)
:0 INCLUSIVE OR TO 1
:SHOULD GIVE RESULT OF 1
:C(CARRY) SHOULD NOT CHANGE.
:0 OR 1
:ITERATE TEST ROUTINE
:TEST "IORI"
:INITIALIZE TEST.
:SET C(AC3)
:1 INCLUSIVE OR TO 0
:SHOULD GIVE RESULT OF 1.
:C(CARRY) SHOULD NOT CHANGE.
:1 OR 0
:ITERATE TEST ROUTINE
:TEST "IORI"
:INITIALIZE TEST.
:SET C(AC0), SET C(CARRY).
:1 INCLUSIVE OR TO 1
:SHOULD GIVE RESULT OF 1.
:C(CARRY) SHOULD NOT CHANGE.
:ITERATE TEST ROUTINE

```

```

SETUP 100
JSR @ENTIN
100
SUB 1,1
IORI 0,1
MOV 0,0,SZC
MOV 1,1,SZR
ERROR
LOOP
JSR @ENTLO
SETUP 100
JSR @ENTIN
100
SUB 2,2
IORI -1,2
MOV 0,0,SZC
COM# 2,2,SZR
ERROR
LOOP
JSR @ENTLO
SETUP 100
JSR @ENTIN
100
ADC 3,3
IORI 0,3
MOV 3,1,SZC
COM# 3,3,SZR
ERROR
LOOP
JSR @ENTLO
SETUP 100
JSR @ENTIN
100
ADC 0,0
IORI -1,0
MOV 0,0,SZC
COM# 0,0,SZR
ERROR
LOOP
JSR @ENTLO

```

```

10035 ECL30
01
02 02145 006276
03 02146 000010
04 02147 102400
05 02150 126400
06 02151 152400
07 02152 176400
08 02153 176400
09 02154 094402
10 02155 103770
11 02156 000000
12 02157 054402
13 02160 107770
14 02161 113770
15 02162 054402
16 02163 113770
17 02165 101002
18 02166 116414
19 02173 122415
20 02174 142414
21 02201 006277
22
23
24
25
26
27 02202 006276
28 02203 000005
29
30
31
32
33
34 02204 102400
35 02205 126400
36 02206 152400
37 02207 176440
38 02210 102000
39 02211 103770
40 02213 133015
41 02214 175014
42
43
44
45 02221 006277

EXM:
SETUP 10
JSR @ENTIN
10
ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUB0 3,3
ADCO 3,3
STA 3,*,*2
IORI 0,0
STA 3,*,*2
IORI 0,1
STA 3,*,*2
IORI 0,2
MOV 0,0,SZC
SUB# 0,3,SZR
ERROR
SUB# 1,0,SNR
SUB# 2,0,SZR
LOOP
JSR @ENTLO
SETUP 5
JSR @ENTIN
5
ACD IORI,0
ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUB0 3,3
ADC 0,0
IORI -1,0
ADD# 0*183,0*283,SNR
MOV# 0*383,0*383,SZR
ERROR
LOOP
JSR @ENTLO

;TEST "IORI"
;INITIALIZE TEST.
;CLEAR ALL AC'S
;AND C(CARRY)
;THE VALUE IN AC0
;SHOULD NOT EFFECT
;OTHER AC'S. CHECK
;AC DESTINATION FOR
;THE IORI INSTRUCTION.
;ITERATE TEST ROUTINE

10036 ECL30
01
02 02222 006276
03 02223 000005
04
05
06 02224 102400
07 02225 126400
08 02226 152400
09 02227 176440
10 02230 126000
11 02231 107770
12 02233 157015
13 02234 101014
14
15
16
17 02241 006277
18
19
20 02242 006276
21 02243 000005
22
23
24 02244 102400
25 02245 126400
26 02246 152400
27 02247 176440
28 02250 152000
29 02251 113770
30
31 02253 163015
32 02254 125014
33
34
35 02261 006277
36
37
38 02262 006276
39 02263 000005
40
41
42 02264 102400
43 02265 126400
44 02266 152400
45 02267 176440
46 02270 176000
47 02271 117770
48
49 02273 107015
50 02274 151014
51
52
53 02301 006277
54

EXO:
SETUP 5
JSR @ENTIN
5
ACD IORI,1
ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUB0 3,3
ADC 1,1
IORI -1,1
ADD# 1*183,1*283,SNR
MOV# 1*383,1*383,SZR
ERROR
LOOP
JSR @ENTLO
ITERATE TEST ROUTINE
SETUP 5
JSR @ENTIN
5
ACD IORI,2
ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUB0 3,3
ADC 2,2
IORI -1,2
ADD# 2*183,2*283,SNR
MOV# 2*383,2*383,SZR
ERROR
LOOP
JSR @ENTLO
ITERATE TEST ROUTINE
SETUP 5
JSR @ENTIN
5
ACD IORI,3
ZEROAC
SUB 0,0
SUB 1,1
SUB 2,2
SUB0 3,3
ADC 3,3
IORI -1,3
ADD# 3*183,3*283,SNR
MOV# 3*383,3*383,SZR
ERROR
LOOP
JSR @ENTLO
ITERATE TEST ROUTINE

```

```
10037 ECL30      EXR:      SETUP 400      TEST IORI      10038 ECL30      EXU:
01 02 02302 006276  JSR @ENTIN    :INITIALIZE TEST. 01
02 03 02303 000400  400           :INITIALIZE TEST. 02
03 04 02304 006501  RAND          :C(CAC0)=RANDOM # IORI 03
04 05 02305 115300  JSR @ENTRA   :TROTH TABLE FOR IORI 04
05 06 02306 111300  MOV# 0,3     :0011 05
06 07 02307 105300  MOV# 0,2     :0101 06
07 08 02308 105300  MOV# 0,1     :C(AC0)=RANDOM # IORI 07
08 09 02310 100000  COM 0,0      :01011 08
09 10 02311 117400  AND 0,3      :GENERATE INCLUSIVE OR 09
10 11 02312 116000  ADC 0,3      :VIA SOFTWARE AND THE 10
11 12 02313 100000  COM 0,0      :IORI INSTRUCTION 11
12 13 02314 040402  STA 0,+2     :C(AC0)=ORIG AC9 12
13 14 02315 113770  IORI 0,2     13
14 15 02317 156414  SUB# 2,3,SZR 14
15 16 02317 156414  ERROR        :ITERATE TEST ROUTINE 15
16 17 02317 156414  LOOP        16
17 18 02324 006277  JSR @ENTLO  17
18 19 02324 006277  18
19 20 02324 006277  19
20 21 02325 006276  EXT:         :TEST *IORI* 20
21 22 02325 006276  JSR @ENTIN  :INITIALIZE TEST. 21
22 23 02326 000400  400         22
23 24 02326 000400  RAND        :C(AC0)=RANDOM # 23
24 25 02327 006301  JSR @ENTRA :SOFTWARE IORI 24
25 26 02330 105700  INCS 0,1   :WITH RESULT IN AC2 25
26 27 02331 131000  MOV 1,2    :HARDWARE IORI WITH 26
27 28 02332 114000  COM 0,3    :RESULT IN AC1. 27
28 29 02332 114000  AND 3,2    28
29 30 02333 173400  AND 3,2    29
30 31 02334 173400  STA 0,+2   30
31 32 02335 040402  IORI 0,1   31
32 33 02335 107770  SUB# 1,2,SZR 32
33 34 02335 107770  ERROR      :ITERATE TEST ROUTINE 33
34 35 02340 132414  LOOP      34
35 36 02340 132414  JSR @ENTLO 35
36 37 02345 006277  36
37 38 02345 006277  37
38 39 02345 006277  38
39 40 02412 163015  ADD# 2+183,2+283,SNR 40
40 41 02413 125014  MOV# 2+383,2+383,SZR 41
41 42 02413 125014  ERROR      42
42 43 02413 125014  XORIACD 3  43
43 44 02420 102400  ZEROCAC 44
44 45 02420 102400  SUB 0,0    :CLEAR ALL AC'S 45
45 46 02421 128400  SUB 1,1    :AND C(CARRY) 46
46 47 02422 152400  SUB 2,2    47
47 48 02423 176400  SUB0 3,3   48
48 49 02424 176400  ADC 3,3    :TEST THAT XOR RESULT 49
49 50 02425 137770  XORI 0,3   :GOES ONLY TO C(ACD) 50
50 51 02425 000000  51
51 52 02427 107015  ADD# 3+183,3+283,SNR 52
52 53 02430 151014  MOV# 3+383,3+383,SZR 53
53 54 02430 151014  ERROR      54
54 55 02435 006277  LOOP      55
55 56 02435 006277  JSR @ENTLO 56
56 57 02435 006277  56
57 58 02435 006277  57
58 59 02435 006277  58
59 60 02435 006277  59
```

```

10039 ECL30
01
02
03
04 02436 006276
05 02440 102040
06 02441 040402
07 02442 123770
08 000000
09 02444 101002
10 02445 101004
11
12
13 02452 006277
14
15
16 02453 006276
17 02454 000005
18 02455 102400
19 02456 110400
20 02457 040402
21 02460 133770
22 000000
23 02462 101002
24 02463 150014
25
26
27 02470 006277
28

10040 ECL30
01
02
03
04 02471 006276
05 02472 000400
06
07 02473 006301
08 02474 105300
09 02475 131000
10 02476 113520
11 02477 107000
12 02500 146400
13 02501 111300
14 02502 040402
15 02503 133770
16 000000
17 02505 132414
18
19
20 02512 006277
21
22
23 02513 006276
24 02514 000005
25
26 02515 006301
27 02516 104700
28 02517 131300
29 02520 114000
30 02521 040402
31 02522 123770
32 000000
33 02524 044402
34 02525 127770
35 000000
36 02527 050402
37 02530 133770
38 000000
39 02532 054402
40 02533 137770
41 000000
42 02535 107015
43 02536 157014
44
45
46 02543 006277
47

EXX:
EXY:
EXZ:

:INITIALIZE TEST.
:EXCLUSIVE OR 1 TO 1
:SHOULD GIVE (0) RESULT.
:(CARRY) SHOULD REMAIN
:UNCHANGED AT (1).
:(AC0)=XORI RESULT
:ITERATE TEST ROUTINE

:INITIALIZE TEST.
:EXCLUSIVE OR 0 TO 1
:SHOULD GIVE (1) RESULT.
:(CARRY) SHOULD REMAIN
:UNCHANGED AT (1).
:ITERATE TEST ROUTINE

:INITIALIZE TEST.
:USE A RANDOM NUMBER
:(AC0)=RANDOM #
:IN EACH AC. XORI AC TO AC
:SHOULD PRODUCE 0 RESULT
:IN EACH AC.

SETUP 5
JSR @ENTIN
5
ADCO 0,0
STA 0,.*+2
XORI 0,0
MOV 0,0,SZC
MOV 0,0,SZR
ERROR
LOOP
JSR @ENTLO

SETUP 5
JSR @ENTIN
5
SUB 0,0
COMO 0,2
STA 0,.*+2
XORI 0,2
MOV 0,0,SZC
COM# 2,2,SZR
ERROR
LOOP
JSR @ENTLO

SETUP 400
JSR @ENTIN
400
RAND
JSR @ENTRA
MOV# 0,1
MOV 1,2
ANDZL 0,2
ADD 0,1
SUB 2,1
MOV# 0,2
STA 0,.*+2
XORI 0,2
SUB# 1,2,SZR
ERROR
LOOP
JSR @ENTLO

SETUP 5
JSR @ENTIN
5
RAND
JSR @ENTRA
NEG# 0,1
MOV# 1,2
COM 0,3
STA 0,.*+2
XORI 0,0
STA 1,.*+2
XORI 0,1
STA 2,.*+2
XORI 0,2
STA 3,.*+2
XORI 0,3
ADD# 0,1,SNR
ADD# 2,3,SZR
ERROR
LOOP
JSR @ENTLO

:INITIALIZE TEST.
:TEST XORI
:(CAC0)=RANDOM #
:RAND XORI RAND
:SOFTWARE SIMULATION

:(AC2)=XORI RESULT
:(CAC1)=CORRECT
:(CAC0)=ORIG OPERAND
:(AC0) SWAPED=OTHER OPERAND
:ITERATE TEST ROUTINE

:INITIALIZE TEST.
:ITERATE TEST ROUTINE

```





```

10043 ECL30
01
02
03 02706 006276
04 02707 000005
05
06 02710 006301
07 02711 105000
08 02712 150510
09 02713 040402
10 02714 133770
11 000000
12 02716 104510
13 02717 112435
14 02720 125004
15
16
17 02725 006277
18

10044 ECL30
01
02
03 02726 006276
04 02727 000100
05 02730 102420
06 02731 143770
07 02731 000000
08 02733 101002
09 02734 101004
10
11
12 02741 006277
13
14 02742 006276
15 02743 000100
16 02744 102020
17 02745 143770
18 000000
19 02747 101003
20 02750 101004
21
22
23 02755 006277
24
25 02756 006276
26 02757 000100
27 02760 102420
28 02761 143770
29 177777
30 02763 101002
31 02764 101004
32
33
34 02771 006277
35
36 02772 006276
37 02775 000100
38 02774 102520
39 02775 143770
40 000001
41 02777 126520
42 03000 106434
43
44
45 03005 006277
46
47 03006 006276
48 03007 000100
49 03010 020027-
50 03011 143770
51 052525
52 03013 101004
53
54
55 03020 006277

!TEST "XORI"
!INITIALIZE TEST.
:INITIALIZE TEST.
:SET CARRY TO 1, ACO = 0
:AND OF 0 AND 0 SHOULD
:RE 0, SHUD NOT EFFECT CARRY

!ITERATE TEST ROUTINE
:ITERATE TEST ROUTINE
:INITIALIZE TEST.
:ANDI OF 1 & 0 SHOULD RE
:0 - CARRY = 0 SHOULD NOT
:BE AFFECTED

!ITERATE TEST ROUTINE
:ITERATE TEST ROUTINE
:INITIALIZE TEST.
:AND OF 0 AND 1 SHOULD BE 0

!ITERATE TEST ROUTINE
:ITERATE TEST ROUTINE
:INITIALIZE TEST.
:AND OF 1 & 1 SHUD RE 1

!ITERATE TEST ROUTINE
:ITERATE TEST ROUTINE
:SET EVERY OTHER BIT
:INITIALIZE TEST.
:AND WITH COMPLIMENT RESULT
:SHOULD BE ZERO

!ITERATE TEST ROUTINE
:ITERATE TEST ROUTINE

```



```

10047 ECL30
01
02
03 03207 006276          Ew03:
04 03210 000100          JSR #ENTIN
05 03211 176420          SUBZ 3,3
06 03212 157770          ANDI 0,3
07
08 03214 175002          MOV 3,3,SZR
09 03215 175004          ERROR
10 LOOP
11
12 03222 006277          JSR #ENTLO
13          SETUP 100
14 03223 006276          JSR #ENTIN
15 03224 000100          LDA 0,-125252
16 03225 176020          ANDI 0,1
17 03226 157770          ANDI 125252,0
18          SUBZ# 0,1,SZR
19 03230 175003          ERROR
20 03231 175004          ERROR
21 LOOP
22
23 03236 006277          JSR #ENTLO
24          SETUP 100
25 03237 006276          JSR #ENTIN
26 03240 000100          LDA 0,-125252
27 03241 176420          ANDI 0,1
28 03242 157770          ANDI 125252,0
29          SUBZ# 0,1,SZR
30 03244 175002          ERROR
31 03245 175004          ERROR
32 LOOP
33
34 03252 006277          JSR #ENTLO
35          SETUP 100
36 03253 006276          JSR #ENTIN
37 03254 000100          LDA 0,-125252
38 03255 176520          ANDI 0,1
39 03256 157770          ANDI 125252,0
40          SUBZ# 0,1,SZR
41 03260 102520          ERROR
42 03261 116434          ERROR
43 LOOP
44
45 03266 006277          JSR #ENTLO
46          SETUP 100
47 03267 006276          JSR #ENTIN
48 03270 000100          LDA 0,-125252
49 03271 034027          ANDI 0,1
50 03272 157770          ANDI 125252,0
51          SUBZ# 0,1,SZR
52 03274 175004          ERROR
53 03275 175004          ERROR
54 LOOP
55 03301 006277          JSR #ENTLO
56          SETUP 100
57 03302 006276          JSR #ENTIN
58 03303 000100          LDA 0,-125252
59 03304 020027          ANDI 0,1
60 03305 105000          ANDI 125252,0
61          SUBZ# 0,1,SZR
62 03310 106434          ERROR
63 LOOP
64
65 03315 006277          JSR #ENTLO
66          SETUP 100
67 03316 006276          JSR #ENTIN
68 03317 000100          LDA 0,-125252
69 03318 020026          ANDI 0,1
70 03319 105000          ANDI 125252,0
71          SUBZ# 0,1,SZR
72 03324 106434          ERROR
73 LOOP
74
75 03331 006277          JSR #ENTLO
76          SETUP 100
77 03332 006276          JSR #ENTIN
78 03333 000100          LDA 0,-125252
79 03334 102020          ANDI 0,0
80 03335 143770          ANDI -1,0
81          SUBZ# 0,0,SZR
82 03337 101003          ERROR
83 03340 100014          ERROR
84 LOOP
85
86 03345 006277          JSR #ENTLO
87          SUBZL 1,1
88 03346 126520          SETUP 100
89 03347 006276          JSR #ENTIN
90 03350 000100          LDA 0,-125252
91 03351 121000          ANDI 0,1
92 03352 143770          ANDI -1,0
93          SUBZ# 0,1,SZR
94 03354 106434          ERROR
95 03356 177777          ERROR
96 LOOP
97
98 03361 006277          JSR #ENTLO
99          MOVZL 1,1,SZR
100 03362 125124          JMP EX10
101
10048 ECL30
EX07:
01
02 03302 006276          JSR #ENTIN
03 03303 000100          LDA 0,-125252
04 03304 020027          ANDI 0,1
05 03305 105000          ANDI 125252,0
06          SUBZ# 0,1,SZR
07 03310 106434          ERROR
08 03311 106434          ERROR
09 LOOP
10
11 03315 006277          JSR #ENTLO
12          SETUP 100
13 03316 006276          JSR #ENTIN
14 03317 000100          LDA 0,-125252
15 03318 020026          ANDI 0,1
16 03319 105000          ANDI 125252,0
17          SUBZ# 0,1,SZR
18 03324 106434          ERROR
19 03325 052525          ERROR
20 03324 106434          ERROR
21 LOOP
22
23 03331 006277          JSR #ENTLO
24          SETUP 100
25 03332 006276          JSR #ENTIN
26 03333 000100          LDA 0,-125252
27 03334 102020          ANDI 0,0
28 03335 143770          ANDI -1,0
29          SUBZ# 0,0,SZR
30 03337 101003          ERROR
31 03340 100014          ERROR
32 LOOP
33
34 03345 006277          JSR #ENTLO
35          SUBZL 1,1
36 03346 126520          SETUP 100
37 03347 006276          JSR #ENTIN
38 03350 000100          LDA 0,-125252
39 03351 121000          ANDI 0,1
40 03352 143770          ANDI -1,0
41          SUBZ# 0,1,SZR
42 03354 106434          ERROR
43 03356 177777          ERROR
44 LOOP
45
46 03361 006277          JSR #ENTLO
47          MOVZL 1,1,SZR
48 03362 125124          JMP EX10
49
EX08:
01
02 03315 006277          JSR #ENTLO
03          SETUP 100
04 03316 006276          JSR #ENTIN
05 03317 000100          LDA 0,-125252
06 03318 020026          ANDI 0,1
07 03319 105000          ANDI 125252,0
08          SUBZ# 0,1,SZR
09 03324 106434          ERROR
10 03325 052525          ERROR
11 03324 106434          ERROR
12 LOOP
13
14 03331 006277          JSR #ENTLO
15          SETUP 100
16 03332 006276          JSR #ENTIN
17 03333 000100          LDA 0,-125252
18 03334 102020          ANDI 0,0
19 03335 143770          ANDI -1,0
20          SUBZ# 0,0,SZR
21 03337 101003          ERROR
22 03340 100014          ERROR
23 LOOP
24
25 03345 006277          JSR #ENTLO
26          SUBZL 1,1
27 03346 126520          SETUP 100
28 03347 006276          JSR #ENTIN
29 03350 000100          LDA 0,-125252
30 03351 121000          ANDI 0,1
31 03352 143770          ANDI -1,0
32          SUBZ# 0,1,SZR
33 03354 106434          ERROR
34 03356 177777          ERROR
35 LOOP
36
37 03361 006277          JSR #ENTLO
38          MOVZL 1,1,SZR
39 03362 125124          JMP EX10
40
EX09:
01
02 03331 006277          JSR #ENTLO
03          SETUP 100
04 03332 006276          JSR #ENTIN
05 03333 000100          LDA 0,-125252
06 03334 102020          ANDI 0,0
07 03335 143770          ANDI -1,0
08          SUBZ# 0,0,SZR
09 03337 101003          ERROR
10 03340 100014          ERROR
11 LOOP
12
13 03345 006277          JSR #ENTLO
14          SUBZL 1,1
15 03346 126520          SETUP 100
16 03347 006276          JSR #ENTIN
17 03350 000100          LDA 0,-125252
18 03351 121000          ANDI 0,1
19 03352 143770          ANDI -1,0
20          SUBZ# 0,1,SZR
21 03354 106434          ERROR
22 03356 177777          ERROR
23 LOOP
24
25 03361 006277          JSR #ENTLO
26          MOVZL 1,1,SZR
27 03362 125124          JMP EX10
28
EX10:
01
02 03345 006277          JSR #ENTLO
03          SUBZL 1,1
04 03346 126520          SETUP 100
05 03347 006276          JSR #ENTIN
06 03350 000100          LDA 0,-125252
07 03351 121000          ANDI 0,1
08 03352 143770          ANDI -1,0
09          SUBZ# 0,1,SZR
10 03354 106434          ERROR
11 03356 177777          ERROR
12 LOOP
13
14 03361 006277          JSR #ENTLO
15          MOVZL 1,1,SZR
16 03362 125124          JMP EX10
17

```

```

10049 ECL30
01 SETUP 100
02 JSR @ENTLO
100
03 RAND
04 JSR @ENTRA
05 MOV 0,1
06 STA 1,.*2
07 STA 1,.*2
08 STA 1,.*2
09 STA 1,.*2
10 SUB# 0,1,SZR
11 ERROR
12 LOOP
13 JSR @ENTLO
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

EX104:
:INITIALIZE TEST.
:AND OF A NUMBER WITH ITSELF
:R(CAO)=RANDOM #
:SHUD NOT CHANGE THE NUMBER
:RANDOM INPUT
:AC0) ANDI RESULT
:AC1) ORIGINAL NUMBER
:ITERATE TEST ROUTINE

EX11:
SETUP 100
JSR @ENTIN
100
ANDIACD 0
ZERDAC
SUB 0,0
SUB 1,1
SUB 2,2
SUBO 3,3
ADC 0,0
STA 0,.*2
ANDI 0,0
ADD# 0+1&3,0+2&3,SNR
MOV# 0+3&3,0+3&3,SZR
ERROR
LOOP
JSR @ENTLO
SETUP 100
JSR @ENTIN
100
ANDIACD 0
ZERDAC
SUB 0,0
SUB 1,1
SUB 2,2
SUBO 3,3
ADC 0,0
STA 0,.*2
ANDI 0,0
ADD# 0+1&3,0+2&3,SNR
MOV# 0+3&3,0+3&3,SZR
ERROR
LOOP
JSR @ENTLO
SETUP 100
JSR @ENTIN
100
ANDIACD 1
ZERDAC
SUB 0,0
SUB 1,1
SUB 2,2
SUBO 3,3
ADC 1,1
STA 1,.*2
ANDI 0,1
ADD# 1+1&3,1+2&3,SNR
MOV# 1+3&3,1+3&3,SZR
ERROR
LOOP
JSR @ENTLO

EX104:
SETUP 100
JSR @ENTIN
100
ANDIACD 2
ZERDAC
SUB 0,0
SUB 1,1
SUB 2,2
SUBO 3,3
ADC 2,2
STA 2,.*2
ANDI 0,2
ADD# 2+1&3,2+2&3,SNR
MOV# 2+3&3,2+3&3,SZR
ERROR
LOOP
JSR @ENTLO
SETUP 100
JSR @ENTIN
100
ANDIACD 3
ZERDAC
SUB 0,0
SUB 1,1
SUB 2,2
SUBO 3,3
ADC 3,3
STA 3,.*2
ANDI 0,3
ADD# 3+1&3,3+2&3,SNR
MOV# 3+3&3,3+3&3,SZR
ERROR
LOOP
JSR @ENTLO

EX13:
10050 ECL30
01 SETUP 100
02 JSR @ENTIN
100
03 ANDIACD 2
04 ZERDAC
05 SUB 0,0
06 SUB 1,1
07 SUB 2,2
08 SUBO 3,3
09 ADC 2,2
10 STA 2,.*2
11 ANDI 0,2
12 ADD# 2+1&3,2+2&3,SNR
13 MOV# 2+3&3,2+3&3,SZR
14 ERROR
15 LOOP
16 JSR @ENTLO
17 SETUP 100
18 JSR @ENTIN
19 ANDIACD 3
20 ZERDAC
21 SUB 0,0
22 SUB 1,1
23 SUB 2,2
24 SUBO 3,3
25 ADC 3,3
26 STA 3,.*2
27 ANDI 0,3
28 ADD# 3+1&3,3+2&3,SNR
29 MOV# 3+3&3,3+3&3,SZR
30 ERROR
31 LOOP
32 JSR @ENTLO
33 SETUP 100
34 JSR @ENTIN
35 ANDIACD 3
36 ZERDAC
37 SUB 0,0
38 SUB 1,1
39 SUB 2,2
40 SUBO 3,3
41 ADC 3,3
42 STA 3,.*2
43 ANDI 0,3
44 ADD# 3+1&3,3+2&3,SNR
45 MOV# 3+3&3,3+3&3,SZR
46 ERROR
47 LOOP
48 JSR @ENTLO
49 SETUP 100
50 JSR @ENTIN
51 ANDIACD 3
52 ZERDAC
53 SUB 0,0
54 SUB 1,1
55 SUB 2,2
56 SUBO 3,3
57 ADC 3,3
58 STA 3,.*2
59 ANDI 0,3
60 ADD# 3+1&3,3+2&3,SNR
61 MOV# 3+3&3,3+3&3,SZR
62 ERROR
63 LOOP
64 JSR @ENTLO

EX14:
18 03463 006277
20 03464 006276
21 03465 000100
24 03466 102400
25 03467 126400
26 03470 152400
27 03471 176400
28 03472 176000
29 03473 054402
30 03474 157770
31 000000
32 03076 107015
33 03077 151014
36 03504 006277

```



10053 ECL30

```
01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
E114A: SETUP 100 JSR @ENTLO  
03 03663 006276 JSR @ENTIN  
04 03664 000100 LOOP  
SUBZ 2,2  
05 03665 15420 HLV 2  
06 03666 153370 MOV 2,2,SZC  
07 03667 151002 MOV 2,2,SZR  
08 03670 151004 ERROR  
09  
10  
11 03675 006277 JSR @ENTLO  
12 SETUP 100  
13 03676 006276 JSR @ENTIN  
14 03677 000100 LOOP  
SUBZL 2,2  
15 03700 152520 HLV 2  
16 03701 153370 MOV# 2,2,SNC  
17 03702 151013 MOV 2,2,SZR  
18 03703 151004 ERROR  
19  
20  
21 03710 006277 JSR @ENTLO  
22 SETUP 100  
23 03711 006276 JSR @ENTIN  
24 03712 000100 LOOP  
25 03713 030025- LDA 2,2  
26 03714 153370 HLV 2  
27 03715 151235 MOVZRH 2,2,SNR  
28 03716 151235 MOVZRN 2,2,SNC  
29 ERROR  
30  
31 03723 006277 JSR @ENTLO  
32 SETUP 100  
33 03724 006276 JSR @ENTIN  
34 03725 000100 LDA 2,2  
35 03726 030024- HLV 2  
36 03727 153370 COM# 2,2,SZR  
37 03730 150014 ERROR  
38  
39  
40 03735 006277 JSR @ENTLO  
41 SETUP 100  
42 03736 006276 JSR @ENTIN  
43 03737 000100 LDA 2,2  
44 03740 030047- LDA 0,2-4  
45 03741 020023- HLV 2  
46 03742 153370 HLV 0  
47 03743 143370 ADD# 2,0,SZR  
48 03744 143014 ERROR  
49  
50  
51 03751 006277 JSR @ENTLO
```

10054 ECL30

```
01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
E114A: SETUP 100 JSR @ENTLO  
03 03752 006276 JSR @ENTIN  
100  
SUBZ 3,3  
05 03754 17420 HLV 3  
06 03755 157370 MOV 3,3,SZC  
07 03756 175002 MOV 3,3,SZR  
08 03757 175004 ERROR  
09  
10  
11 03760 006277 JSR @ENTLO  
12 SETUP 100  
13 03765 006276 JSR @ENTIN  
14 03766 000100 LOOP  
SUBZL 3,3  
15 03767 175520 HLV 3  
16 03770 157370 MOV# 3,3,SNC  
17 03771 175013 MOV 3,3,SZR  
18 03772 175004 ERROR  
19  
20  
21 03777 006277 JSR @ENTLO  
22 SETUP 100  
23 04000 006276 JSR @ENTIN  
24 04001 000100 LDA 3,2  
25 04002 030025- HLV 3  
26 04003 157370 MOVZRH 3,3,SNR  
27 04004 175235 MOVZRN 3,3,SNC  
28 04005 175235 ERROR  
29  
30  
31 04012 006277 JSR @ENTLO  
32 SETUP 100  
33 04013 006276 JSR @ENTIN  
34 04014 000100 LDA 3,-2  
35 04015 030024- HLV 3  
36 04016 157370 COM# 3,3,SZR  
37 04017 174014 ERROR  
38  
39  
40 04024 006277 JSR @ENTLO  
41 SETUP 100  
42 04025 006276 JSR @ENTIN  
43 04026 000100 LDA 3,-4  
44 04027 030047- LDA 0,-4  
45 04030 020023- HLV 3  
46 04031 157370 HLV 0  
47 04032 143370 ADD# 3,0,SZR  
48 04033 163014 ERROR  
49  
50  
51 04040 006277 JSR @ENTLO
```

;NOW WE TRY AC3  
;INITIALIZE TEST.

;HALF OF 0 = 0  
;CARRY SHUD NOT CHANGE

;ITERATE TEST ROUTINE  
;INITIALIZE TEST.

;ITERATE TEST ROUTINE  
;INITIALIZE TEST.

;ITERATE TEST ROUTINE  
;INITIALIZE TEST.

;ITERATE TEST ROUTINE  
;INITIALIZE TEST.

;SEE IF AC2 WORKS 100  
;INITIALIZE TEST.

;HALF OF 0 = 0  
;CARRY SHUD NOT CHANGE

;ITERATE TEST ROUTINE  
;INITIALIZE TEST.

;ITERATE TEST ROUTINE  
;INITIALIZE TEST.

;ITERATE TEST ROUTINE  
;INITIALIZE TEST.

;ITERATE TEST ROUTINE  
;INITIALIZE TEST.

```

10055 ECL30
01
02 04041 126520 EX19: SUBRZL 1,1
03 04042 006276 :TRY ALL ONE BIT NUMBERS
04 04043 000100 :INITIALIZE TEST.
05 04044 121120 JSR @ENTIN
06 04045 143370 MOVZL 1,0
07 04046 106434 :AC1)*2 TO AC0
08 04047 106434 HLV 0
09 04048 106434 SUBZ # 0,1,SZR
10 04049 106434 :AC1 SOFTWARE RESULT
11 04050 006277 LOOP
12 04051 125120 JSR @ENTLO
13 04052 127134 MOVZL 1,1
14 04053 000764 ADDZL# 1,1,SZR
15 04054 000764 JMP EX19
16
17 04057 006276 EX20: SETUP 100
18 04058 000100 JSR @ENTIN
19 RAND
20 04061 006301 JSR @ENTRA
21 04062 105000 MOV 0,1
22 04063 101133 MOVZL # 0,0,SNC
23 04064 000405 JMP .+5
24 04065 100400 NEG 0,0
25 04066 101220 MOVZR 0,0
26 04067 100400 NEG 0,0
27 04070 000402 JMP .+2
28 04071 101220 MOVZR 0,0
29 04072 147370 HLV 1
30 04073 106434 SUBZ# 0,1,SZR
31 ERROR
32 LOOP
33 04100 006277 JSR @ENTLO
34
10056 ECL30
01
02 04101 006276 EY20: SETUP 100
03 04102 000100 JSR @ENTIN
04 RAND
05 04103 006301 JSR @ENTRA
06 04104 111000 MOV 0,2
07 04105 101133 MOVZL # 0,0,SNC
08 04106 000405 JMP .+5
09 04107 100400 NEG 0,0
10 04110 101220 MOVZR 0,0
11 04111 100400 NEG 0,0
12 04112 000402 JMP .+2
13 04113 101220 MOVZR 0,0
14 04114 153370 HLV 2
15 04115 112434 SUBZ# 0,2,SZR
16 ERROR
17 LOOP
18 04122 006277 JSR @ENTLO
19
20 04123 006276 EY20: SETUP 100
21 04124 000100 JSR @ENTIN
22 RAND
23 04125 006301 JSR @ENTRA
24 04126 115000 MOV 0,3
25 04127 101133 MOVZL # 0,0,SNC
26 04130 000405 JMP .+5
27 04131 100400 NEG 0,0
28 04132 101220 MOVZR 0,0
29 04133 100400 NEG 0,0
30 04134 000402 JMP .+2
31 04135 101220 MOVZR 0,0
32 04136 157370 HLV 3
33 04137 116434 SUBZ# 0,3,SZR
34 ERROR
35 LOOP
36 04144 006277 JSR @ENTLO
37

```



```

10057 ECL30
01
02
03
04 04145 020022-
05 04146 040040
06 04147 020021-
07 04150 020042 X01:
08 04151 006276
09 04152 001100
10 04153 020040
11 04154 102440
12 04155 103370
13 04156 030040
14 04157 132434
15 04173 132014
16
17
18 04164 006277
19
20
21 04165 006276 X02:
22 04166 000100
23 04167 024040
24 04170 102520
25 04171 103370
26 04172 030040
27 04173 132014
28
29 04200 014040
30 04201 00401
31
32 04202 006277
33
34
35 04203 006276 X03:
36 04204 000100
37 04205 020022-
38 04206 040041
39 04207 126620
40 04210 123000
41 04211 040040
42 04212 163770
43
44 04214 040042
45 04215 020020-
46 04216 004401
47 04217 103370 X03A:
48 04220 101400
49 04221 006004
50 04222 024254
51 04223 171000
52 04224 132434
53
54
55 04231 006277
56

```

```

10058 ECL30
01
02 04232 000402
03 04233 000000 X04:
04
05 04234 006276
06 04235 000100
07 04236 020022-
08 04237 040041
09 04240 040773
10 04241 126620
11 04242 123000
12 04243 040040
13 04244 163770
14
15 04246 040042
16 04247 020020-
17 04250 103370
18 04251 101400
19 04252 000404
20 04253 020040
21 04254 020757
22 04255 106454
23
24
25 04262 006277

```

```

      JMP *+2
      SETUP 100
      JSR @ENTLN
      LDA 0,=420
      STA 0,SP
      LDA 0,=77777
      STA 0,USL
      SETUP 100
      JSR @ENTLN
      LDA 1,SP
      SUBO 0,0
      LDA 2,SP
      SUBZ# 1,2,SZR
      ERROR
      JSR @ENTLN
      SETUP 100
      JSR @ENTLN
      LDA 1,SP
      SUBZL 0,0
      MSP 0
      LDA 2,SP
      ADC# 1,2,SZR
      ERROR
      DSZ SP
      JMP *+1
      LOOP *+1
      JSR @ENTLN
      SETUP 100
      JSR @ENTLN
      LDA 0,=420
      STA 0,SF
      SUBZR 1,1
      ADD 1,0
      STA 0,SP
      ADDI -10,0
      STA 0,USL
      LDA 0,=100
      MSP 0
      INC 0,0
      JMP *+4
      LDA 0,SP
      LDA 1,X04-1
      SUBZ# 0,1,SZR
      ERROR
      JSR @ENTLN

```

```

      :TRY AGAIN AND SEE IF SP IS
      :INITIALIZE TEST.
      :RETURNED TO THE PROPER VALUE
      :ITERATE TEST ROUTINE

```

```

      :**** MSP *****
      LDA 0,=420
      STA 0,SP
      LDA 0,=77777
      STA 0,USL
      :ADDING 0 TO SP SHOULD NOT
      :INITIALIZE TEST.
      :CHANGE ITS VALUE
      :AC1 =ORIG VALUE, AC2 =RESULTANT
      :ITERATE TEST ROUTINE
      :TRY IT ADDING ONE TO SP
      :INITIALIZE TEST.
      :AC1 =ORIG, AC2 = NEW VALUE
      :ITERATE TEST ROUTINE
      :CAUSE AN OVERFLOW
      :INITIALIZE TEST.

```

```

      :SET STACK LIMIT SO OVFL OCCURS
      :SEE IF PC PUSHED ON STACK IS
      :THAT OF MSP INSTR, AND THAT
      :OVERFLOW DID OCCUR
      :SHD =PC OF MSP
      :IF NOT EQUAL WRONG PC
      :IF AC0 = 101, NO OVERFLOW
      :ITERATE TEST ROUTINE

```

```

1059 ECL30
01
02 04263 020022-
03 04264 040040
04 04265 020021-
05 04266 040042  XA01:
06 SETUP 100
07 04267 006276
08 04270 000100
09 04271 020040
10 04272 126440
11 04273 107370
12 04274 030040
13 04275 112434
14
15
16 04302 006277
17
18
19 04303 006276  XA02:
20 04304 000100
21 04305 020040
22 04306 126520
23 04307 107370
24 04310 030040
25 04311 112014
26
27 04316 014040
28 04317 000401
29
30 04320 006277
31
32
33 04321 006276  XA03:
34 04322 000100
35 04323 024022-
36 04324 040041
37 04325 102620
38 04326 107000
39 04327 040040
40 04330 167770
41
42 04332 040042
43 04333 024020-
44 04334 101040
45 04335 004001
46 04336 107370  XA03A:
47 04337 125400
48 04340 000405
49 04341 020254
50 04342 126560
51 04343 173000
52 04344 112434
53
54
55 04351 006277

LDA 0,=420
STA 0,SP
LDA 0,=77777
STA 0,USL
SETUP 100
JSR @ENTIN
LOOP
LDA 0,SP
SUB0 1,1
MSP 1
LDA 2,SP
SUBZ# 0,2,SZR
ERROR
DSZ SP
JMP *+1
LOOP
JSR @ENTLO

:REPEAT THESE TESTS WITH AC1
:INITIALIZE TEST.
:ITERATE TEST ROUTINE
:INITIALIZE TEST.
:ITERATE TEST ROUTINE
:INITIALIZE TEST.
:ITERATE TEST ROUTINE
:ALSO C=1 0M STK
:ITERATE TEST ROUTINE

```

```

1060 ECL30
01
02 04352 000402
03 04353 000000  XA04:
04 SETUP 100
05 04354 006276
06 04355 000100
07 04356 020022-
08 04357 040041
09 04360 040773
10 04361 126620
11 04362 123000
12 04363 040040
13 04364 163770
14
15 04366 040042
16 04367 024020-
17 04370 107370
18 04371 125400
19 04372 000404
20 04373 020040
21 04374 030757
22 04375 112434
23
24
25 04402 006277

JSR @ENTIN
LDA 0,=420
STA 0,SP
SUBZ 1,1
ADD 1,0
STA 0,SP
ADDI -10,0
LOOP
STA 0,USL
LDA 1,=100
MSP 1
INC 1,1
JMP *+4
LDA 0,SP
SUBZ# 0,2,SZR
ERROR
LOOP
JSR @ENTLO

:INITIALIZE TEST.
:ITERATE TEST ROUTINE

```

```

10061 ECL30
01
02 04403 020022-
03 04404 040040
04 04405 020021-
05 04406 040042
06
07 04407 066276
08 04410 000100
09 04411 020040
10 04412 152440
11 04413 113370
12 04414 024040
13 04415 106434
14
15
16 04422 066277
17
18
19 04423 066276
20 04424 000100
21 04425 020040
22 04426 152520
23 04427 113370
24 04430 024040
25 04431 106014
26
27 04436 014040
28 04437 000401
29
30 04440 066277
31
32
33 04441 066276
34 04442 000100
35 04443 030022-
36 04444 050041
37 04445 126620
38 04446 133000
39 04447 050040
40 04450 177770
41
42 04452 050042
43 04453 030020-
44 04454 004001
45 04455 113370
46 04456 151400
47 04457 000404
48 04460 024254
49 04461 161000
50 04462 122434
51
52
53 04467 066277

```

```

LDA 0,=420
STA 0,SP
LDA 0,=17777
STA 0,USL
SETUP 100
JSR @ENTLO
LDA 0,SP
SUBO 2,2
MSP 2
LDA 1,SP
SUBZ# 0,1,SZR
ERRR
LOOP
JSR @ENTLO
ITERATE TEST ROUTINE
INITIALIZE TEST.
SETUP 100
JSR @ENTIN
LDA 0,SP
SUBZL 2,2
MSP 2
LDA 1,SP
ADC# 0,1,SZR
ERRR
DSZ SP
JMP .+1
LOOP
JSR @ENTLO
ITERATE TEST ROUTINE
INITIALIZE TEST.
SETUP 100
JSR @ENTIN
LDA 2,=420
STA 2,SP
SUBZR 1,1
ADD 1,2
STA 2,SP
ADDI -10,2
STA 2,USL
LDA 2,=100
JSR 2,+1
MSP 2
INC 2,2
JMP .+4
LDA 1,SVPC
MOV 3,0
SUBZ# 1,0,SZR
ERRR
LOOP
JSR @ENTLO
ITERATE TEST ROUTINE

```

```

10062 ECL30
01
02 04470 000402
03 04471 000000
04
05 04472 066276
06 04473 000100
07 04474 020022-
08 04475 040041
09 04476 040773
10 04477 126620
11 04500 123000
12 04501 040040
13 04502 163770
14
15 04504 040042
16 04505 030020-
17 04506 113370
18 04507 151400
19 04510 000404
20 04511 020040
21 04512 024757
22 04513 106434
23
24
25 04520 066277

```

```

LDA 0,=420
STA 0,SP
LDA 0,=17777
STA 0,USL
SETUP 100
JSR @ENTIN
LDA 0,=420
STA 0,SP
SUBZR 1,1
ADD 1,0
STA 0,SP
ADDI -10,0
LOOP
JSR @ENTLO
ITERATE TEST ROUTINE
INITIALIZE TEST.
SETUP 100
JSR @ENTIN
LDA 0,=420
STA 0,SP
SUBZR 1,1
ADD 1,0
STA 0,SP
ADDI -10,0
LOOP
JSR @ENTLO
ITERATE TEST ROUTINE

```

```

1063 ECL30
01
02 04521 020022-
03 04522 040040
04 04523 020021-
05 04524 040042
06
07 04525 006276
08 04526 000100
09 04527 020040
10 04528 176440
11 04529 117370
12 04530 030040
13 04531 112434
14
15
16 04540 006277
17
18 04541 006276
19 04542 000100
20 04543 020040
21 04544 176520
22 04545 117370
23 04546 024040
24 04547 111400
25 04550 132434
26
27 04555 014040
28 04556 000401
29
30 04557 006277
31
32 04560 006276
33 04561 000100
34 04562 004401
35 04563 165000
36 04564 167770
37
38 04566 034022-
39 04567 034041
40 04570 152620
41 04571 157000
42 04572 054040
43 04573 177770
44
45 04575 054042
46 04576 034020-
47 04577 117370
48 04600 175400
49 04601 009403
50 04602 030254
51 04603 132434
52
53
54 04610 006277

10064 ECL30
01
02 04611 000402
03 04612 000000
04
05 04613 006276
06 04614 000100
07 04615 020022-
08 04616 040041
09 04617 040773
10 04620 126620
11 04621 123000
12 04622 040040
13 04623 163770
14
15 04625 040042
16 04626 034020-
17 04627 117370
18 04630 175400
19 04631 000404
20 04632 020040
21 04633 024757
22 04634 106434
23
24
25 04641 006277

:AGAIN USING AC3
:INITIALIZE TEST.
XCO1:
LDA 0,=420
STA 0,SP
LDA 0,=77777
STA 0,USL
SETUP 100
JSR @ENTIN
LDA 0,SP
SUBR 3,3
MSP 3
LDA 2,SP
SUBZ# 0,2,SZR
ERROR
LOOP
JSR @ENTLO
SETUP 100
JSR @ENTIN
LDA 0,SP
SUBR 3,3
MSP 3
LDA 1,SP
INC 0,2
SUBZ# 1,2,SZR
ERROR
DSZ SP
JMP *+1
LOOP
JSR @ENTLO
SETUP 100
JSR @ENTIN
LDA 3,=420
STA 3,SP
SUBR 2,2
ADD 2,3
STA 3,SP
ADDI -10,3
LDA 3,USL
LDA 3,=100
MSP 3
INC 3,3
JMP *+3
LDA 2,SVPC
SUBZ# 1,2,SZR
ERROR
LOOP
JSR @ENTLO

:ITERATE TEST ROUTINE
:INITIALIZE TEST.
XCO2:
LDA 0,=420
STA 0,SP
LDA 0,=77777
STA 0,USL
SETUP 100
JSR @ENTIN
LDA 0,=420
STA 0,SP
SUBR 1,1
ADD 1,0
ADDI -10,0
STA 0,USL
LDA 3,=100
MSP 3
INC 3,3
JMP *+4
LDA 0,SP
LDA 1,XCO4-1
SUBZ# 0,1,SZR
ERROR
LOOP
JSR @ENTLO

:ITERATE TEST ROUTINE
:INITIALIZE TEST.
XCO3:
LDA 3,=420
STA 3,SP
SUBR 2,2
ADD 2,3
STA 3,SP
ADDI -10,3
LDA 3,USL
LDA 3,=100
MSP 3
INC 3,3
JMP *+3
LDA 2,SVPC
SUBZ# 1,2,SZR
ERROR
LOOP
JSR @ENTLO

:ITERATE TEST ROUTINE
:INITIALIZE TEST.
XCO4:
LDA 0,=420
STA 0,SP
LDA 0,=77777
STA 0,USL
SETUP 100
JSR @ENTIN
LDA 0,=420
STA 0,SP
SUBR 1,1
ADD 1,0
ADDI -10,0
STA 0,USL
LDA 3,=100
MSP 3
INC 3,3
JMP *+4
LDA 0,SP
LDA 1,XCO4-1
SUBZ# 0,1,SZR
ERROR
LOOP
JSR @ENTLO

:ITERATE TEST ROUTINE
:INITIALIZE TEST.
XCO5:
LDA 3,=420
STA 3,SP
SUBR 2,2
ADD 2,3
STA 3,SP
ADDI -10,3
LDA 3,USL
LDA 3,=100
MSP 3
INC 3,3
JMP *+3
LDA 2,SVPC
SUBZ# 1,2,SZR
ERROR
LOOP
JSR @ENTLO

:ITERATE TEST ROUTINE
:INITIALIZE TEST.
XCO6:
LDA 0,=420
STA 0,SP
LDA 0,=77777
STA 0,USL
SETUP 100
JSR @ENTIN
LDA 0,=420
STA 0,SP
SUBR 1,1
ADD 1,0
ADDI -10,0
STA 0,USL
LDA 3,=100
MSP 3
INC 3,3
JMP *+4
LDA 0,SP
LDA 1,XCO6-1
SUBZ# 0,1,SZR
ERROR
LOOP
JSR @ENTLO

```

10065 ECL30

```
01
02
03
04 04642 102440
05 04643 030242
06 04644 024031-
07 04645 041000
08 04646 151400
09 04647 125404
10 04650 000775
11 04651 020242
12 04652 126440
13
14 04653 006276
15 04654 000010
16 04655 106770
17 04656 000402
18
19
20 04663 006277
21 04664 125400
22 04665 030020-
23 04666 132434
24 04667 000764
25
26
27 04670 102000
28 04671 030242
29 04672 024031-
30 04673 041000
31 04674 151400
32 04675 125404
33 04676 000775
34 04677 020242
35 04700 126440
36
37 04701 006276
38 04702 000010
39 04703 106770
40
41
42 04710 006277
43 04711 125400
44 04712 030020-
45 04713 132434
46 04714 000765
```

:\*\*\*\*\* SNB \*\*\*\*\*

```
SUBO 0,0
LDA 2,ARBUF
LDA 1,=-10
STA 0,0,2
INC 2,2
INC 1,1,SZR
JMP *-3
LDA 0,ARBUF
SUBO 1,1
SETUP 10
JSR BENTIN
SNB 0,1
INC 1,1
JMP *-2
ERROR
LOOP
JSR BENTLO
INC 1,1
LDA 2,=-100
SUBZ# 1,2,SZR
JMP A01

ADC 0,0
LDA 2,ARBUF
LDA 1,=-10
STA 0,0,2
INC 2,2
INC 1,1,SZR
JMP *-3
LDA 0,ARBUF
SUBO 1,1
SETUP 10
JSR BENTIN
SNB 0,1
ERROR
LOOP
JSR BENTLO
INC 1,1
LDA 2,=-100
SUBZ# 1,2,SZR
JMP A02

:***** SNB *****
: FILL BUFFER WITH ZEROS
SUBO 0,0
LDA 2,ARBUF
LDA 1,=-10
STA 0,0,2
INC 2,2
INC 1,1,SZR
JMP *-3
LDA 0,ARBUF
SUBO 1,1
SETUP 10
JSR BENTIN
SNB 0,1
INC 1,1
JMP *-2
ERROR
LOOP
JSR BENTLO
INC 1,1
LDA 2,=-100
SUBZ# 1,2,SZR
JMP A03

: FILL WITH ALL ONE'S
SUBO 0,0
LDA 2,ARBUF
LDA 1,=-10
STA 0,0,2
INC 2,2
INC 1,1,SZR
JMP *-3
LDA 0,ARBUF
SUBO 1,1
SETUP 10
JSR BENTIN
SNB 0,1
ERROR
LOOP
JSR BENTLO
INC 1,1
LDA 2,=-100
SUBZ# 1,2,SZR
JMP A04

: SKIP ON NON-ZERO BIT SHOULD ALWAYS SKIP
: INITIALIZE TEST.
SUBO 1,1
SETUP 10
JSR BENTIN
SNB 0,1
INC 1,1
JMP *-2
ERROR
LOOP
JSR BENTLO
INC 1,1
LDA 2,=-100
SUBZ# 1,2,SZR
JMP A05

: ITERATE TEST ROUTINE
: DONE THE WHOLE TABLE?
SUBO 1,1
SETUP 10
JSR BENTIN
SNB 0,1
INC 1,1
JMP *-2
ERROR
LOOP
JSR BENTLO
INC 1,1
LDA 2,=-100
SUBZ# 1,2,SZR
JMP A06
```

10066 ECL30

```
01
02 04715 020027-
03 04716 030242
04 04717 024031-
05 04720 041000
06 04721 151400
07 04722 125404
08 04723 000775
09 04724 020242
10 04725 126440
11
12 04726 006276
13 04727 000020
14 04730 106770
15
16 04735 125400
17 04736 106770
18 04737 000402
19
20 04744 125400
21
22 04745 006277
23 04746 030020-
24 04747 132434
25 04750 000756
26
27 04751 020242
28 04752 152440
29
30 04753 006276
31 04754 000020
32 04755 112770
33
34 04762 151400
35 04763 112770
36 04764 000402
37
38 04771 151400
39
40 04772 006277
41 04773 024020-
42 04774 132434
43 04775 000756
```

: FILL WITH ALL 1'S AND 0'S

```
LDA 0,=-125252
LDA 2,ARBUF
LDA 1,=-10
STA 0,0,2
INC 2,2
INC 1,1,SZR
JMP *-3
LDA 0,ARBUF
SUBO 1,1
SETUP 20
JSR BENTIN
SNB 0,1
ERROR
SNB 0,1
INC 1,1
SNB 0,1
JMP *-2
ERROR
INC 1,1
LOOP
JSR BENTLO
LDA 2,=-100
SUBZ# 1,2,SZR
JMP A03

LDA 0,ARBUF
SUBO 2,2
SETUP 20
JSR BENTIN
SNB 0,2
ERROR
INC 2,2
SNB 0,2
JMP *-2
ERROR
INC 2,2
JSR BENTLO
LDA 1,=-100
SUBZ# 1,2,SZR
JMP A04

: ITERATE TEST ROUTINE
: TRY AGAIN ACO AND AC2
: INITIALIZE TEST.
SUBO 0,0
SETUP 20
JSR BENTIN
SNB 0,2
ERROR
INC 2,2
JSR BENTLO
LDA 1,=-100
SUBZ# 1,2,SZR
JMP A04

: SKIP NON-ZERO BIT SHUD SKIP EVERYOTHER
: INITIALIZE TEST.
SUBO 1,1
SETUP 20
JSR BENTIN
SNB 0,1
INC 1,1
JMP *-2
ERROR
SNB 0,1
INC 1,1
LDA 2,=-100
SUBZ# 1,2,SZR
JMP A05

: ITERATE TEST ROUTINE
: SKIP ON NON-ZERO BIT SHOULD NEVER SKIP
: INITIALIZE TEST.
SUBO 1,1
SETUP 10
JSR BENTIN
SNB 0,1
INC 1,1
JMP *-2
ERROR
LOOP
JSR BENTLO
INC 1,1
LDA 2,=-100
SUBZ# 1,2,SZR
JMP A01

: ITERATE TEST ROUTINE
: DONE ALL 8 WORDS ??
SUBO 1,1
SETUP 10
JSR BENTIN
SNB 0,1
INC 1,1
JMP *-2
ERROR
LOOP
JSR BENTLO
INC 1,1
LDA 2,=-100
SUBZ# 1,2,SZR
JMP A01

: FILL WITH ALL ONE'S
SUBO 0,0
LDA 2,ARBUF
LDA 1,=-10
STA 0,0,2
INC 2,2
INC 1,1,SZR
JMP *-3
LDA 0,ARBUF
SUBO 1,1
SETUP 10
JSR BENTIN
SNB 0,1
ERROR
LOOP
JSR BENTLO
INC 1,1
LDA 2,=-100
SUBZ# 1,2,SZR
JMP A02

: SKIP ON NON-ZERO BIT SHOULD ALWAYS SKIP
: INITIALIZE TEST.
SUBO 1,1
SETUP 10
JSR BENTIN
SNB 0,1
INC 1,1
JMP *-2
ERROR
LOOP
JSR BENTLO
INC 1,1
LDA 2,=-100
SUBZ# 1,2,SZR
JMP A03

: ITERATE TEST ROUTINE
: DONE THE WHOLE TABLE?
SUBO 1,1
SETUP 10
JSR BENTIN
SNB 0,1
INC 1,1
JMP *-2
ERROR
LOOP
JSR BENTLO
INC 1,1
LDA 2,=-100
SUBZ# 1,2,SZR
JMP A04
```

10067 ECL30

```

01 JSR +1
02 MOV 3,0
03 ADI 3,0
04 SUB 1,1,SKP
05 125252
06 SETUP 100
07 JSR @ENTIN
08 05003 006276
09 05004 000100
10 05005 050031-
11 05006 106770 A05:
12 ERROR
13 05013 125400
14 05014 106770
15 05015 000402
16 ERROR
17 05022 125400
18 05023 151404
19 05024 000762
20 05025 100110
21 05026 006277
22 :TRY AGAIN WITH - BIT COUNT
23 JSR @ENTLO
24 05027 094401
25 05030 171000
26 05031 150010
27 05032 126401
28 05033 052525
29 SETUP 100
30 05034 006276
31 05035 000100
32 05036 020031-
33 05037 146770 A06:
34 05040 000402
35 05045 104010
36 05046 146770
37 05046 146770
38 ERROR
39 05053 104010
40 05054 101404
41 05055 000762
42 05056 110110
43 LOOP
44 05057 006277

```

10068 ECL30

```

01 ***** CLM *****
02
03
04 LDA 2,ABUF
05 SUBZL 0,0
06 STA 0,0,2
07 MOVZL 0,0
08 STA 0,1,2
09 SETUP 100
10 JSR @ENTIN
11 05065 006276
12 05066 000100
13 05067 024242
14 05070 102440
15 05071 106370
16 05072 000402
17 ERROR
18 05077 006277
19 LOOP
20
21 05100 006276
22 05101 000100
23 LDA 1,ABUF
24 05103 020017-
25 05104 106370
26 05105 000402
27 ERROR
28 LOOP
29 JSR @ENTLO
30
31 SETUP 100
32 JSR @ENTIN
33 LDA 1,ABUF
34 05115 024242
35 05116 102520
36 05117 106370
37 ERROR
38 LOOP
39 05124 006277
40
41 SETUP 100
42 JSR @ENTIN
43 05125 006276
44 05126 000100
45 05127 024242
46 05131 101120
47 05132 106370
48 ERROR
49 LOOP
50 JSR @ENTLO
51
52 LDA 1,=3
53 SETUP 10
54 JSR @ENTIN
55 05141 006276
56 05142 000010
57 05143 030242
58 05144 132370
59 05145 000402
60 ERROR
61 LOOP

```

10069 ECL30

```

01 ***** CLM *****
02
03
04 LDA 2,ABUF
05 SUBZL 0,0
06 STA 0,0,2
07 MOVZL 0,0
08 STA 0,1,2
09 SETUP 100
10 JSR @ENTIN
11 05065 006276
12 05066 000100
13 LDA 1,ABUF
14 SUBO 0,0
15 CLM 0,1
16 JMP +2
17 ERROR
18 JSR @ENTLO
19
20 SETUP 100
21 JSR @ENTIN
22 05100 006276
23 LDA 1,ABUF
24 05103 020017-
25 05104 106370
26 05105 000402
27 ERROR
28 LOOP
29 JSR @ENTLO
30
31 SETUP 100
32 JSR @ENTIN
33 LDA 1,ABUF
34 05115 024242
35 05116 102520
36 05117 106370
37 ERROR
38 LOOP
39 05124 006277
40
41 SETUP 100
42 JSR @ENTIN
43 05125 006276
44 05126 000100
45 05127 024242
46 05131 101120
47 05132 106370
48 ERROR
49 LOOP
50 JSR @ENTLO
51
52 LDA 1,=3
53 SETUP 10
54 JSR @ENTIN
55 05141 006276
56 05142 000010
57 05143 030242
58 05144 132370
59 05145 000402
60 ERROR
61 LOOP

```

10070 ECL30

```

01 ***** CLM *****
02
03
04 LDA 2,ABUF
05 SUBZL 0,0
06 STA 0,0,2
07 MOVZL 0,0
08 STA 0,1,2
09 SETUP 100
10 JSR @ENTIN
11 05065 006276
12 05066 000100
13 LDA 1,ABUF
14 SUBO 0,0
15 CLM 0,1
16 JMP +2
17 ERROR
18 JSR @ENTLO
19
20 SETUP 100
21 JSR @ENTIN
22 05100 006276
23 LDA 1,ABUF
24 05103 020017-
25 05104 106370
26 05105 000402
27 ERROR
28 LOOP
29 JSR @ENTLO
30
31 SETUP 100
32 JSR @ENTIN
33 LDA 1,ABUF
34 05115 024242
35 05116 102520
36 05117 106370
37 ERROR
38 LOOP
39 05124 006277
40
41 SETUP 100
42 JSR @ENTIN
43 05125 006276
44 05126 000100
45 05127 024242
46 05131 101120
47 05132 106370
48 ERROR
49 LOOP
50 JSR @ENTLO
51
52 LDA 1,=3
53 SETUP 10
54 JSR @ENTIN
55 05141 006276
56 05142 000010
57 05143 030242
58 05144 132370
59 05145 000402
60 ERROR
61 LOOP

```

0069 ECL30  
01 05152 006277  
02 05153 125404  
03 05154 000765

JSR @ENTLO  
INC I,1,SZR  
JMP B04

:ITERATE TEST ROUTINE  
:QUIT WHEN AC1 = 0

```
10070 ECL30
01
02
03 05155 006276      B05:      SETUP 100      :TRY AGAIN WITH MINUS ONE
04 05156 000100      JSR @ENTIN    :INITIALIZE TEST.
05 05157 102000      ADC 0,0
06 05160 030242      LDA 2,ABUF    :SET AC0 AT -1
07 05161 112370      CLM 0,2      :ADDR OF TABLE TO AC2
08 05162 000402      JMP *+2      :END ERROR IF NO SKIP
09      ERROR
10      LOOP
11 05167 006277      JSR @ENTLO    :ITERATE TEST ROUTINE
12
13
14 05170 006276      B06:      SETUP 100      :IF ACS = AC0 TABLE FOLLOWS INSTR
15 05171 000100      JSR @ENTIN    :INITIALIZE TEST.
16 05172 102440      SUBO 0,0     :CLEAR AC0
17 05173 102370      CLM 0,0     :NO SKIP THIS CASE
18 05174 000001      1
19 05175 000002      2
20 05176 000402      JMP *+2      :DID SKIP !!
21      ERROR
22      LOOP
23 05203 006277      JSR @ENTLO    :ITERATE TEST ROUTINE
24
25
26 05204 006276      B07:      SETUP 100      :WITH ACS = 1 SHUD ALWAYS SKIP
27 05205 000100      JSR @ENTIN    :INITIALIZE TEST.
28 05206 102520      SUBZL 0,0
29 05207 102370      CLM 0,0
30 05210 000001      1
31 05211 000002      2
32      ERROR
33      LOOP
34 05216 006277      JSR @ENTLO    :ITERATE TEST ROUTINE
35
36
37
38 05217 006276      B07A:     SETUP 100      :TRY ANOTHER AC
39 05220 000100      JSR @ENTIN    :INITIALIZE TEST.
40 05221 126520      SUBZL 1,1
41 05222 126370      CLM 1,1
42 05223 000001      1
43 05224 000002      2
44      ERROR
45      LOOP
46 05231 006277      JSR @ENTLO    :ITERATE TEST ROUTINE
47
48
49 05232 006276      B07B:     SETUP 100      :TRY AC2 THIS TIME
50 05233 000100      JSR @ENTIN    :INITIALIZE TEST.
51 05234 152520      SUBZL 2,2
52 05235 152370      CLM 2,2
53 05236 000001      1
54 05237 000002      2
55      ERROR
56      LOOP
57 05244 006277      JSR @ENTLO    :ITERATE TEST ROUTINE
58
59
60 05245 006276      B07C:     SETUP 100      :ONCE AGAIN, AC3 THIS TIME
        JSR @ENTIN    :INITIALIZE TEST.
```

0071 ECL30  
01 05246 000100  
02 05247 176520  
03 05250 176370  
04 05251 000001  
05 05252 000002  
06  
07  
08 05257 006277

100  
SUBZL 3,3  
CLM 3,3  
1  
2  
ERROR  
LOOP  
JSR @ENTLO

:ITERATE TEST ROUTINE

10072 ECL30  
01  
02  
03 05260 006276  
04 05261 000100  
05 05262 020025-  
06 05263 102370  
07 05264 000001  
08 05265 000002  
09  
10  
11 05272 006277  
12  
13  
14 05273 006276  
15 05274 000100  
16  
17 05275 006301  
18 05276 102370  
19 05277 100000  
20 05300 077777  
21  
22  
23 05305 006277  
24  
25  
26 05306 006276  
27 05307 000100  
28  
29 05310 006301  
30 05311 105000  
31 05312 126370  
32 05313 100000  
33 05314 077777  
34  
35  
36 05321 006277  
37  
38  
39 05322 006276  
40 05323 000100  
41  
42 05324 006301  
43 05325 110000  
44 05326 152370  
45 05327 100000  
46 05330 077777  
47  
48  
49 05335 006277  
50  
51  
52 05336 006276  
53 05337 000100  
54  
55 05340 006301  
56 05341 115000  
57 05342 176370  
58 05343 100000  
59 05344 077777  
60

SETUP 100  
JSR @ENTIN  
100  
LOA 0,=2  
CLM 0,0  
1  
2  
ERROR  
LOOP  
JSR @ENTLO

:ITERATE TEST ROUTINE

:RANDOM NUMBER IN ACS  
:INITIALIZE TEST.  
:TABLE VALUES MAX NEG AND MAX POS  
:C(AC0)=RANDOM #  
:L = 100000 AND H = 77777  
:SHOULD ALWAYS SKIP  
:DIDN'T YOU RASCAL!!  
:ITERATE TEST ROUTINE  
:TRY AGAIN WITH AC1 = ACS  
:INITIALIZE TEST.  
:C(AC0)=RANDOM #  
:ITERATE TEST ROUTINE  
:TRY IT WITH AC2 = ACS  
:INITIALIZE TEST.  
:C(AC0)=RANDOM #  
:ITERATE TEST ROUTINE  
:HOW ABOUT AC3 ?  
:INITIALIZE TEST.  
:C(AC0)=RANDOM #

SETUP 100  
JSR @ENTIN  
100  
RAND  
JSR @ENTRA  
MOV 0,1  
CLM 1,1  
180  
77777  
ERROR  
LOOP  
JSR @ENTLO

809A:

SETUP 100  
JSR @ENTIN  
100  
RAND  
JSR @ENTRA  
MOV 0,1  
CLM 1,1  
180  
77777  
ERROR  
LOOP  
JSR @ENTLO

809B:

SETUP 100  
JSR @ENTIN  
100  
RAND  
JSR @ENTRA  
MOV 0,2  
CLM 2,2  
180  
77777  
ERROR  
LOOP  
JSR @ENTLO

809C:

SETUP 100  
JSR @ENTIN  
100  
RAND  
JSR @ENTRA  
MOV 0,5  
CLM 3,5  
180  
77777  
ERROR





```

10075 ECL30
01
02
03 05442 006276
04 05443 000100
05 05444 004401
06 05445 024015-
07 05446 137000
08 05447 054253
09 05448 102670
10 000001
11 05452 030040
12 05453 021000
13 05454 024253
14 05455 106434
15
16 05462 014040
17 05463 000401
18
19 05464 006277
20
21
22 05465 006276
23 05466 000100
24 05467 004401
25 05470 024015-
26 05471 137000
27 05472 054253
28 05473 102670
29 172557
30 05475 030040
31 05476 021000
32 05477 024253
33 05500 106434
34
35 05505 014040
36 05506 000401
37
38 05507 006277

TST2:
:MODE 1 ADDRESSING - DIRECT
:INITIALIZE TEST.
:ADDRESS OF NEXT INSTR AFTER PSJ
:JMP ADDRESS = 1 + PC OF 2ND WORD OF PSJ
LDA 2,SP
LDA 0,0,2
LDA 1,TEMP
SUB7# 0,1,SZR
ERROR
DSZ SP
JMP .+1
LOOP
JSR @ENTLO
:ITERATE TEST ROUTINE

TST3:
:MODE ONE - INDIRECT ADDRESSING
:INITIALIZE TEST.
:CHECK THAT PC PUSHED ON STACK
LDA 2,SP
LDA 0,0,2
LDA 1,TEMP
SUB7# 0,1,SZR
ERROR
DSZ SP
JMP .+1
LOOP
JSR @ENTLO
:ITERATE TEST ROUTINE

TST4:
:MODE TWO DIRECT ADDRESSING
:INITIALIZE TEST.
:EFF ADDR = CONTENTS OF 2ND WRD PSJ
:PLUS CONTENTS AC2
SETUP 100
JSR @ENTIN
JSR .+1
LDA 1,56
ADD 1,5
STA 3,TEMP
SUBZL 2,2
PSHJ .+1,2
LDA 2,SP
LDA 0,0,2
LDA 1,TEMP
SUB7# 0,1,SZR
ERROR
DSZ SP
JMP .+1
LOOP
JSR @ENTLO
:ITERATE TEST ROUTINE

TST5:
:MODE TWO -INDIRECT ADDRESSING
:INITIALIZE TEST.
:EFF ADR = 2ND WORD PSJ PLUS
:CONTENTS AC2
SETUP 100
JSR @ENTIN
JSR .+1
LDA 1,56
ADD 1,5
STA 3,TEMP
LDA 2,TEMP
PSHJ 80,2
LDA 2,SP
LDA 0,0,2
LDA 1,TEMP
SUB7# 0,1,SZR
ERROR
DSZ SP
JMP .+1
LOOP
JSR @ENTLO
:ITERATE TEST ROUTINE

10076 ECL30
01
02
03 05510 006276
04 05511 000100
05 05512 004401
06 05513 024015-
07 05514 137000
08 05515 054253
09 05516 152520
10 05517 103270
11 005520
12 05521 030040
13 05522 021000
14 05523 024253
15 05524 106434
16
17 05531 014040
18 05532 000401
19
20 05533 006277
21
22 05534 006276
23 05535 000100
24 05536 004401
25 05537 024015-
26 05540 137000
27 05541 054253
28 05542 030014-
29 05543 103270
30
31 100000
32 05545 030040
33 05546 021000
34 05547 024253
35 05550 106434
36
37 05555 014040
38 05556 000401
39
40 05557 006277
41
42

```

```

10077 ECL30
01 05560 006276
02 05561 000100
03 05562 004401
04 05563 024016
05 05564 137000
06 05565 054253
07 05566 103670
08 05567 000000
09 05570 030040
10 05571 021000
11 05572 024253
12 05573 106434
13 05600 014040
14 05601 000401
15 05602 006277
16
17
18
19
20
21
22 05603 006276
23 05604 000100
24 05605 004401
25 05606 177770
26 000007
27 05610 054253
28 05611 171400
29 05612 050403
30 05613 103670
31 100000
32 05615 005616
33 05616 030040
34 05617 021000
35 05620 024253
36 05621 106434
37 05626 014040
38 05627 000401
39
40
41 05630 006277

TST6:
      SETUP 100
      JSR @ENTIN
      100
      JSR .+1
      LDA 1,5
      ADD 1,3
      STA 3,TEMP
      PSNJ 0,3
      LDA 2,SP
      LDA 0,0,2
      LDA 1,TEMP
      SUBZ# 0,1,SZR
      ERROR
      DSZ SP
      JMP .+1
      JSR @ENTLO
      LOOP
      TST7:
      SETUP 100
      JSR @ENTIN
      100
      JSR .+1
      ADDI 7,3
      STA 3,TEMP
      INC 3,2
      STA 2,+3
      PSNJ @0,3
      .+1
      LDA 2,SP
      LDA 0,0,2
      LDA 1,TEMP
      SUBZ# 0,1,SZR
      ERROR
      DSZ SP
      JMP .+1
      LOOP
      JSR @ENTLO

      :MODE 3 DIRECT ADDRESSING
      :INITIALIZE TEST.
      :ADDRESS = CONTENTS 2ND WORD PSNJ
      :PLUS CONTENTS AC3
      :ITERATE TEST ROUTINE
      :MODE THREE - INDIRECT ADDRESSING
      :INITIALIZE TEST.
      :EFF ADDR = CONTENTS OF 2ND WORD OF PSNJ
      :FOR @
      :PLUS CONTENTS AC3
      :PSNJ @THIS LOC TO NXT
      :ITERATE TEST ROUTINE

10078 ECL39
01 05631 004401
02 05632 177770
03 000023
04 05634 054402
05 05635 101001
06 05636 005655
07 05637 006276
08 05640 000100
09 05641 020022
10 05642 040041
11 05643 126620
12 05644 123000
13 05645 040040
14 05646 040042
15 05647 004401
16 05650 024016
17 05651 137000
18 05652 054043
19 05653 102770
20 05655 005655
21 05656 030040
22 05656 173770
23 05656 177773
24 05660 021000
25 05661 024755
26 05662 106434
27
28
29
30 05667 006277
31
32 05670 024022
33 05671 044040
34 05672 044041
35 05673 024021
36 05674 044042
37 05675 024013
38 05676 044043

TSTR1:
      JSR @ENTIN
      100
      LDA 0,+420
      STA 0,SF
      SUBZR 1,1
      ADD 1,0
      STA 0,SP
      STA 0,USL
      JSR .+1
      LDA 1,5
      ADD 1,3
      STA 3,USF
      PSNJ .+2,+0
      LDA 2,SP
      ADDI -5,2
      LDA 0,0,2
      LDA 1,TSTR1-1
      SUBZ# 0,1,SZR
      ERROR
      LOOP
      JSR @ENTLO
      LDA 1,+420
      STA 1,SP
      STA 1,SF
      LDA 1,+77777
      STA 1,USL
      LDA 1,+STFL
      STA 1,USF

      :AGAIN CHECK PC PUSHED IN RETURN BLOCK
      :INITIALIZE TEST.
      :GO TO NEXT INSTR IN EITHER CASE
      :ITERATE TEST ROUTINE

```

10079 ECL30

```

01 OSZ PASSIN
02 JMP DOMOR
03 ISZ PASS
04 MOV 0,0,SKP
05 HALT
06 STA 0,PASSVL
07 LDA 0,PASSIN
08 STA 0,PASSIN
09 READS 0
10 MOVL# 0,0,5ZC
11 JMP *3
12 ELDA 0,SAREG
13 ANDI 184,0
14 MOV 0,0,SZR
15 JMP PSCK1
16 JSR @INNESS
17 PASHES
18 MOVZ 1,1
19 LDA 1,PASS
20 JSR @IPDEC
21 LDA 3,.EGGS
22 LDA 0,0,3
23 MOV 0,0,SNR
24 JMP *8
25 OSZ 3,3
26 JMP *46
27 IORST 0,3,3
28 LDA 3,0,3
29 STA 0,2,3
30 JMP 0,1,78
31 DOMOR: CALL
32 JSR @ICAL
33 REL
34 40 05740 001015
35 41
36 42
37 43
38 44
39 45
40 46
41 47
42 48
43 49
44 50
45 51
46 52
47 53
48 54
49 55
50 56
51 57
52 58
53 59
54 60

```

10080 ECL30

```

01 : *****EGGS & DIRT DATA BLOCKS*****
02 :
03 :
04 :
05 :
06 :
07 :
08 :
09 :
10 :
11 :
12 :
13 :
14 :
15 :
16 :
17 :
18 :
19 :
20 :
21 :
22 :
23 :
24 :
25 :
26 :
27 :
28 :
29 :
30 :
31 :
32 :
33 :
34 :
35 :
36 :
37 :
38 :
39 :
40 :
41 :
42 :
43 :
44 :
45 :
46 :
47 :
48 :
49 :
50 :
51 :
52 :
53 :
54 :
55 :
56 :
57 :
58 :
59 :
60 :

```

\*\*\*\*\*EGGS & DIRT DATA BLOCKS\*\*\*\*\*

```

:EGGS:
:DTOS AUTO MODE SWITCH
:PRIMARY DEVICE CODE TO BE TESTED
:CAT SWITCH, SET IF CAT LOADED
:PASS COUNT, # OF TIMES TO RHN
:RETURN POINT TO RESTORE DTOS
:DEFAULT SWITCH REGISTER

```

```

:EGGS:
:DTOS AUTO MODE SWITCH
:PRIMARY DEVICE CODE TO BE TESTED
:CAT SWITCH, SET IF CAT LOADED
:PASS COUNT, # OF TIMES TO RHN
:RETURN POINT TO RESTORE DTOS
:DEFAULT SWITCH REGISTER

```

PRBEND: PRBEND

.TXT /COPYRIGHT(C)DGC,1974,76

046114

051040

043511

052110

020123

042522

053122

042105

000000

141705

144114

031520

031705

130469

000000

000000

06005 000200

06006 175772

06007 000000

06010 000000

06011 000000

06012 000000

06013 000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

000000

0081 ECL30  
01 000420  
02 177774  
03 177776  
04 000002  
05 052525  
06 125252  
07 000240  
08 177770  
09 000212  
10 104400  
11 022000  
12 000377  
13 000060  
14 000011  
15 033031  
16 000144  
17 000200  
18 000400  
19 001777  
20 000276  
21 000272  
22 000004

0082 ECL30

|      |        |       |       |       |        |       |       |       |
|------|--------|-------|-------|-------|--------|-------|-------|-------|
| A01  | 004653 | 65/13 | 65/24 | 65/11 | 65/28  | 65/34 | 66/03 | 66/09 |
| A02  | 004701 | 65/36 | 65/46 | 68/12 | 68/23  | 68/34 | 68/44 | 68/56 |
| A03  | 004726 | 66/11 | 66/25 | 14/08 | 15/21  | 15/34 | 16/05 | 16/28 |
| A04  | 004753 | 66/29 | 66/43 | 15/22 | 16/15  | 16/30 | 16/46 |       |
| A05  | 005006 | 67/11 | 67/19 | 15/23 | 16/16  | 16/32 | 16/47 |       |
| A06  | 005037 | 67/33 | 67/41 | 16/34 | 16/59  | 16/75 | 16/87 |       |
| ABUF | 000242 | 10/29 | 65/05 | 26/11 | 26/15  | 26/31 | 26/37 | 25/41 |
|      |        | 70/06 | 68/04 | 30/50 | 30/53  | 31/52 | 31/54 | 33/39 |
|      |        | 16/58 |       | 33/51 | 34/21  | 34/23 | 34/49 | 34/52 |
| AC0  | 000225 | 10/16 | 14/08 | 35/44 | 36/16  | 36/34 | 36/52 | 37/18 |
| AC1  | 000226 | 10/17 | 15/22 | 38/31 | 38/43  | 38/55 | 39/12 | 39/26 |
| AC2  | 000227 | 10/18 | 15/23 | 41/33 | 41/36  | 42/53 | 43/16 | 44/22 |
| AC3  | 000230 | 10/19 | 16/34 | 44/44 | 44/54  | 45/10 | 45/21 | 45/32 |
|      |        | 25/43 | 26/11 | 46/10 | 46/21  | 46/33 | 46/44 | 47/11 |
|      |        | 30/50 | 30/53 | 47/33 | 47/44  | 47/54 | 48/10 | 48/21 |
|      |        | 33/51 | 34/21 | 49/12 | 49/31  | 49/49 | 50/17 | 50/35 |
|      |        | 35/44 | 36/16 | 51/30 | 51/39  | 51/50 | 52/20 | 52/30 |
|      |        | 38/31 | 38/43 | 52/50 | 53/10  | 53/20 | 53/30 | 53/39 |
|      |        | 41/33 | 41/36 | 54/20 | 54/30  | 54/39 | 54/50 | 55/10 |
|      |        | 44/44 | 44/54 | 56/36 | 57/17  | 57/29 | 57/54 | 58/24 |
|      |        | 46/10 | 46/21 | 59/54 | 60/24  | 61/15 | 61/27 | 61/52 |
|      |        | 47/33 | 47/44 | 63/27 | 63/53  | 64/24 | 65/19 | 65/41 |
|      |        | 49/12 | 49/31 | 66/34 | 66/38  | 67/13 | 67/17 | 67/36 |
|      |        | 51/30 | 51/39 | 68/28 | 68/38  | 68/49 | 68/60 | 68/70 |
|      |        | 52/50 | 53/10 | 70/45 | 70/56  | 71/07 | 72/10 | 72/22 |
|      |        | 54/20 | 54/30 | 73/01 | 73/16  | 74/22 | 74/41 | 75/16 |
|      |        | 56/36 | 57/17 | 76/33 | 77/15  | 77/38 | 78/41 |       |
|      |        | 59/54 | 60/24 | 7/33  | 35/32  | 36/04 | 36/22 | 36/40 |
|      |        | 63/27 | 63/53 | 7/20  | 30/10  | 30/19 | 30/28 | 30/37 |
|      |        | 66/34 | 66/38 | 8/13  | 49/18  | 49/36 | 50/04 | 50/22 |
|      |        | 68/28 | 68/38 | 7/27  | 34/05  | 80/05 |       |       |
|      |        | 70/45 | 70/56 | 13/14 | 16/48  |       |       |       |
|      |        | 73/01 | 73/16 | 68/09 |        |       |       |       |
|      |        | 76/33 | 77/15 | 801   | 005100 |       |       |       |
|      |        |       |       | 68/20 |        |       |       |       |
|      |        |       |       | 68/31 |        |       |       |       |
|      |        |       |       | 68/41 |        |       |       |       |
|      |        |       |       | 68/53 |        |       |       |       |
|      |        |       |       | 70/02 |        |       |       |       |
|      |        |       |       | 70/13 |        |       |       |       |
|      |        |       |       | 70/25 |        |       |       |       |
|      |        |       |       | 70/37 |        |       |       |       |
|      |        |       |       | 70/48 |        |       |       |       |
|      |        |       |       | 70/59 |        |       |       |       |
|      |        |       |       | 72/02 |        |       |       |       |
|      |        |       |       | 72/13 |        |       |       |       |
|      |        |       |       | 72/25 |        |       |       |       |
|      |        |       |       | 72/38 |        |       |       |       |
|      |        |       |       | 72/51 |        |       |       |       |
|      |        |       |       | 73/04 |        |       |       |       |
|      |        |       |       | 11/08 |        |       |       |       |
|      |        |       |       | 11/04 |        |       |       |       |
|      |        |       |       | 11/09 |        |       |       |       |
|      |        |       |       | 13/33 |        |       |       |       |
|      |        |       |       | 18/37 |        |       |       |       |

ACD 000105 MC  
ADDAL 000063 MC  
ANDIA 000154 MC  
ASSEN 000073 MC  
AUTO 005741  
800 005065  
801 005100  
802 005113  
803 005125  
804 005141  
805 005155  
806 005170  
807 005204  
807A 005217  
807B 005232  
807C 005245  
808 005260  
809 005273  
809A 005300  
809B 005322  
809C 005336  
810 005352  
BAMBL 000272  
BAMBY 000262

BEGIN 001342  
 86NAD 000202  
 CAL 000767  
 CALO 000237  
 CAL1 000240  
 CAL2 000241  
 CALL 000016  
 CATSW 005743  
 CATSW 001144  
 CHAR1 001173  
 CHAR2 001502  
 CHAR3 001210  
 CHAR4 001221  
 CHARE 000232  
 CHDRZ 000233  
 CHRVS 001266  
 CRY 000231  
 DEV 005742  
 DIRT 005776  
 DDMOR 000537  
 DT088 000200  
 E6GS 005741  
 ENTER 000300

11/14  
 9/45  
 10/08  
 10/26  
 10/27  
 10/28  
 6/19  
 9/31  
 12/11  
 19/32  
 21/30  
 21/39  
 21/40  
 21/27  
 10/21  
 10/22  
 21/05  
 10/20  
 80/06  
 9/05  
 79/03  
 9/45  
 9/16  
 26/15  
 31/52  
 34/23  
 36/34  
 38/55  
 42/53  
 45/10  
 46/33  
 49/49  
 51/50  
 53/20  
 54/39  
 57/29  
 61/15  
 64/24  
 67/13  
 68/60  
 71/07  
 72/10  
 74/22  
 77/38  
 11/23  
 26/04  
 34/03  
 37/02  
 40/23  
 44/36  
 47/25  
 49/02  
 51/23  
 52/42  
 54/13  
 56/21  
 59/33

25/02  
 17/29  
 17/30  
 17/31  
 18/38  
 18/08  
 21/03  
 80/07  
 21/53  
 21/50  
 22/03  
 16/25  
 80/53  
 80/04  
 25/27  
 26/58  
 33/15  
 34/52  
 37/18  
 39/26  
 44/11  
 45/32  
 46/54  
 48/21  
 50/35  
 52/10  
 53/39  
 55/10  
 58/28  
 61/52  
 65/41  
 67/36  
 70/10  
 72/22  
 74/41  
 75/16  
 25/20  
 31/04  
 35/03  
 37/03  
 41/03  
 45/02  
 46/13  
 47/36  
 49/16  
 51/33  
 53/03  
 54/33  
 57/21  
 60/05

63/18  
 67/08  
 70/14  
 72/14  
 75/03  
 75/03  
 11/24  
 29/08  
 34/50  
 37/19  
 37/38  
 38/14  
 40/46  
 44/45  
 46/11  
 47/34  
 49/13  
 51/31  
 52/41  
 54/51  
 56/37  
 59/55  
 63/30  
 67/22  
 70/11  
 72/23  
 75/19  
 11/26  
 41/43  
 16/18  
 11/19  
 16/58  
 16/12  
 9/56  
 6/24  
 26/14  
 31/51  
 34/22  
 36/33  
 38/53  
 42/52  
 45/09  
 46/33  
 47/53  
 49/48  
 51/49  
 53/19  
 54/38  
 57/28  
 61/14  
 64/23  
 67/12  
 68/48  
 71/06  
 74/21  
 77/37  
 47/02  
 47/13  
 47/26  
 47/55  
 47/74

63/32  
 67/30  
 70/14  
 72/26  
 75/22  
 75/22  
 25/17  
 30/54  
 34/53  
 37/38  
 38/14  
 41/37  
 45/11  
 46/22  
 47/34  
 49/32  
 51/40  
 52/41  
 53/11  
 54/51  
 57/18  
 60/25  
 63/54  
 67/44  
 70/34  
 72/36  
 75/38  
 18/13  
 43/06  
 72/42  
 23/14  
 11/25  
 16/51  
 16/04  
 9/56  
 6/24  
 26/14  
 26/36  
 31/53  
 34/48  
 36/51  
 39/11  
 43/15  
 45/20  
 46/43  
 48/09  
 50/16  
 52/09  
 53/29  
 54/49  
 57/28  
 61/26  
 65/18  
 67/16  
 68/59  
 70/09  
 72/09  
 74/40  
 77/40  
 78/28

ENTWLO 000277  
 80/07  
 21/57  
 25/29  
 29/01  
 33/27  
 35/23  
 37/37  
 40/19  
 44/33  
 47/22  
 48/32  
 51/10  
 52/39  
 54/10  
 56/36  
 59/27  
 63/15  
 66/20  
 68/17  
 70/33  
 72/48  
 75/35  
 76/17  
 26/04  
 33/19  
 36/02  
 39/03  
 44/03  
 45/24  
 46/47  
 48/02  
 49/34  
 50/02  
 52/03  
 53/23  
 55/04  
 58/05  
 61/19  
 61/33  
 62/05  
 63/07

ENTRA 000301  
 80/07  
 21/57  
 25/29  
 29/01  
 33/27  
 35/23  
 37/37  
 40/19  
 44/33  
 47/22  
 48/32  
 51/10  
 52/39  
 54/10  
 56/36  
 59/27  
 63/15  
 66/20  
 68/17  
 70/33  
 72/48  
 75/35  
 76/17  
 26/04  
 33/19  
 36/02  
 39/03  
 44/03  
 45/24  
 46/47  
 48/02  
 49/34  
 50/02  
 52/03  
 53/23  
 55/04  
 58/05  
 61/19  
 61/33  
 62/05  
 63/07

ENTR 000225 MC  
 80/07  
 21/57  
 25/29  
 29/01  
 33/27  
 35/23  
 37/37  
 40/19  
 44/33  
 47/22  
 48/32  
 51/10  
 52/39  
 54/10  
 56/36  
 59/27  
 63/15  
 66/20  
 68/17  
 70/33  
 72/48  
 75/35  
 76/17  
 26/04  
 33/19  
 36/02  
 39/03  
 44/03  
 45/24  
 46/47  
 48/02  
 49/34  
 50/02  
 52/03  
 53/23  
 55/04  
 58/05  
 61/19  
 61/33  
 62/05  
 63/07

ENTR 000225 MC  
 80/07  
 21/57  
 25/29  
 29/01  
 33/27  
 35/23  
 37/37  
 40/19  
 44/33  
 47/22  
 48/32  
 51/10  
 52/39  
 54/10  
 56/36  
 59/27  
 63/15  
 66/20  
 68/17  
 70/33  
 72/48  
 75/35  
 76/17  
 26/04  
 33/19  
 36/02  
 39/03  
 44/03  
 45/24  
 46/47  
 48/02  
 49/34  
 50/02  
 52/03  
 53/23  
 55/04  
 58/05  
 61/19  
 61/33  
 62/05  
 63/07

ENTR 000225 MC  
 80/07  
 21/57  
 25/29  
 29/01  
 33/27  
 35/23  
 37/37  
 40/19  
 44/33  
 47/22  
 48/32  
 51/10  
 52/39  
 54/10  
 56/36  
 59/27  
 63/15  
 66/20  
 68/17  
 70/33  
 72/48  
 75/35  
 76/17  
 26/04  
 33/19  
 36/02  
 39/03  
 44/03  
 45/24  
 46/47  
 48/02  
 49/34  
 50/02  
 52/03  
 53/23  
 55/04  
 58/05  
 61/19  
 61/33  
 62/05  
 63/07

0085 ECL 30

|       |        |       |
|-------|--------|-------|
| EMI4A | 003752 | 54/02 |
| EMI5  | 003765 | 54/12 |
| EMI6  | 004000 | 54/22 |
| EMI7  | 004013 | 54/32 |
| EMI8  | 004025 | 54/41 |
| EX0   | 001343 | 25/08 |
| EX00  | 002544 | 41/02 |
| EX01  | 002613 | 41/39 |
| EX02  | 002706 | 43/02 |
| EX03  | 002726 | 44/02 |
| EX04  | 002742 | 44/13 |
| EX04A | 002756 | 44/24 |
| EX05  | 002772 | 44/35 |
| EX06  | 003006 | 44/46 |
| EX07  | 003302 | 44/01 |
| EX08  | 003316 | 48/12 |
| EX09  | 003332 | 48/23 |
| EX1   | 001356 | 25/19 |
| EX10  | 003347 | 48/35 |
| EX10A | 003364 | 49/01 |
| EX11  | 003401 | 49/15 |
| EX12  | 003422 | 49/33 |
| EX13  | 003443 | 50/01 |
| EX14  | 003464 | 50/19 |
| EX14A | 003505 | 51/02 |
| EX15  | 003520 | 51/12 |
| EX16  | 003533 | 51/22 |
| EX17  | 003546 | 51/32 |
| EX18  | 003560 | 51/41 |
| EX19  | 004042 | 55/03 |
| EX2   | 001376 | 25/32 |
| EX20  | 004057 | 55/16 |
| EX2.1 | 001401 | 25/36 |
| EX2.2 | 001423 | 25/46 |
| EX3   | 001424 | 26/02 |
| EX3.1 | 001427 | 26/06 |
| EX3.2 | 001453 | 26/18 |
| EX4   | 001454 | 26/25 |
| EX4.1 | 001457 | 26/29 |
| EX4.2 | 001477 | 26/40 |
| EX5   | 001500 | 26/46 |
| EX5.1 | 001503 | 26/50 |
| EX5.2 | 001523 | 27/01 |
| EX6   | 001524 | 28/02 |
| EX6.1 | 001527 | 28/06 |
| EX6.2 | 001605 | 29/04 |
| EX7   | 001606 | 30/03 |
| EX7R  | 005677 | 79/02 |
| EX7.1 | 001654 | 30/46 |
| EXD   | 001675 | 31/02 |
| EXD.1 | 001700 | 31/06 |
| EXD.2 | 001772 | 31/57 |
| EXF   | 001773 | 33/06 |
| EXG   | 002007 | 33/18 |
| EXH   | 002023 | 33/30 |
| EXI   | 002037 | 34/01 |
| EXJ   | 002053 | 34/05 |
| EXJ.1 | 002056 | 34/28 |

0086 FCL 30

|       |        | 34/29 |  |  |  |  |
|-------|--------|-------|--|--|--|--|
| EXJ.2 | 002112 | 34/26 |  |  |  |  |
| EXK   | 002113 | 34/31 |  |  |  |  |
| EXL   | 002145 | 35/02 |  |  |  |  |
| EXM   | 002202 | 35/24 |  |  |  |  |
| EXN   | 002222 | 36/01 |  |  |  |  |
| EXO   | 002242 | 36/19 |  |  |  |  |
| EXP   | 002262 | 36/37 |  |  |  |  |
| EXR   | 002302 | 37/01 |  |  |  |  |
| EXT   | 002325 | 37/22 |  |  |  |  |
| EXU   | 002346 | 38/02 |  |  |  |  |
| EXV   | 002364 | 38/16 |  |  |  |  |
| EXW   | 002416 | 39/02 |  |  |  |  |
| EXX   | 002453 | 39/15 |  |  |  |  |
| EXY   | 002471 | 40/03 |  |  |  |  |
| EXZ   | 002513 | 40/22 |  |  |  |  |
| EY03  | 003021 | 45/01 |  |  |  |  |
| EY04  | 003035 | 45/12 |  |  |  |  |
| EY04A | 003051 | 45/23 |  |  |  |  |
| EY05  | 003065 | 45/34 |  |  |  |  |
| EY04  | 003101 | 45/45 |  |  |  |  |
| EY14A | 003574 | 52/02 |  |  |  |  |
| EY15  | 003607 | 52/12 |  |  |  |  |
| EY16  | 003622 | 52/22 |  |  |  |  |
| EY17  | 003635 | 52/32 |  |  |  |  |
| EY18  | 003647 | 52/41 |  |  |  |  |
| EY20  | 004101 | 56/01 |  |  |  |  |
| EZ03  | 003114 | 46/01 |  |  |  |  |
| EZ04  | 003130 | 46/12 |  |  |  |  |
| EZ04A | 003144 | 46/24 |  |  |  |  |
| EZ05  | 003160 | 46/35 |  |  |  |  |
| EZ06  | 003174 | 46/46 |  |  |  |  |
| EZ14A | 003663 | 53/02 |  |  |  |  |
| EZ15  | 003676 | 53/12 |  |  |  |  |
| EZ16  | 003711 | 53/22 |  |  |  |  |
| EZ17  | 003724 | 53/32 |  |  |  |  |
| EZ1A  | 003736 | 53/41 |  |  |  |  |
| EZ20  | 004123 | 56/20 |  |  |  |  |
| HEADE | 001253 | 16/23 |  |  |  |  |
| ICAL  | 000220 | 10/08 |  |  |  |  |
| ICAT  | 000112 | 9/24  |  |  |  |  |
| IMESS | 000221 | 10/09 |  |  |  |  |
| INIT  | 000566 | 16/22 |  |  |  |  |
| INITI | 000610 | 11/17 |  |  |  |  |
| IP0EC | 000224 | 14/24 |  |  |  |  |
| IP0CT | 000223 | 10/12 |  |  |  |  |
|       |        | 13/11 |  |  |  |  |
|       |        | 13/08 |  |  |  |  |
|       |        | 13/42 |  |  |  |  |
|       |        | 13/10 |  |  |  |  |
| ISIZE | 000222 | 12/06 |  |  |  |  |
| ITR   | 000206 | 4/52  |  |  |  |  |
| ITRCI | 000207 | 9/53  |  |  |  |  |
| ITREC | 000211 | 9/55  |  |  |  |  |
| ITRER | 000210 | 9/54  |  |  |  |  |
|       |        | 23/44 |  |  |  |  |
|       |        | 9/46  |  |  |  |  |
|       |        | 14/07 |  |  |  |  |
|       |        | 9/31  |  |  |  |  |
|       |        | 25/43 |  |  |  |  |
|       |        | 13/22 |  |  |  |  |
|       |        | 13/10 |  |  |  |  |
|       |        | 9/59  |  |  |  |  |
| ISIZE | 000222 | 12/06 |  |  |  |  |
| ITR   | 000206 | 4/52  |  |  |  |  |
| ITRCI | 000207 | 9/53  |  |  |  |  |
| ITREC | 000211 | 9/55  |  |  |  |  |
| ITRER | 000210 | 9/54  |  |  |  |  |
|       |        | 23/44 |  |  |  |  |
|       |        | 9/46  |  |  |  |  |
|       |        | 14/07 |  |  |  |  |
|       |        | 9/31  |  |  |  |  |
|       |        | 25/43 |  |  |  |  |
|       |        | 13/22 |  |  |  |  |
|       |        | 13/10 |  |  |  |  |
|       |        | 9/59  |  |  |  |  |
| KGRIF | 001316 | 16/44 |  |  |  |  |
| LISYN | 000215 | 16/40 |  |  |  |  |





0089 ECL30

|       |        |       |       |       |       |
|-------|--------|-------|-------|-------|-------|
| X01   | 004151 | 78/15 | 78/36 |       |       |
| X02   | 004165 | 57/08 |       |       |       |
| X03   | 004205 | 57/20 |       |       |       |
| X03A  | 004217 | 57/34 |       |       |       |
| X04   | 004234 | 57/47 |       |       |       |
| XA01  | 004267 | 58/04 | 58/09 | 58/21 |       |
| XA02  | 004303 | 59/06 |       |       |       |
| XA03  | 004321 | 59/18 |       |       |       |
| XA03A | 004336 | 59/32 |       |       |       |
| XA04  | 004350 | 59/46 |       |       |       |
| X801  | 004407 | 60/04 | 60/09 | 60/21 |       |
| X802  | 004423 | 61/06 |       |       |       |
| X803  | 004441 | 61/18 |       |       |       |
| XB03A | 004455 | 61/32 |       |       |       |
| XB04  | 004472 | 61/45 |       |       |       |
| XC01  | 004525 | 62/04 | 62/09 | 62/21 |       |
| XC02  | 004541 | 63/06 |       |       |       |
| XC03  | 004560 | 63/17 |       |       |       |
| XC03A | 004577 | 63/31 | 63/36 |       |       |
| XC04  | 004613 | 63/36 | 63/47 |       |       |
| X081A | 000134 | 64/04 | 64/09 | 64/21 |       |
| Z080A | 000042 | 67/03 | 38/19 | 38/51 |       |
|       |        | 7/06  | 34/34 | 35/05 |       |
|       |        | 38/20 | 38/32 | 39/44 |       |
|       |        | 10/29 | 10/30 | 49/19 |       |
|       |        | 9/22  | 16/54 | 36/05 | 36/41 |
|       |        | 11/20 | 11/26 | 50/05 | 50/23 |
|       |        |       |       | 49/37 | 50/23 |
|       |        |       |       | 38/43 |       |
|       |        |       |       | 35/33 |       |
|       |        |       |       | 38/51 |       |
|       |        |       |       | 38/19 |       |
|       |        |       |       | 64/21 |       |
|       |        |       |       | 64/09 |       |
|       |        |       |       | 63/47 |       |
|       |        |       |       | 63/36 |       |
|       |        |       |       | 63/31 |       |
|       |        |       |       | 63/17 |       |
|       |        |       |       | 62/04 |       |
|       |        |       |       | 61/45 |       |
|       |        |       |       | 61/32 |       |
|       |        |       |       | 61/18 |       |
|       |        |       |       | 61/06 |       |
|       |        |       |       | 60/04 |       |
|       |        |       |       | 59/46 |       |
|       |        |       |       | 59/32 |       |
|       |        |       |       | 59/18 |       |
|       |        |       |       | 59/06 |       |
|       |        |       |       | 58/04 |       |
|       |        |       |       | 57/47 |       |
|       |        |       |       | 57/34 |       |
|       |        |       |       | 57/20 |       |
|       |        |       |       | 57/08 |       |



LISTING

096-000224-03

PROGRAM

EXERCISER FOR ECLIPSE  
PART 6

TAPE

095-000245-03

ABSTRACT

'ECLIPSE31' IS AN EXERCISER PROGRAM USED TO TEST THE RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF THE ECLIPSE COMPUTER. 'ECLIPSE31' EXERCISES THE DOUBLE WORD INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES OF ITS RELIABLE OPERATION.



```

01 0801 ECL31 MACRO REV 04.00      00116133 12/31/76
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

```

```

10082 ECL31
02
03
04
05
06
07

```

```

.TITL ECL31
.ECLIPSE31
.ECLIPSE31 - CONTINUATION OF ECLIPSE30
.PART 8 OF EXERCISER FOR ECLIPSE

```

```

*****
NAME: ECLIPSE31.SR      PART NUMBER: 894-888644
*****
DESCRIPTION: ECLIPSE EXERCISER, PART 8
*****
REVISION HISTORY:
REV.      DATE
00      12/28/74
01      04/11/75
02      08/08/76
03      12/31/76
*****
COPYRIGHT (C) DATA GENERAL CORPORATION, 1974, 1975, 1976
ALL RIGHTS RESERVED.
*****

```

10883 ECL31

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

EXERCISER FOR ECLIPSE1 PART 8
-----
11. PROGRAM NAME
12. ECLIPSE31
13. GENERAL DESCRIPTION
14. ECLIPSE31 IS AN EXERCISER PROGRAM USED TO TEST THE
15. RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF
16. THE ECLIPSE COMPUTER. 'ECLIPSE31' EXERCISES THE DOUBLE
17. WORD INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES
18. OF ITS RELIABLE OPERATION.
19.
20. THE INSTRUCTIONS EXERCISED ARE AS FOLLOWS:
21. ELET,ELDA,ESTA,EJMP AND EJSR
22.
23. LOCATIONS 288 TO 295 IN PAGE 8 ARE FIXED FOR ECLIPSE31
24. PROGRAM.
25. LOCATION 283 KEEPS TRACK OF NUMBER OF PASSES RUN
26. THROUGH ECLIPSE31 PROGRAM.
27. LOCATION 281 KEEPS TRACK OF THE TEST RUNNING AT
28. PRESENT AND IS USEFUL FOR DEBUG WHEN LOOPING
29. OCCURS IN THE PROGRAM.
30. LOCATION 282 CONTAINS THE STARTING ADDRESS OF
31. ECLIPSE31 PROGRAM.
32. LOCATION 288 IS USED BY DTOS.
33. LOCATION 284 KEEPS TRACK OF INTERNAL PASS COUNT
34. WHICH IS FIXED BY LOCATION 285.
35.
36. FIRST PASS THROUGH ECLIPSE31 TEST WILL RUN SUPERFAST.
37. NEXT PASSES WILL RUN SLOWER AS EACH TEST IS RUN SEVERAL
38. TIMES TO RUN ALL RANDOM NUMBER COMBINATIONS.
39.
40. MACHINE REQUIREMENTS
41. -----
42.
43. ECLIPSE PROCESSOR
44. 4K READ-WRITE MEMORY
45. 15.2
46. CONSOLE EQUIPMENT
47.

```

10884 ECL31

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

SWITCH SETTINGS
-----
14. THIS PROGRAM USES DATA SWITCHES AS FOLLOWS
14.1
14.2
14.3
14.4

SW*0" - USE CONTENTS OF "SWREG" IF 0
SW*1" - LOOP ON FAILING TEST IF 0
SW*2" - PROCEED TO NEXT TEST IF 1
SW*3" - INHIBIT PRINTING TO TTY IF 1
SW*4" - DO NOT PRINT X ERRORS IF 0
SW*5" - PRINT FAILURE RATE IF 0
SW*6" - INHIBIT PRINTING PASS COUNT IF 1
SW*7" - INHIBIT OUTPUT TO LINE PRINTER IF 0
SW*8" - OUTPUT TO LINE PRINTER IF 1

EDTOS PRESUMES THAT ALL SWITCHES ARE 0. RETURN IS
MADE TO THE OPER SYS UPON AN ERROR AFTER FIRST PRINTING
THE FAILURE REPORT. IN STAND ALONE MODE, THE OPERATOR
IS ASKED TO PRESET THESE SWITCHES (THE MACHINE HALTS
FOR HIM TO DO SO) AND PRESS CONTINUE. THIS PRESET VALUE
CAN BE CHANGED AT ANY TIME BY SETTING DATA SWITCH "0"
TO A ONE. THE PROGRAM WILL THEN FOLLOW ACTUAL SETTINGS.

STAND ALONE STARTING ADDRESS = 280
IF 'CAT1' OR 'KITTEN' WAS LOADED FROM DTOS AND RESTART
WAS NEEDED, THEN USE AS FOLLOWS:
STARTING ADDR = 178 (FOR START WITH NO 'CAT1')
STARTING ADDR = 171 (FOR START WITH 'CAT1')

MONITOR LOCATION 283 TO CHECK THE CURRENT PASS COUNT

MONITOR LOCATION X8888 TO MAKE SURE THAT 'CAT1' OR
'KITTEN' IS RUNNING. IN CASES WHERE PROGRAM IS
STARTED WITH 'CAT1' OR 'KITTEN' LOCATION X8888 WILL SHOW
A PATTERN CHANGING FROM ZEROS) TO ALL ONES
TO AN INC/SNAP PATTERN.

(X= THE NUMBER OF THE HIGHEST MEMORY MODULE IN THE
SYSTEM AND MAY BE A VALUE 6 - 7)

```

18885 ECL31

```

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

OPERATING PROCEDURE/OPERATOR INPUT
-----
LOAD THE PROGRAM VIA THE BINARY LOADER OR INSERT A
PRELOADED MEMORY MODULE.
SET SWITCHES TO 20%.
PRESS START.
PROGRAM WILL HALT AFTER PRINTING THE MESSAGE
*SET DATA SWITCHES AND PRESS CONTINUE*.
SET REQUIRED SWITCH SETTINGS AND PRESS CONTINUE.
THE PROGRAM WILL RUN UNTIL MANUALLY STOPPED. IN CASE
OF MALFUNCTIONING, THE PROGRAM WILL PRINT ERROR
MESSAGE AND TAKE APPROPRIATE ACTION AS PER THE SN
SETTINGS.

PROGRAM OUTPUT/ERROR DESCRIPTION
-----
FOR ANY ERRORS DETECTED, THE PROGRAM WILL PRINT ERROR
REPORT OR X FAILURES DEPENDING UPON THE SN SETTINGS.
ERROR REPORT CONSISTS OF ALL ACCUMULATORS, CARRY,
RELOCATED PROGRAM COUNTER OF THE TEST THAT IS FAILING
AND PC IN THE LISTING AT THE TIME OF FAILURE.
THE PROGRAM WILL LOOP IN THE TEST THAT IS FAILING IF
SN#1X IS 0.
THE PRINTING OF ERROR REPORT CAN BE ABORTED BY SETTING
SW#2 TO 1.
IF LOOPING OCCURS IN THE PROGRAM, STOP THE COMPUTER
AND CHECK LOCATION 201 TO FIND OUT THE TEST THAT WAS
RUNNING BEFORE THE LOOPING OCCURRED.

```

18886 ECL31

```

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

PROGRAM DESCRIPTION/THEORY OF OPERATION
-----
EACH TEST IS COMPLETE IN ITSELF, SO THE PROGRAM CAN
BE STARTED FROM ANY TEST WITHOUT CAUSING ANY
INITIALIZATION ERRORS.
WHEN 'ECLIPSE31' IS STARTED AT LOCATION 200 OR BY
OTDS, IT WILL SIZE UP THE MEMORY AND WILL PRINT
THE TOP OF THE MEMORY.
AFTER SETTING UP THE SWITCHES AND PRESSING CONTINUE,
THE EXERCISER WILL RUN THE FIRST PASS VERY FAST. IN
THE FIRST PASS EACH TEST IS RUN ONLY ONCE, ALL OTHER
PASSES WILL TAKE MORE TIME AS EACH TEST IS RUN
ACCORDING TO THE TEST COUNT SPECIFIED IN EACH TEST.
AFTER THE 1ST PASS, ECLIPSE31 IS RELOCATED IN THE
AVAILABLE MEMORY FOR ALL NEXT PASSES AND THE AREA
BELOW AND ABOVE THE RELOCATED PROGRAM IS USED
FOR SCRATCH BUFFER AREA. REFER TO THE LISTING TO
FIND OUT THE INFORMATION ABOUT EACH TEST.

RESTRICTIONS/HISC
-----
CERTAIN INSTRUCTIONS LIKE BLM, XET, BAN, ETC..
DO ALLOW INTERRUPTS TO OCCUR DURING THEIR
EXECUTION. THIS FEATURE OF THOSE INSTRUCTIONS IS
NOT CHECKED IN THIS TEST.

```







10011 ECL31

```

01 .MACRO
02 ELEFT
03 SETUP
04 RAND
05 ANDI 0,LFGA1+1
06 STA +1
07 JBR +26,,3
08 ADDI 10000,3
09 IORI 3,LEFLOC
10 STA 1,,3
11 ADI 3,LEFLC
12 STA 177,0
13 ANDI 0,,42
14 MOV 1,LEFLOC,0
15 ELEFT
16 SUB 0,1
17 IORI 10000,1
18 STA 1,,+2
19 ELEFT
20 LDA 1,LFGA1+1
21 SUBM 0,,32R
22 ERROR
23 LOOP
24 JNP
25 LFGA1: 0
26
27

```

10012 ECL31

```

01 .MACRO
02 ELEFT
03 ELEFS A1,A2,A3,A4
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

```

```

10012 ECL31
01 .MACRO
02 ELEFT
03 ELEFS A1,A2,A3,A4
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

```

x

x

18013 ECL31

```

01 .MACRO TESTLOC
02 JSR S,HELP
03 STA
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

```

18014 ECL31

```

01 .MACRO ESTA2
02 TESTLOC
03 SETUP 180.
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19

```

```

01 STA 0,S,STLCR
02 MOV 0,S
03 LDA 0,S,STLCR
04 MOVZR 0,S,1,S7C
05 ADI 1,S,1
06 HLV 0
07 STA 0,S,2
08 ESTA 1,0,S,1
09 LDA 0,S,STLCR
10 SUB# 0,S,STLCR
11 ERROR
12 LOOP

```

```

01 ?C(STLCR)RNDM ADDR
02 ?AC1= RNDM DATA
03 ?AC1=(RNDM ADDR/2)+BIT 15
04 ?AC0=RNDM ADDR/2
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19

```

x

x

18015 ECL31

```

01 .MACRO ESTAS
02 TSTLOC
03 SETUP
04 A1
05 STA 0,STLCL
06 A2 1,STLCL
07 LDA 1,STLCL
08 SUBM 1,0,SNR
09 JMP A3
10 STA 0,STLCL
11 RAND
12 MOV 0,1
13 LDA 0,STLCL
14
15
16
17 .MACRO ESTA4
18 ESTAS
19 MOVZR
20 ADI 1,A4
21 HLV 0
22 IORI 10000,0
23 STA 0,+3
24 STA 0,+4
25 STA 1,0,+4
26 ELDA 0,0,+4
27 SUBM 1,0,SNR
28 ERROR
29 LOOP
30

```

18016 ECL31

```

01 .MACRO ESTAS
02 TSTLOC
03 SETUP
04 RAND
05 100.
06 STA 0,STSV0
07 LOBFAD
08 STA 0,STLCL
09 LOBFAD
10 LDA 1,STLCL
11 SUBM 0,1,SNR
12 JMP A2
13 ADDR 0,0
14 STA 0,STLCL
15 NISFAD
16 MOV 0,2
17 LDA 3,STLCL
18 STA 3,0,2
19 ADDR 3,3
20 LDA 2,STLCL
21 STA 2,0,3
22 MOVZR 0,1,STC
23 ADI 1,+1
24 HLV 0
25 ADDR 0,0
26 STA 0,+3
27 LDA 1,STSV0
28 ESTA 1,0,+1
29 LDA 0,STLCL
30 SUBM 0,1,SNR
31 ERROR
32 LOOP
33

```

```

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

```

X

X

10017 ECL31

```

01 .MACRO
02 EJMPL
03 TSTLOC
04 JSR
05 #FILL
06 100.
07
08 MOV
09 LDA
10 STA
11 ELEM
12 STA
13 STA
14 A1
15 ERROR
16 LOOP
17
18 X

```

```

;FILL SCRATCH AREA WITH
;JSR CENTER

;ACB=AC2=RANDOM ADDR
;(JMP #JMLC1) IS STORED IN
RANDOM ADDRESS.
;ADDR OF (A1+3) IS
STORED IN JMLC1 FOR
;PROPER RETURN
;

```

10016 ECL31

```

01 .MACRO
02 EJMPL
03 TSTLOC
04 JSR
05 #FILL
06 100.
07
08 MOV
09 LDA
10 STA
11 ELEM
12 STA
13 STA
14 A1
15 ERROR
16 LOOP
17
18 X

```

```

;FILL SCRATCH AREA WITH
;JSR CENTER

;ACB=AC3=RANDOM ADDRESS
;STORE (JMP #JMLC1) IN
RANDOM ADDRESS
;ADDR OF (A1+3) IS STORED
IN JMLC1 FOR PROPER RETURN
;RANDOM ADDR. PC OF (A1+1)
IS STORED IN A1
;INSTRUCTION
;

```

10019 ECL31

\*MACRO

```

01 EJMPS
02 TSTLOC
03 JSR
04 #JFILL
05 100,
06 A2
07 MOV
08 LDA
09 STA
10 STA
11 STA
12 STA
13 STA
14 STA
15 STA
16 STA
17 STA
18 STA
19 STA
20 STA

```

```

IFILL SCRATCH AREA WITH
JSR #ENTER
FACB=ACB#RNDM ADDRESS
FJMP #JMLC1 IS STORED
FIN RNDM ADDRESS
FADDR OF (A1*3) IS STORED
FIN JMLC1 FOR GOOD RETURN
FRNDM ADDR/2)+BIT 15 IS
STORED IN ACA4 AND
FRNDM ADDR/2) IS STORED
FIN A1 INSTRUCTION

```

```

01 A31
02 ERROR
03 LOOP
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20

```

10020 ECL31

\*MACRO

```

01 EJMPS
02 TSTLOC
03 JSR
04 #JFILL
05 100,
06 A2
07 MOV
08 LDA
09 STA
10 STA
11 STA
12 STA
13 STA
14 STA
15 STA
16 STA
17 STA
18 STA
19 STA
20 STA
21 STA
22 STA
23 STA
24 STA
25 STA
26 STA
27 STA
28 STA
29 STA
30 STA
31 STA
32 STA
33 STA
34 STA
35 STA
36 STA

```

```

IFILL SCRATCH AREA WITH
JSR #ENTER
FACB=ACB#RNDM ADDR 1
FJMP #JMLC2
FIN RNDM ADDR 1)+RNDM ADDR 2)
FACB=ACB#RNDM ADDR 2
FSTORE (JMP #JMLC1) IN
FRNDM ADDR 2
FSTORE ADDR OF (A1*3) IN
FJMLC1 FOR GOOD RETURN
FACB#RNDM ADDR 1
FRNDM ADDR 1=PC OF A1*1)
FIS IN ACB
FJMP #JMLC2
FIN RNDM ADDR 1)+RNDM ADDR 2)
FIS STORED IN ACA5 AND
FACB#RNDM ADDR 1)/2
FADD THE INDIRECT BIT TO
FACB AND STORE IN A1

```

```

01 A31
02 ERROR
03 LOOP
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

```



18823 ECL31

```

01
02
03
04
05
06
07

```

```

*MACRO RNDADR
  RAND
  JSR A1
  A2
*

```

```

*GET ADDRESS IN THE RANGE
  IC(A1) AND C(A2)

```

18824 ECL31

```

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

```

```

* STANDARD MACROS
*
*MACRO LOOP
  JSR @ENTLO
*
*MACRO SETUP
  JSR @ENTIN
  A1
*
*MACRO RAND
  JSR @ENTRA
*
*MACRO CALL
  JSR @ICAL
  A1
*
*MACRO ERROR
  JMP *8
  JMP *9
  STA $JACS
  JSR @ENTER
*
*MACRO JMPER
  LDA $JITRER
  MOV $J,SJZR
  JMP A1
*

```

```

ITERATE TEST ROUTINE
INITIALIZE TEST.
IC(CAB)BRANDON #
ICALL SUBROUTINE A1
IF A ERROR PRESENT
  JMP TO A1

```



```

10025 ECL31
01
02
03 ***** DIAGNOSTIC PROGRAM PREAMBLE *****
04 .LOC 0
05 DIRT 0
06
07 .LOC 40
08
09
10 .LOC 45
11 EGGS
12
13 .ZREL
14 .BLK 8.
15
16
17
18
19
20 .LOC 0
21 .EGGS
22 .MEMPOT: 0
23 .ICAT: 0
24
25
26
27
28 .LOC 170
29 .OFF:
30 .CHK
31 .CHK
32
33 .JMP 0*1
34 .NMESS
35
36
37
38
39 .LOC 176
40 .BLK 2
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

```

```

01
02
03 ***** LOCAL ZREL *****
04 .END OF PROGRAM
05 .RELOCATION RANGE OF PROGRAM
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

```

```

01
02
03 ***** LOCAL ZREL *****
04 .END OF PROGRAM
05 .RELOCATION RANGE OF PROGRAM
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

```

```

10027 ECL31
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

J ROUTINE TO RELOCATE TEST PROGRAM
      RAM
      LDA 0,RELOC
      LDA 1,RS
      LDA 2,BANBLK
      LDA 3,BANBLK+8
      BAN
      CALL
      JBR #ICAL
      BEGIN
      J INITIAL LOCATIONS OF RELOCATING SUBROUTINES
      JTEST INITIALIZER
      JTEST TERMINATOR
      JERROR ROUTINE
      JRANDM NUMBER GEN.
      JRNDAD
      JTOP OF BOTTOM BUFFER
      JBUFF+99,
      JBUFF
      JMFIL
      J CORRECTED POINTERS TO RELOCATED SUBROUTINES
      JENTYD INIT
      JENTYD LOP
      JENTRI ERR
      JRAND
      JRNADRI
      JLOPFI
      JHIGFI
      JMFIL
      JTOP OF LOWER BUFFER
      JTOP OF UPPER BUFFER
      JMINLOC
      JMAXLOC

```

```

10028 ECL31
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

      .LOC 500
      RELOCATABLE LOWER BUFFER USED IN THE FOLLOWING TESTS
      *****
      LOGUFFS .BLK 100,
      *****
      *****SIZE SYSTEM & RESERVE MEMORY*****
      NSTRT: IORST
      JSR #ISIZE
      NEG 2,2
      COM 2,2
      STA 2,MENTOP
      ELDA 0,CATSH
      MOV 0,0,SNR
      JHP NOCAT
      ELDA 0,PRGEND
      LDA 1,#1777
      ADD 1,0
      ADC# 0,2,SNC
      JMP NOCAT
      SUB
      LDA 1,#400
      ADD 2,1
      STA 1,ICAT
      JMP STLOC
      LDA 1,#800
      SUB 1,2
      SUB 1,1
      ESTA 1,CATSH
      LDA 1,MINLOC
      SUB 1,2
      STA 2,MAXLOC

```

18080 ECL31

```

01
02
03
04 08700 08022 0AMESI JBR @INSS
05 08701 08158 MESIZ
06 08702 08401 LDA 1,HEMTOF
07 08703 08180 MOV @R
08 08704 08224 JBR @IPOCT
09 08705 08282 JBR @INSS
10 08706 08154 KRLF
11 08707 08400 SUB 1,1
12 08710 08420 STA 1,PASS
13
14 08711 08478 ELDA 1,AUTO
15 08712 08500 MOV 1,1,SR
16 08713 08504 JBR START
17 08714 08543
18
19 08715 08622 JBR @INSS
20 08716 08154 SETB @INSS
21 08717 08622 JBR
22 08720 08154 KRLF
23 08721 08307 HALT
24 08722 08451 JBR
25 08723 08477 READS
26 08724 08470 EDA
27 08725 08642 JBR START
28 08726 08648
29
30 08727 08400 STARTI
31 08728 08400 REGI
32 08730 08420 STA 3,PASS
33 08731 08424 STA 3,RELOC
34 08732 08427 LDA 1,MINLOC
35 08733 08520 LDA 2,MAXLOC
36 08734 08500 ADD 1,2
37 08735 08110 SBI 1,2
38 08736 08234 STA 2,TTOP
39 08737 08236 STA 2,MIBF
40 08740 08424 LDA 1,BUFF
41 08741 08435 STA 1,LOWB
42 08742 08286 JBR BANDY+1

```

18080 ECL31

```

01
02
03
04
05
06 08743 08400 INITI INC
07 08744 08420 STA
08 08745 08226 STA
09
10 08746 08177 LDA @1,3
11 08747 08206 STA @1,TR
12 08750 08207 STA @1,TRCT
13
14 08751 08014 LDA @,RELOC
15 08752 08400 SUB @,3
16 08753 08421 STA @,LISTING
17
18 08754 08400 SUB @,3
19 08755 08421 STA @,ITR
20 08756 08421 STA @,ITREC
21
22 08757 08420 LDA @,PASS
23 08758 08504 MOV @,3,SR
24 08761 08040 JMP INITI
25
26 08762 08200 SUBZL
27 08763 08400 STA @,3,TR
28 08764 08420 STA @,ITRCT
29
30 08765 08226 INITI1 LDA @,ACB
31 08766 08226 JMP @ITRET

```

\*\*\*\*\*TEST UTILITY SUBROUTINES\*\*\*\*\*

F SUBROUTINE TO INITIALIZE A TEST LOOP

```

TEST LOOP INITIALIZER
SAVE RETURN LOCATION
SAVE CONTENTS OF ACB
GET # OF ITERATIONS
SET ITER. VALUE
SET ITER. COUNT
COMPUTE AND SAVE
THE LISTING ADDRESS
FOR THIS TEST.
CLEAR ERROR SWITCH
CLEAR ERROR COUNT
TEST FOR FIRST PASS
THIS IS 1ST PASS
SET ITERATIONS FOR
11 LOOP ONLY.
RESTORE ACB AND
EXIT TO TEST

```

\*\*\*\*\*OUTPUT STRT MESSAGE & READ SWITCHES\*\*\*\*\*

PRINT SIZE OF MEMORY

```

PRINT PASS COUNT
PRUNING IN AUTO MODE
IVES START PROGRAM.
NO, PRINT SET SWITCHES MESS.
READ NEW STATE OF SWITCHES
CLEAR PASS COUNT
SET RELOCATION TO 8
STORE TOPMOST USABLE
ADDRESS IN TSTOP
INITIALISE TOP OF UPPER BUFF
INITIALISE BOTTOM OF LOWER
BUFFER FOR INITIAL PASS
AND START THE PROGRAM

```

18031 ECL31

```

1  SUBROUTINE TO TERMINATE A TEST LOOP
2
3  LOP1: STA 3,LOPRET
4  D&Z LOP2
5  J&P LOP3
6  00771 000448
7  00772 034218
8  00773 175005
9  00774 002213
10 00775 034206
11 00776 004207
12
13 00777 074477 LOP1: READS
14 01000 175112 MOVLM
15 01001 000403 JMP +3
16 01002 136470 ELDA
17
18 01004 177100 ADDL
19 01005 177100 ADDL
20 01006 000421 JMP
21 01007 040226 STA 0,AC0
22 01010 044227 STA 1,AC1
23 01011 000230 STA 2,AC2
24 01012 000222 J&R @INSS
25 01013 001400 SUB PERCENT
26 01014 100400 STA
27 01015 004211 LDA
28 01016 042211 STA
29 01017 000224 LDA
30 01020 143710 MUL
31 01021 030206 LDA
32 01022 153710 DIV
33 01023 000225 J&R
34 01024 200226 LDA
35 01025 024227 LDA
36 01026 030230 LDA
37 01027 176400 LOP2: SUB
38 01030 004211 STA
39
40 01031 034210 LOP3: LDA
41 01032 175004 MOV
42 01033 074477 READS
43 01034 175112 MOVLM
44 01035 000403 JMP +3
45 01036 136470 ELDA
46 01037 000300 ADDLM
47 01040 177115 JMP
48 01041 000201 PLOPRET
49 01042 002213

```

18032 ECL31

```

1  ERROR SUBROUTINE: SETS ITRER FLAG, PRINTS MACHINE STATE.
2
3  ERR: STA 3,ERRET
4  01043 054212 STA 0,AC0
5  01044 040226 SUBCL 0,0
6  01045 102560 STA
7  01046 040232 ISZ ITRC
8  01047 010211 LDA 0,ITRER
9  01050 000210 ADCZ# 0,3,SNR
10 01051 100033 JMP
11 01052 000451 ERR:
12
13 01053 054210 STA 3,ITRER
14 01054 044227 STA 1,AC1
15 01055 000230 STA 2,AC2
16 01056 000222 J&R @INSS
17 01057 001407 ERMSG
18 01060 024203 LDA 1,PASS
19 01061 100420 INCZ 1,1
20 01062 000225 J&R @IPDEC
21 01063 000222 J&R @INSS
22 01064 001501 HEADER
23 01065 101024 MOVZ 0,0
24 01066 101024 LDA 1,CRY
25 01067 000225 J&R @IPDEC
26 01070 101040 MOV0 0,0
27 01071 024226 LDA 1,AC0
28 01072 000224 J&R @IPOCT
29 01073 024227 LDA 1,AC1
30 01074 000224 J&R @IPOCT
31 01075 024230 LDA 1,AC2
32 01076 000224 J&R @IPOCT
33 01077 024231 LDA 1,AC3
34 01078 000224 J&R @IPOCT
35 01100 000224 LDA 1,AC1
36 01101 024212 LDA 0,RELOC
37 01102 020214 SUBZ 0,1
38 01103 100420 J&R @IPOCT
39 01104 000224 LDA 1,ERRET
40 01105 024212 J&R @IPOCT
41 01106 000224 J&R @INSS
42 01107 000222 J&R @INSS
43 01110 201544 KCRLF
44
45 01111 004227 LDA 1,AC1
46 01112 030230 LDA 2,AC2
47 01113 136470 ELDA 3,AUTO
48 01114 000226 MOV 3,3,SNR
49 01115 175005 JMP ERRI
50
51 01117 000277 IOWST
52 01118 034010 LDA 3,EGGS
53 01119 035404 LDA 3,4,3
54 01120 001400 JMP 0,3
55
56 01123 020226 ERRI: LDA 0,AC0
57 01124 034231 LDA 3,AC3
58 01125 000212 JMP ERRET

```

```

9033 ECL31
81
82
83
84 01106 02021
85 01137 02401
86 01130 10084
87 01131 00140
88 01132 10000
89 01133 10541
90 01134 10700
91 01135 10100
92 01136 10510
93 01137 10000
94 01140 02403
95 01141 10000
96 01142 04021
97 01143 00140
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

10034 ECL31
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

ALLOCATES MEMORY FOR COPIES
OF TEST PROGRAM. WORKS IN CONJUNCTION
WITH GAM ROUTINE IN PAGE ZERO, WHICH
ACTUALLY COPIES THE TEST PROGRAM TO
THE NEW LOCATION.

RELOCATE SUBROUTINE:
E18Z CATSN
JMP *2
JSR #CAT
RAND #ENTRA
JSR #ENTRA
LDA 1,MAXLOC
LDA 2,MINLOC
SUBZ# 2,1,8NC
JMP REL2
SUBZ 1,9,8ZC
JMP *1
ADD #1
SUBZ# 2,1,8NC
REL1
JMP REL
LDA #RELOC
SUBZ# 1,9,8ZC
REL1
JMP REL
SUBZ# 1,2,8NC
JMP *3
LDA 1,MAXLOC
SUB 0,0,8R#
ADD #1
REL1
STA 1,RELOC
ELDA 3,PRGEND
INC 3,3
STA 3,LONB#
SUB 3,3
XCH 0,3
XCH 2,1
XCH 3,3
XCH 3,3
CALL #ICAL
JBR #ICAL
BEGIN

RELOCATE THE PROGRAM...
IC(AC0)=RANDOM #
IF C(MINLOC)>C(MAXLOC),
MEMORY IS TOO SMALL FOR
RELOCATION.
IC(AC1) IS RANDOM MODULO
IC(MAXLOC).
IF C(AC1) IS < C(MINLOC)
USE C(MINLOC).
IC(AC0)=SOURCE. C(AC1)=DEST.
SOURCE < DESTINATION, MOVE DOWN
IC(AC1)=DISTANCE TO MOVE
IF PROGRAMS DON'T OVERLAP,
ITS OK TO MOVE UP
WHEN OVERLAP WILL OCCUR
MOVE SOURCE PROGRAM TO
TOP OF MEMORY
UPDATE LOW=8UFF POSITION.

ICG TO PAGE #
IFOR THE XFER.
ICALL SUBROUTINE BEGIN

```

10035 ECL31

01 /  
02 / RANDOM ADDRESS GENERATOR  
03 /

```

04 01235 844227 RNDMADI STA 1,AC1
05 01236 850230 2,AC2
06 01237 125200 MOVR 1,1
07 01238 84432 STA 1,CRY
08 01241 833481 LDA 2,013
09 01242 827460 LDA 1,0013
10 01243 132400 SUB 1,2
11 01244 124510 XCR 0,1
12 01245 164710 XCH 0,1
13 01246 153710 DIV 1,0013
14 01247 827480 LDA 1,0
15 01250 123600 ADD 1,0
16 01251 884932 LDA 1,CRY
17 01252 125100 MOVL 1,1
18 01253 824227 LDA 1,AC1
19 01254 828230 LDA 2,AC2
20 01255 861482 JHP 2,3

```

01 / ROUTINE TO FILL BUFFER WITH JSR CENTER

```

22 01257 824923 JMPILL
23 01258 824923 STA 3,JNEX
24 01259 838315 LDA 1,JMLC4
25 01260 838315 LDA 0,LOBRF
26 01261 834312 LDA 3,LOBFU
27 01262 156400 SUB 2,3
28 01263 174800 COM 3,3
29 01264 845800 STA 1,0,2
30 01265 151400 INC 2,2
31 01266 175404 INC 3,3,SZR
32 01267 800775 JMP 3,3
33 01268 838315 LDA 2,HIGBF
34 01269 838315 LDA 3,HISFU
35 01271 834316 SUB 2,3
36 01272 156400 COM 3,3
37 01273 174800 STA 1,0,2
38 01274 845800 LDA 2,2
39 01275 151400 INC 3,3,SZR
40 01276 175404 INC 3,3,SZR
41 01277 800775 JMP 3,3
42 01278 882481 JHP 0,JNEX
43 01300 880000 JNEXI 0

```

10036 ECL31

01 /  
02 / \*\*\*\*\*PRINT ROUTINES\*\*\*\*\*  
03 /

```

04 01302 175100 PDEC: MOVL 3,3
05 01303 854237 STA 3,PDERET
06 01304 175200 MOVR 3,3
07 01305 884441 JSR PDEC3
08 01306 823420 10000,
09 01307 801750 100,
10 01310 800144 10,
11 01311 800012 1,
12 01312 800001 0,
13 01313 800000 POC1: MOVL 3,3
14 01314 175100 STA 3,PDERET
15 01315 854237 MOVR 3,3
16 01316 175200 JSR PDEC3
17 01317 884427 100000,
18 01320 100000 10000,
19 01321 010000 1000,
20 01322 801000 100,
21 01323 888100 1,
22 01324 800010 1,
23 01325 800001 0,
24 01326 800000 PDEC2: LDA 0,011
25 01327 800022 LDA 2,01,2
26 01328 031377 MOVR 2,2,SNR
27 01329 151010 JMP PDEC3+1
28 01331 151010 SUBC 0,0
29 01332 800415 SUBC 0,0
30 01333 102450 JMP PDEC2
31 01334 146452 SUBZ 2,1,SZC
32 01335 800404 INC 0,0
33 01336 146450 INC 0,0
34 01337 101400 JMP 3,4
35 01338 888774 POC2: MOVRW 2,2,SZR
36 01341 151234 SUBC 2,2,SZC
37 01342 152462 LDA 2,500
38 01343 830021 ADD 2,0,SZR
39 01344 143084 INC 3,2,SMP
40 01345 171401 INC 3,2,SMP
41 01346 171401 JSR CHAR
42 01347 884423 MOV 2,3,SZR
43 01350 135004 JHP POC1
44 01351 800756 LDA 3,PDERET
45 01352 834237 MOVR 3,3
46 01353 175200 JMP 0,3
47 01354 801400 INC 3,3
48 01355 175400 STA 3,MESSR
49 01356 854848 LDA 2,1,3
50 01357 831777 LDA 0,2,377
51 01360 828880-HESS1: LOA 1,0,2
52 01361 825000 INCZ 2,2
53 01362 151488 AND 1,0
54 01363 123400 SUBS 0,1
55 01364 166700 JSR CHAR
56 01365 884405 MOV 1,0,SNR
57 01366 131005 JMP MESRET
58 01367 882440 JSR CHAR
59 01370 884402

```

```

PDECESS WITH TAB
EXIT, ALL DIGITS PRINTED
FORM THE DIGIT
AND SET C(CARRY)
SKIP IF LAST DIGIT
SKIP IF ZERO SUPPRESS
MAKE COUNT INTO ASCII
PRINT
SKIP IF TAB EXIT
NEXT DIGIT
MESSAGE PRINTER
JSR (MESS)
MESSAGE ADDRESS

```



0039 ECL31  
01 01453 000746  
02  
03 01454 000000 CHR8V: 0

CHAR1

JMP

ITEMP SAVE FOR AC2

10040 ECL31

```
01  
02  
03  
04 01455 000215 PASHES: .TXTE I<15><12>PASS I  
05 000500  
06 051523  
07 150240  
08 000000  
09 01462 000215 PERCENT .TXTE I<15><12>X FAIL=I  
10 120245  
11 040706  
12 146311  
13 000275  
14 01467 000215 ERMSG: .TXTE I<15><12>ERROR IN PASS: I  
15 000215  
16 151300  
17 147722  
18 120322  
19 047311  
20 000240  
21 051501  
22 030123  
23 000240  
24 01501 000215 HEADER .TXTE I<15><12>15><12>  
25 000215  
26 01503 151300 CRY AC0 AC1 AC2 AC3 LISTING LOGICAL<15><12>I  
27 000031  
28 141001  
29 120000  
30 040411  
31 130703  
32 004640  
33 141501  
34 120002  
35 040411  
36 031703  
37 140011  
38 051711  
39 144724  
40 043016  
41 140011  
42 043717  
43 141711  
44 140101  
45 000015  
46 000000  
47 01530 000215 MESIZ: .TXTE I<15><12>LAST LOGICAL ADDRESS=I  
48 040714  
49 152123  
50 140040  
51 043717  
52 141711  
53 140101  
54 040040  
55 042104  
56 142722  
57 051023  
58 000275  
59 01544 000215 KCRLF: .TXTE I<15><12>I  
60 000000
```



```

0041 ECL31
01 01546 142523 SETSN: JTXE ISET DATA SWITCHS AND PRESS CONTINUEI
02 040504
03 040504
04 040724
05 051600
06 144787
07 141784
08 051510
09 040640
10 042110
11 050240
12 142702
13 051623
14 141640
15 047317
16 144784
17 052516
18 000305

```

```

10042 ECL31
01
02
03
04
05
06
07
08
09 01570 100510 BEGIN: XOR 0,0
10
11
12
13
14
15
16
17 01571 006305
18 01572 006144
19
20
21 01573 006310
22 01574 006311
23 01575 000233
24 01576 000234
25 01577 101000
26 01500 126848
27 01601 131008
28 01602 155000
29 01603 040403
30 01604 102000
31 01605 102078
32 000000
33 01607 101003
34
35 01614 124014
36
37 01621 150014
38
39 01626 174014
40
41
42 01633 006306

```

```

; *****FIRST TEST*****
;
; TESTING "ELEF" = LONG 'LEF' INSTRUCTION
;
; FIRST, CHECK ELEF WITHOUT INDIRECT ADDRESSING
;
LEF00: ELEF3 0,0,SNC,ELEF
        SETUP 100.
        JSR #ENTIN
        RANDR ONE,TSTTOP
        RAND
        JSR #ENTRA
        JSR #ORNADR
        ONE
        TSTTOP
        MOV 0,0
        ADC0 0+123,0+123
        MOV 0+123,0+223
        MOV 0+223,0+323
        STA 0,+03
        ADC 0,0
        ELEF 0,0,0
        MOV 0,0,SNC
        ERROR 0+123,0+123,SZR ?
        COM# 0+223,0+223,SZR ?
        ERROR 0+323,0+323,SZR ?
        COM# 0+323,0+323,SZR ?
        ERROR
        JSR #ENTLO

```

```

; INITIALIZE TEST.
;
;C(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(ONE) AND C(TSTTOP)
;
; ALL AC'S ARE SET TO 177777
; RAND CARRY IS INITIALISEN
;BY ADC0
;
; ELEF MUST NOT CHANGE ANY AC'S
;OR STATE OF CARRY EXCEPT ACP

```

```

ITERATE TEST ROUTINE

```

```

10043 ECL31
01
02
03 01034 006305
04 01035 000144
05
06
07 01036 000310
08 01037 000311
09 01040 000233
10 01041 000234
11 01042 100000
12 01043 100000
13 01044 100000
14 01045 101000
15 01046 044003
16 01047 100000
17 01050 100070
18 01052 100000
19 01052 100000
20
21 01057 10014
22
23 01064 174014
24
25 01071 100014
26
27
28 01076 006306

```

```

LEPA1:
ELEF3 1,Z,SZC,ELEF
SETUP 100
JSR #ENTIN
100
RNDADR ONE,TSTTOP
RAND
JSR #ENTRA
JSR #RNADR
ONE
TSTTOP
MOV 0,1
ADCC 1+103,1+103
MOV 1+103,1+203
MOV 1+203,1+303
STA 1,0+0
ADC 1,1
ELEF 1,0,0
FOR STATE OF CARRY EXCEPT AC1
MOV 1,1,0ZC
ERROR 1+103,1+103,SZR I
COMM 1+203,1+203,SZR I
ERROR 1+203,1+203,SZR I
COMM 1+303,1+303,SZR I
ERROR 1+303,1+303,SZR I
COMM
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE

```

```

10044 ECL31
01
02
03 01077 006305
04 01700 000144
05
06
07 01701 000310
08 01702 000311
09 01703 000233
10 01704 000234
11 01705 110000
12 01706 170040
13 01707 101000
14 01710 100000
15 01711 050403
16 01712 100000
17 01713 170070
18 01715 000000
19 01715 101003
20
21 01722 174014
22
23 01727 100014
24
25 01734 124014
26
27
28 01741 006306

```

```

LEPA2:
ELEF3 2,0,SNC,ELEF
SETUP 100
JSR #ENTIN
100
RNDADR ONE,TSTTOP
RAND
JSR #ENTRA
JSR #RNADR
ONE
TSTTOP
MOV 0,2
ADCC 2+103,2+103
MOV 2+103,2+203
MOV 2+203,2+303
STA 2,0+0
ADC 2,2
ELEF 2,0,0
FOR STATE OF CARRY EXCEPT AC2
MOV 2,2,SNC
ERROR 2+103,2+103,SZR I
COMM 2+203,2+203,SZR I
ERROR 2+203,2+203,SZR I
COMM 2+303,2+303,SZR I
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE

```

```

10045 ECL31
01
02
03 01077 006305
04 01700 000144
05
06
07 01701 000310
08 01702 000311
09 01703 000233
10 01704 000234
11 01705 110000
12 01706 170040
13 01707 101000
14 01710 100000
15 01711 050403
16 01712 100000
17 01713 170070
18 01715 000000
19 01715 101003
20
21 01722 174014
22
23 01727 100014
24
25 01734 124014
26
27
28 01741 006306

```

```

18045 ECL31          LEFAS1          ELEFS          3,2,SZC,ELEF
01          SETUP          100.
02          JSR #ENTIN
03 01742 086300
04 01743 080144
05          RAND
06          RNDADR ONE,TSTTOP
07 01744 086310          JSR #ENTRA
08 01745 086311          JSR #RNADR
09 01746 088233          ONE
10 01747 088234          TSTTOP
11 01750 119800          MOV          0,3
12 01751 182800          ADCZ          3+183,3+183
13 01752 185800          MOV          3+183,3+283
14 01753 131000          MOV          3+283,3+383
15 01754 054403          STA          3,,+3
16 01755 178800          ADC          3,0,0
17 01756 176070          ELEF          3,0,0
18 01760 175002          MOV          3,3,SZC
19          ERROR
20          COM#          3+183,3+183,SZR I
21 01765 180014          ERROR
22          COM#          3+283,3+283,SZR I
23 01772 124014          ERROR
24          COM#          3+383,3+383,SZR I
25 01777 158014          ERROR
26          LOOP
27          JSR #ENTLO          IITERATE TEST ROUTINE
28 02004 086300

```

INITIALIZE TEST.

IC(AC0)=RANDOM #  
GET ADDRESS IN THE RANGE  
IC(ONE) AND C(TSTTOP)  
FALL ACIS ARE SET TO 17777  
RAND CARRY IS INITIALIZED  
BY ADCZ  
TELEF MUST NOT CHANGE ANY AC'S  
FOR STATE OF CARRY EXCEPT ACS

```

18046 ECL31          LEP801          ELEF1          0
01          SETUP          4000.
02          JSR #ENTIN
03 02005 086305          INITIALIZE TEST.
04 02006 087840          RAND
05          JSR #ENTRA
06 02007 086310          ANDI          77777,0
07 02010 143770          MOV          0,0
08 02012 081900          ADC          0+183,0+183
09 02013 126000          STA          0,,+3
10 02014 048403          XCH          0+183,0+183
11 02015 104710          YCM          0+183,0+183
12 02016 102078          ELEF          0,0,0
13          SUBW          0,0+183,SZR
14          ERROR
15 02020 106414          LOOP
16          JSR #ENTLO          IITERATE TEST ROUTINE
17
18 02025 086300

```

IC(AC0)=RANDOM #  
INITIALLY.  
RANDOM ADDR (5OFFSET)  
PAPER ELEF,AC(0) MUST BE =  
PRNDM ADDR (5OFFSET)  
ITERATE TEST ROUTINE

```

10048 ECL31
01
02 LEF81: LEF81 2
   SETUP 4000.
   JSR #ENTIN
   4000.
   RAND
   JSR #ENTRA
   ANDI 77777,0
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
LEF82:
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
ELEF1 2
SETUP 4000.
JSR #ENTIN
4000.
RAND
JSR #ENTRA
ANDI 77777,0
MOV 0,2
ADC 2+183,2+183
STA 2,+3
XCH 2,2+183
ELEF 2,0,0
SUB# 2,2+183,SZR
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE
INITIALIZE TEST.
IC(AC0)=RANDOM #
INITIALLY,
FAC(2)=17777 AND
FAC(1)=RANDOM ADDR (=OFFSET)
AFTER ELEF,AC(2) MUST BE =
FRNDM ADDR (=OFFSET)
ITERATE TEST ROUTINE

```

```

10047 ECL31
01
02 LEF81: LEF81 1
   SETUP 4000.
   JSR #ENTIN
   4000.
   RAND
   JSR #ENTRA
   ANDI 77777,0
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
LEF82:
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
ELEF1 1
SETUP 4000.
JSR #ENTIN
4000.
RAND
JSR #ENTRA
ANDI 77777,0
MOV 0,1
ADC 1+183,1+183
STA 1,+3
XCH 1,1+183
ELEF 1,0,0
SUB# 1,1+183,SZR
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE
INITIALIZE TEST.
IC(AC0)=RANDOM #
INITIALLY,
FAC(1)=17777 AND
FAC(1)=RANDOM ADDR (=OFFSET)
AFTER ELEF,AC(1) MUST BE =
FRNDM ADDR (=OFFSET)
ITERATE TEST ROUTINE

```

```

10049 ECL31
01 LEF03:
02 ELEC1 3
03 SETUP 4000.
04 JSR #ENTIN
05 4000.
06 RAND.
07 JSR #ENTRA
08 ANDI 77777,R
09 MOV 0,3
10 ADC 3+163,3+163
11 STA 3,+3
12 XCH 3,3+163
13 ELEC1
14 SUBM 3,3+163,SZR
15 ERROR
16 LOOP
17 JSR #ENTLO
18 02110 006306

10050 ECL31
01 LEF08:
02 ELEC2 0,R
03 SETUP 2000.
04 JSR #ENTIN
05 2000.
06 RAND.
07 JSR #ENTRA
08 ANDI 77777,R
09 JSR +1
10 ADDI 0,+3
11 ADDL# 0,3,SZR
12 JMP LFCB
13 ADD 0,3
14 STA 0,+4
15 NOV 3,0+163
16 ADC 0,0
17 ELEC1
18 SUBM 0,0+163,SZR
19 ERROR
20 LOOP
21 JSR #ENTLO
22 02136 006306

INITIALIZE TEST.
;
;C(AC0)=RANDOM #
;
;
;AC(0)=17777
;
;AFTER EXECUTING ELEC1,
;AC(0) MUST BE ED. TO
;[RNDM ADDR +PC OF INSTRUCTION
;FOLLOWING ELEC1]
;
;
;
;
;ITERATE TEST ROUTINE
;

```

```

10049 ECL31
01 LEF03:
02 ELEC1 3
03 SETUP 4000.
04 JSR #ENTIN
05 4000.
06 RAND.
07 JSR #ENTRA
08 ANDI 77777,R
09 MOV 0,3
10 ADC 3+163,3+163
11 STA 3,+3
12 XCH 3,3+163
13 ELEC1
14 SUBM 3,3+163,SZR
15 ERROR
16 LOOP
17 JSR #ENTLO
18 02110 006306

INITIALIZE TEST.
;
;C(AC0)=RANDOM #
;
;
;INITIALLY,
;AC(3)=17777 AND
;AFTER ELEC1,AC(3) MUST BE =
;RNDM ADDR (=OFFSET)
;
;ITERATE TEST ROUTINE
;

```

```

10049 ECL31
01 LEF03:
02 ELEC1 3
03 SETUP 4000.
04 JSR #ENTIN
05 4000.
06 RAND.
07 JSR #ENTRA
08 ANDI 77777,R
09 MOV 0,3
10 ADC 3+163,3+163
11 STA 3,+3
12 XCH 3,3+163
13 ELEC1
14 SUBM 3,3+163,SZR
15 ERROR
16 LOOP
17 JSR #ENTLO
18 02110 006306

INITIALIZE TEST.
;
;C(AC0)=RANDOM #
;
;
;INITIALLY,
;AC(3)=17777 AND
;AFTER ELEC1,AC(3) MUST BE =
;RNDM ADDR (=OFFSET)
;
;ITERATE TEST ROUTINE
;

```

|             |              |        |                  |  |  |
|-------------|--------------|--------|------------------|--|--|
| 10051 ECL31 |              | LEPC1: | ELEF2 1,1        |  |  |
| 01          |              |        | SETUP 2000.      |  |  |
| 02          |              |        | JSR #ENTR        |  |  |
| 03          | 02137 006305 |        | RAND             |  |  |
| 04          | 02140 003720 |        | ANDI 77777,R     |  |  |
| 05          | 02141 006310 | LFC1:  | JSR #ENTR        |  |  |
| 06          | 02142 133770 |        | ANDI 77777,R     |  |  |
| 07          | 02144 054401 |        | JSR +1           |  |  |
| 08          | 02145 177770 |        | ADDI 0,13        |  |  |
| 09          | 02147 177112 |        | ADDL* 0,3,52C    |  |  |
| 10          | 02150 000771 |        | JMP LFC2         |  |  |
| 11          | 02151 117000 |        | ADD 0,3          |  |  |
| 12          | 02152 040404 |        | STA 0,1+4        |  |  |
| 13          | 02153 171000 |        | MOV 3,1+183      |  |  |
| 14          | 02154 100000 |        | ADC 1,1          |  |  |
| 15          | 02155 100000 |        | ELEF 1,0,1       |  |  |
| 16          | 02157 132414 |        | SUB# 1,1+183,52R |  |  |
| 17          | 02164 006306 |        | ERROR            |  |  |
| 18          |              |        | LOOP             |  |  |
| 19          |              |        | JSR #ENTLO       |  |  |
| 20          |              |        |                  |  |  |
| 21          |              |        |                  |  |  |
| 22          |              |        |                  |  |  |
| 23          |              |        |                  |  |  |

|             |              |        |                  |  |  |
|-------------|--------------|--------|------------------|--|--|
| 10052 ECL31 |              | LEPC2: | ELEF2 2,2        |  |  |
| 01          |              |        | SETUP 2000.      |  |  |
| 02          |              |        | JSR #ENTR        |  |  |
| 03          | 02165 006305 |        | RAND             |  |  |
| 04          | 02166 003720 |        | ANDI 77777,R     |  |  |
| 05          | 02167 006310 | LFC2:  | JSR #ENTR        |  |  |
| 06          | 02170 133770 |        | ANDI 77777,R     |  |  |
| 07          | 02172 004401 |        | JSR +1           |  |  |
| 08          | 02173 177770 |        | ADDI 0,13        |  |  |
| 09          | 02175 177112 |        | ADDL* 0,3,52C    |  |  |
| 10          | 02176 000771 |        | JMP LFC2         |  |  |
| 11          | 02177 117000 |        | ADD 0,3          |  |  |
| 12          | 02200 040404 |        | STA 0,1+4        |  |  |
| 13          | 02201 175000 |        | MOV 3,2+183      |  |  |
| 14          | 02202 100000 |        | ADC 2,2          |  |  |
| 15          | 02203 172470 |        | ELEF 2,0,1       |  |  |
| 16          | 02205 000000 |        | SUB# 2,2+183,52R |  |  |
| 17          | 02206 156414 |        | ERROR            |  |  |
| 18          |              |        | LOOP             |  |  |
| 19          |              |        | JSR #ENTLO       |  |  |
| 20          |              |        |                  |  |  |
| 21          |              |        |                  |  |  |
| 22          |              |        |                  |  |  |
| 23          |              |        |                  |  |  |

|                                |  |  |  |  |  |
|--------------------------------|--|--|--|--|--|
| INITIALIZE TEST.               |  |  |  |  |  |
| PC(AC0)=RANDOM #               |  |  |  |  |  |
| IAC(1)=17777                   |  |  |  |  |  |
| IAC(2)=17777                   |  |  |  |  |  |
| AFTER EXECUTING ELEF,          |  |  |  |  |  |
| IAC(1) MUST BE EQ. TO          |  |  |  |  |  |
| I(RNDH ADDR +PC OF INSTRUCTION |  |  |  |  |  |
| IFOLLOWING ELEF)               |  |  |  |  |  |
| ITERATE TEST ROUTINE           |  |  |  |  |  |





10055 ECL31

01

02 02265 006305

03 02266 006144

04 02267 006310

05 02270 115000

06 02271 006310

07 02272 143770

08 02274 117110

09 02275 000774

10 02276 040404

11 02277 105000

12 02300 107000

13 02301 173470

14 02303 146414

15 02310 006306

LEP01

SETUP 100,

JSR #ENTIN

100,

RAND

JSR #ENTRA

MOV 0,3

RAND

JSR #ENTRA

ANDI 77777,0

ADDL# 0,3,0ZC

JMP LPE0

STA 0,0,04

MOV 3,1

ADD 0,1

ELEF 2,0,3

SUB# 0,1,0ZR

ERROR

LOOP

JSR #ENTLO

LPE01

IC(AC0)=RANDOM #

FAC0=RNDM #

IC(AC0)=RANDOM #

FAC0=RNDM #

IC(AC0)=RANDOM #

FAC0=RNDM #

IC(AC0)=RANDOM #

FAC0=RNDM #

IC(AC0)=RANDOM #

FAC0=RNDM #

IC(AC0)=RANDOM #

FAC0=RNDM #

IC(AC0)=RANDOM #

FAC0=RNDM #

IC(AC0)=RANDOM #

FAC0=RNDM #

IC(AC0)=RANDOM #

FAC0=RNDM #

10056 ECL31

01

02

03

04

05

06

07 02311 006305

08 02312 000144

09

10

11 02313 006310

12 02314 006311

13 02315 000233

14 02316 000234

15 02317 111000

16 02320 025000

17 02321 125112

18 02322 000771

19 02323 040245

20 02324 026245

21 02325 103770

22 02327 100000

23 02327 040403

24 02330 100000

25 02331 102070

26 02331 100000

27 02333 106414

28

29

30 02340 006306

LEP01

JSR #ENTIN

100,

RNDADR ONE,TSTTOP

RAND

JSR #ENTRA

JSR #RNADR

ONE

TSTTOP

MOV 0,2

LDA 1,0,2

MOVL# 1,1,0ZC

JMP LPE0

STA 0,LEFLOC

LOA 1,0,LEFLOC

ADDI 100000,0

STA 0,0,03

ADC 0,0

ELEF 0,0,0,0

SUB# 0,1,0ZR

ERROR

LOOP

JSR #ENTLO

LEP01

JSR #ENTIN

100,

RNDADR ONE,TSTTOP

RAND

JSR #ENTRA

JSR #RNADR

ONE

TSTTOP

MOV 0,2

LDA 1,0,2

MOVL# 1,1,0ZC

JMP LPE0

STA 0,LEFLOC

LOA 1,0,LEFLOC

ADDI 100000,0

STA 0,0,03

ADC 0,0

ELEF 0,0,0,0

SUB# 0,1,0ZR

ERROR

LOOP

JSR #ENTLO

LEP01

JSR #ENTIN

100,

RNDADR ONE,TSTTOP

RAND

JSR #ENTRA

JSR #RNADR

ONE

TSTTOP

MOV 0,2

LDA 1,0,2

MOVL# 1,1,0ZC

JMP LPE0

STA 0,LEFLOC

LOA 1,0,LEFLOC

ADDI 100000,0

STA 0,0,03

ADC 0,0

ELEF 0,0,0,0

SUB# 0,1,0ZR

ERROR

LOOP

JSR #ENTLO

LEP01

JSR #ENTIN

100,

RNDADR ONE,TSTTOP

RAND

JSR #ENTRA

JSR #RNADR

ONE

TSTTOP

MOV 0,2

LDA 1,0,2

MOVL# 1,1,0ZC

JMP LPE0

STA 0,LEFLOC

LOA 1,0,LEFLOC

ADDI 100000,0

STA 0,0,03

ADC 0,0

ELEF 0,0,0,0

SUB# 0,1,0ZR

ERROR

LOOP

JSR #ENTLO

LEP01

JSR #ENTIN

100,

RNDADR ONE,TSTTOP

RAND

JSR #ENTRA

JSR #RNADR

ONE

TSTTOP

MOV 0,2

LDA 1,0,2

MOVL# 1,1,0ZC

JMP LPE0

STA 0,LEFLOC

LOA 1,0,LEFLOC

ADDI 100000,0

STA 0,0,03

ADC 0,0

ELEF 0,0,0,0

SUB# 0,1,0ZR

ERROR

LOOP

JSR #ENTLO

LEP01

JSR #ENTIN

100,

RNDADR ONE,TSTTOP

RAND

JSR #ENTRA

JSR #RNADR

ONE

TSTTOP

MOV 0,2

LDA 1,0,2

MOVL# 1,1,0ZC

JMP LPE0

STA 0,LEFLOC

LOA 1,0,LEFLOC

ADDI 100000,0

STA 0,0,03

ADC 0,0

ELEF 0,0,0,0

SUB# 0,1,0ZR

ERROR

LOOP

JSR #ENTLO



|             |    |                   |                       |
|-------------|----|-------------------|-----------------------|
| 18056 ECL31 | 01 | SETUP 10P.        | INITIALIZE TEST.      |
|             | 02 | JSR #ENTIN        |                       |
|             | 03 | 100.              |                       |
|             | 04 | RAND              |                       |
|             | 05 | RNDADR ONE,TSTTOP |                       |
|             | 06 | JSR #ENTRA        | PC(AC0)=RANDOM #      |
|             | 07 | #RNADR            |                       |
|             | 08 | TSTTOP            | PC(AC0)=RANDOM #      |
|             | 09 | JSR #ENTRA        | PC(AC0)=RANDOM #      |
|             | 10 | ONE               | PC(AC0)=RANDOM #      |
|             | 11 | TSTTOP            | PC(AC0)=RANDOM #      |
|             | 12 | LDA               | PC(AC0)=RANDOM #      |
|             | 13 | ADDI 14,,3        | PC(ONE) AND C(TSTTOP) |
|             | 14 | SUBZ# 3,1,SNC     | PC(AC0)=RANDOM #      |
|             | 15 | JMP LFF1          | PC(AC0)=RANDOM #      |
|             | 16 | LDA 1,0,3         | PC(AC0)=RANDOM #      |
|             | 17 | MOVL# 1,1,SZC     | PC(AC0)=RANDOM #      |
|             | 18 | JMP LFF1          | PC(AC0)=RANDOM #      |
|             | 19 | ADDI 100000,0     | PC(AC0)=RANDOM #      |
|             | 20 | STA 0,+,3         | PC(AC0)=RANDOM #      |
|             | 21 | LDA 1,0,3         | PC(AC0)=RANDOM #      |
|             | 22 | ELEF 3,00,1       | PC(AC0)=RANDOM #      |
|             | 23 | SUBW 3,1,SZR      | PC(AC0)=RANDOM #      |
|             | 24 | ERROR             | PC(AC0)=RANDOM #      |
|             | 25 | LOOP              | PC(AC0)=RANDOM #      |
|             | 26 | JSR #ENTLO        | PC(AC0)=RANDOM #      |
|             | 27 |                   | PC(AC0)=RANDOM #      |
|             | 28 |                   | PC(AC0)=RANDOM #      |
|             | 29 |                   | PC(AC0)=RANDOM #      |
|             | 30 |                   | PC(AC0)=RANDOM #      |
|             | 31 |                   | PC(AC0)=RANDOM #      |
|             | 32 |                   | PC(AC0)=RANDOM #      |
|             | 33 |                   | PC(AC0)=RANDOM #      |

|             |    |                   |                       |
|-------------|----|-------------------|-----------------------|
| 18057 ECL31 | 01 | SETUP 10P.        | INITIALIZE TEST.      |
|             | 02 | JSR #ENTIN        |                       |
|             | 03 | 100.              |                       |
|             | 04 | RAND              |                       |
|             | 05 | RNDADR ONE,TSTTOP |                       |
|             | 06 | JSR #ENTRA        | PC(AC0)=RANDOM #      |
|             | 07 | #RNADR            |                       |
|             | 08 | TSTTOP            | PC(AC0)=RANDOM #      |
|             | 09 | JSR #ENTRA        | PC(AC0)=RANDOM #      |
|             | 10 | ONE               | PC(AC0)=RANDOM #      |
|             | 11 | TSTTOP            | PC(AC0)=RANDOM #      |
|             | 12 | LDA               | PC(AC0)=RANDOM #      |
|             | 13 | ADDI 14,,3        | PC(ONE) AND C(TSTTOP) |
|             | 14 | SUBZ# 3,1,SNC     | PC(AC0)=RANDOM #      |
|             | 15 | JMP LFF1          | PC(AC0)=RANDOM #      |
|             | 16 | LDA 1,0,3         | PC(AC0)=RANDOM #      |
|             | 17 | MOVL# 1,1,SZC     | PC(AC0)=RANDOM #      |
|             | 18 | JMP LFF1          | PC(AC0)=RANDOM #      |
|             | 19 | ADDI 100000,0     | PC(AC0)=RANDOM #      |
|             | 20 | STA 0,+,3         | PC(AC0)=RANDOM #      |
|             | 21 | LDA 1,0,3         | PC(AC0)=RANDOM #      |
|             | 22 | ELEF 3,00,1       | PC(AC0)=RANDOM #      |
|             | 23 | SUBW 3,1,SZR      | PC(AC0)=RANDOM #      |
|             | 24 | ERROR             | PC(AC0)=RANDOM #      |
|             | 25 | LOOP              | PC(AC0)=RANDOM #      |
|             | 26 | JSR #ENTLO        | PC(AC0)=RANDOM #      |
|             | 27 |                   | PC(AC0)=RANDOM #      |
|             | 28 |                   | PC(AC0)=RANDOM #      |
|             | 29 |                   | PC(AC0)=RANDOM #      |
|             | 30 |                   | PC(AC0)=RANDOM #      |
|             | 31 |                   | PC(AC0)=RANDOM #      |
|             | 32 |                   | PC(AC0)=RANDOM #      |
|             | 33 |                   | PC(AC0)=RANDOM #      |

|             |    |                   |                       |
|-------------|----|-------------------|-----------------------|
| 18058 ECL31 | 01 | SETUP 10P.        | INITIALIZE TEST.      |
|             | 02 | JSR #ENTIN        |                       |
|             | 03 | 100.              |                       |
|             | 04 | RAND              |                       |
|             | 05 | RNDADR ONE,TSTTOP |                       |
|             | 06 | JSR #ENTRA        | PC(AC0)=RANDOM #      |
|             | 07 | #RNADR            |                       |
|             | 08 | TSTTOP            | PC(AC0)=RANDOM #      |
|             | 09 | JSR #ENTRA        | PC(AC0)=RANDOM #      |
|             | 10 | ONE               | PC(AC0)=RANDOM #      |
|             | 11 | TSTTOP            | PC(AC0)=RANDOM #      |
|             | 12 | LDA               | PC(AC0)=RANDOM #      |
|             | 13 | ADDI 14,,3        | PC(ONE) AND C(TSTTOP) |
|             | 14 | SUBZ# 3,1,SNC     | PC(AC0)=RANDOM #      |
|             | 15 | JMP LFF1          | PC(AC0)=RANDOM #      |
|             | 16 | LDA 1,0,3         | PC(AC0)=RANDOM #      |
|             | 17 | MOVL# 1,1,SZC     | PC(AC0)=RANDOM #      |
|             | 18 | JMP LFF1          | PC(AC0)=RANDOM #      |
|             | 19 | ADDI 100000,0     | PC(AC0)=RANDOM #      |
|             | 20 | STA 0,+,3         | PC(AC0)=RANDOM #      |
|             | 21 | LDA 1,0,3         | PC(AC0)=RANDOM #      |
|             | 22 | ELEF 3,00,1       | PC(AC0)=RANDOM #      |
|             | 23 | SUBW 3,1,SZR      | PC(AC0)=RANDOM #      |
|             | 24 | ERROR             | PC(AC0)=RANDOM #      |
|             | 25 | LOOP              | PC(AC0)=RANDOM #      |
|             | 26 | JSR #ENTLO        | PC(AC0)=RANDOM #      |
|             | 27 |                   | PC(AC0)=RANDOM #      |
|             | 28 |                   | PC(AC0)=RANDOM #      |
|             | 29 |                   | PC(AC0)=RANDOM #      |
|             | 30 |                   | PC(AC0)=RANDOM #      |
|             | 31 |                   | PC(AC0)=RANDOM #      |
|             | 32 |                   | PC(AC0)=RANDOM #      |
|             | 33 |                   | PC(AC0)=RANDOM #      |

|             |    |                   |             |    |                   |
|-------------|----|-------------------|-------------|----|-------------------|
| 10059 ECL31 | 01 |                   | 10060 ECL31 | 01 |                   |
|             | 02 | LEFF31            |             | 02 | SETUP 100.        |
|             | 03 | JSR #ENTIN        |             | 03 | JSR #ENTIN        |
|             | 04 | RAND              |             | 04 | RAND              |
|             | 05 | LFF31             |             | 05 | LEFF31            |
|             | 06 | JSR #ENTRA        |             | 06 | JSR #ENTIN        |
|             | 07 | STA 0,LF5V0       |             | 07 | STA 0,LF5V0       |
|             | 08 | RNDADR ONE,TSTTOP |             | 08 | RAND              |
|             | 09 | RAND              |             | 09 | JSR #ENTRA        |
|             | 10 | JSR #ENTRA        |             | 10 | ANDI 77777,0      |
|             | 11 | JSR #ENTRA        |             | 11 | ANDI 77777,0      |
|             | 12 | ONE               |             | 12 | STA 0,LF60+1      |
|             | 13 | TSTTOP            |             | 13 | JSR #+1           |
|             | 14 | LDA               |             | 14 | ADDI +16,,3       |
|             | 15 | JSR #LFSV0        |             | 15 | IORI 100000,3     |
|             | 16 | MOV 3,1           |             | 16 | STA 3,LEFLOC      |
|             | 17 | ADD 0,1           |             | 17 | ADI 1,3           |
|             | 18 | LDA 2,TSTTOP      |             | 18 | STA 3,LEFLC       |
|             | 19 | SUBZ# 1,2,5NC     |             | 19 | ELEF 2,0,LEFLOC,0 |
|             | 20 | JMP LFF3          |             | 20 | SUB# 2,0,SZR      |
|             | 21 | MOV 1,2           |             | 21 | ERROR             |
|             | 22 | LDA 1,0,2         |             | 22 | LOOP              |
|             | 23 | MOVL# 1,1,5ZC     |             | 23 | JSR #ENTILO       |
|             | 24 | JMP LFF3          |             | 24 | JMP +0,3          |
|             | 25 | ADDI 100000,0     |             | 25 | #LEFLC            |
|             | 26 | STA 0,,+3         |             | 26 | 0                 |
|             | 27 | MOV 1,0           |             | 27 | 0                 |
|             | 28 | ELEF 2,0,3        |             | 28 | 0                 |
|             | 29 | SUB# 2,0,SZR      |             | 29 | 0                 |
|             | 30 | ERROR             |             | 30 | 0                 |
|             | 31 | LOOP              |             | 31 | 0                 |
|             | 32 | JSR #ENTILO       |             | 32 | 0                 |
|             | 33 |                   |             | 33 | 0                 |

```

; INITIALIZE TEST.
; NOW IS THE TIME TO CHECK MULTI-INDIRECT ELEF
;
LEFF31
JSR #ENTIN
RAND
LEFF31
JSR #ENTIN
RAND
JSR #ENTRA
ANDI 77777,0
STA 0,LF60+1
JSR +1
ADDI +16,,3
IORI 100000,3
STA 3,LEFLOC
ADI 1,3
STA 3,LEFLC
ELEF 2,0,LEFLOC,0
SUB# 2,0,SZR
ERROR
LOOP
JSR #ENTILO
JMP +0,3
#LEFLC
0
0
0
0

```

```

; INITIALIZE TEST.
; GET ADDRESS IN THE RANGE
; C(ONE) AND C(TSTTOP)
;
JC(AC0)#RANDOM #
;
J(AC0)#RANDOM #
;
; MAKE SURE THE C(RNDM ADDR)
; DOES NOT INDIRECT
; IF YES-GET NEW RNDM NO
; SET INDIRECT BIT IN ELEF
;
PARTER EXECUTING ELEF,
FACE MUST BE #
IC(OFFSET+C(AC3))
ITERATE TEST ROUTINE

```

10061 ECL31

```
01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200
```

LEFG11

SETUP 100.

INITIALIZE TEST.

```
JSR %ENTIN  
100.  
RAND  
JSR %ENTRA  
ANDI 77777,0  
STA 0,LEFG1+1  
JSR +1  
ADDI 10,0,3  
IORI 100000,3  
STA 3,LEFLC  
ELEF 0,0,0,1  
LDA 1,LEFG1+1  
SUBM 0,1,5ZR  
LOOP  
ITERATE TEST ROUTINE  
JMP +*3  
0LEFLC  
0
```

LEFG21

SETUP 100.

INITIALIZE TEST.

```
JSR %ENTIN  
100.  
RAND  
JSR %ENTRA  
ANDI 77777,0  
STA 0,LEFG2+1  
JSR +1  
ADDI +20,0,3  
IORI 100000,3  
STA 3,LEFLC  
ADI 1,3  
STA 3,LEFLC  
ANDI 177,0  
MOV 0,2  
1,LEFLC,0  
ELEF  
SUB 0,1  
IORI 100000,1  
STA 1,0,2  
ELEF 0,0,0,2  
LDA 1,LEFG2+1  
SUBM 0,1,5ZR  
LOOP  
ITERATE TEST ROUTINE  
JMP +*3  
0LEFLC  
0
```

10062 ECL31

```
01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200
```

LEFG21

SETUP 100.

INITIALIZE TEST.

```
JSR %ENTIN  
100.  
RAND  
JSR %ENTRA  
ANDI 77777,0  
STA 0,LEFG2+1  
JSR +1  
ADDI +20,0,3  
IORI 100000,3  
STA 3,LEFLC  
ADI 1,3  
STA 3,LEFLC  
ANDI 177,0  
MOV 0,2  
1,LEFLC,0  
ELEF  
SUB 0,1  
IORI 100000,1  
STA 1,0,2  
ELEF 0,0,0,2  
LDA 1,LEFG2+1  
SUBM 0,1,5ZR  
LOOP  
ITERATE TEST ROUTINE  
JMP +*3  
0LEFLC  
0
```

10063 ECL31

```

01 LEFG3: ELEF 3,3
02 SETUP 100.
03 JSR #ENTIN.
04 RAND.
05 JACB#RANDOM #
06 ANDI 77777,0
07 JBR #ENTRA
08 STA 0,LEFG3+1
09 JBR +1
10 ADDI +26,+3
11 IORI 100000,3
12 STA 3,LEFLOC
13 ADI 1,3
14 STA 3,LEFLC
15 ANDI 177,6
16 MOV 0,3
17 ELEF 1,LEFLOC,0
18 SUB 0,1
19 IORI 100000,1
20 STA 1,+2
21 ELEF 0,+0,3
22 LDA 1,LEFG3+1
23 SUB# 0,1,SZM
24 LOOP
25 JBR #ENTLO
26 JMP +3
27 #LEFLC
28 0

```

10064 ECL31

```

01 / INITIALIZE TEST.
02 /
03 / MISC. TEST OF "ELEF"
04 /
05 /
06 LEFH01 SETUP 1000.
07 JSR #ENTIN
08 JBR +1
09 MOV 3,1
10 ELEF 2,2,1
11 ELEF 3,2,2
12 ELEF 0,10,+0
13 SUB 0,2
14 SUB# 2,1,SZR
15 LOOP
16 JBR #ENTLO
17 ITERATE TEST ROUTINE
18 /
19 /
20 /
21 /
22 /
23 /
24 /
25 /
26 /
27 /
28 /

```

10065 ECL31

```

01 INITIALIZE TEST.
02 /
03 /
04 /
05 /
06 /
07 /
08 /
09 /
10 /
11 /
12 /
13 /
14 /
15 /
16 /
17 /
18 /
19 /
20 /
21 /
22 /
23 /
24 /
25 /
26 /
27 /
28 /
29 /
30 /
31 /
32 /
33 /
34 /
35 /
36 /
37 /

```

|       |       |        |                   |  |                                  |
|-------|-------|--------|-------------------|--|----------------------------------|
| 10865 | ECL31 |        |                   |  |                                  |
| 01    |       | ELDA   | 1,0,SNC,ELDA      |  |                                  |
| 02    |       | ELEPS  | 1,0,SNC,ELDA      |  |                                  |
| 03    |       | SETUP  | 100,              |  |                                  |
| 04    |       | JSR    | 0ENTIN            |  | INITIALIZE TEST.                 |
| 05    |       | 100,   |                   |  |                                  |
| 06    |       | RNDADR | ONE,TSTTOP        |  |                                  |
| 07    |       | RAND   |                   |  |                                  |
| 08    | 02754 | JSR    | 0ENTRA            |  | IC(AC0)*RANDOM #                 |
| 09    | 02755 | 0RNADR |                   |  | IGET ADDRESS IN THE RANGE        |
| 10    | 02756 | ONE    |                   |  | IC(ONE) AND C(TSTTOP)            |
| 11    | 02757 | TSTTOP |                   |  |                                  |
| 12    | 02758 | MOV    | 0,1               |  |                                  |
| 13    | 02759 | ADCC   | 1,103,1,103       |  | IF ALL AC'S ARE SET TO 17777     |
| 14    | 02760 | MOV    | 1,103,1,0203      |  |                                  |
| 15    | 02761 | MOV    | 1,0203,1,0303     |  | IF AND CARRY IS INITIALISED      |
| 16    | 02762 | STA    | 1,0,03            |  | IF BY ADCC                       |
| 17    | 02763 | ADC    | 1,1               |  |                                  |
| 18    | 02764 | ELDA   | 1,0,0             |  | IF ELDA MUST NOT CHANGE ANY AC'S |
| 19    | 02770 | MOV    | 1,1,0NC           |  |                                  |
| 20    | 02771 | ERROR  |                   |  | FOR STATE OF CARRY EXCEPT AC1    |
| 21    | 02772 | COMW   | 1,103,1,103,0ZR   |  |                                  |
| 22    | 02773 | COMW   | 1,0203,1,0203,0ZR |  |                                  |
| 23    | 02774 | COMW   | 1,0303,1,0303,0ZR |  |                                  |
| 24    | 02775 | COMW   | 1,0303,1,0303,0ZR |  |                                  |
| 25    | 02776 | COMW   | 1,0303,1,0303,0ZR |  |                                  |
| 26    | 02777 | LOOP   |                   |  |                                  |
| 27    | 02778 | JSR    | 0ENTLO            |  | ITERATE TEST ROUTINE             |
| 28    | 02779 |        |                   |  |                                  |
| 29    | 02780 |        |                   |  |                                  |
| 30    | 02781 |        |                   |  |                                  |
| 31    | 02782 |        |                   |  |                                  |
| 32    | 02783 |        |                   |  |                                  |
| 33    | 02784 |        |                   |  |                                  |
| 34    | 02785 |        |                   |  |                                  |
| 35    | 02786 |        |                   |  |                                  |

|       |       |        |                   |  |                                  |
|-------|-------|--------|-------------------|--|----------------------------------|
| 10866 | ECL31 |        |                   |  |                                  |
| 01    |       | ELDA   | 1,0,SNC,ELDA      |  |                                  |
| 02    |       | ELEPS  | 1,0,SNC,ELDA      |  |                                  |
| 03    |       | SETUP  | 100,              |  |                                  |
| 04    | 02758 | JSR    | 0ENTIN            |  | INITIALIZE TEST.                 |
| 05    | 02759 | 100,   |                   |  |                                  |
| 06    | 02754 | RNDADR | ONE,TSTTOP        |  |                                  |
| 07    | 02755 | RAND   |                   |  |                                  |
| 08    | 02756 | JSR    | 0ENTRA            |  | IC(AC0)*RANDOM #                 |
| 09    | 02757 | 0RNADR |                   |  | IGET ADDRESS IN THE RANGE        |
| 10    | 02758 | ONE    |                   |  | IC(ONE) AND C(TSTTOP)            |
| 11    | 02759 | TSTTOP |                   |  |                                  |
| 12    | 02760 | MOV    | 0,1               |  |                                  |
| 13    | 02761 | ADCC   | 1,103,1,103       |  | IF ALL AC'S ARE SET TO 17777     |
| 14    | 02762 | MOV    | 1,103,1,0203      |  |                                  |
| 15    | 02763 | MOV    | 1,0203,1,0303     |  | IF AND CARRY IS INITIALISED      |
| 16    | 02764 | STA    | 1,0,03            |  | IF BY ADCC                       |
| 17    | 02765 | ADC    | 1,1               |  |                                  |
| 18    | 02766 | ELDA   | 1,0,0             |  | IF ELDA MUST NOT CHANGE ANY AC'S |
| 19    | 02770 | MOV    | 1,1,0NC           |  |                                  |
| 20    | 02771 | ERROR  |                   |  | FOR STATE OF CARRY EXCEPT AC1    |
| 21    | 02772 | COMW   | 1,103,1,103,0ZR   |  |                                  |
| 22    | 02773 | COMW   | 1,0203,1,0203,0ZR |  |                                  |
| 23    | 02774 | COMW   | 1,0303,1,0303,0ZR |  |                                  |
| 24    | 02775 | COMW   | 1,0303,1,0303,0ZR |  |                                  |
| 25    | 02776 | COMW   | 1,0303,1,0303,0ZR |  |                                  |
| 26    | 02777 | LOOP   |                   |  |                                  |
| 27    | 02778 | JSR    | 0ENTLO            |  | ITERATE TEST ROUTINE             |
| 28    | 02779 |        |                   |  |                                  |
| 29    | 02780 |        |                   |  |                                  |
| 30    | 02781 |        |                   |  |                                  |
| 31    | 02782 |        |                   |  |                                  |
| 32    | 02783 |        |                   |  |                                  |
| 33    | 02784 |        |                   |  |                                  |
| 34    | 02785 |        |                   |  |                                  |
| 35    | 02786 |        |                   |  |                                  |

```

10067 ECL31
01
02
03
04 03015 006305
05 03016 000144
06
07
08 03017 006310
09 03020 006311
10 03021 000233
11 03022 000234
12 03023 110000
13 03024 170000
14 03025 101000
15 03026 100000
16 03027 050403
17 03028 100000
18 03031 100000
19 03033 101000
20 03033 101000
21
22 03040 170014
23
24 03045 100014
25
26 03002 100014
27
28
29 03057 006306
30
31

LDA A2:
ELDA1 2,2,2C,ELDA
ELEFS 2,2,2C,ELDA
SETUP 100.
JSR @ENTIN
100.
RANDR ONE,TSTTOP
RAND
JSR @ENTRA
JSR @RNADR
ONE
TSTTOP
MOV 0,0
ADCC 2+103,2+103
MOV 2+103,2+103
MOV 2+103,2+103
STA 2,+3
ADC 2,2
ELDA 2,0,0
MOV 2,2,2C
ERROR 2+103,2+103,2R 1
COM# 2+103,2+103,2R 1
COM# 2+103,2+103,2R 1
ERROR 2+103,2+103,2R 1
COM# 2+103,2+103,2R 1
ERROR 2+103,2+103,2R 1
LOOP
JSR @ENTLO
ITERATE TEST ROUTINE

INITIALIZE TEST.
IC(AC0)=RANDOM #
IGET ADDRESS IN THE RANGE
IC(ONE) AND C(TSTTOP)
F
FALL AC'S ARE SET TO 17777
F
RAND CARRY IS INITIALIZED
F
FBI ADCZ
F
FELDA MUST NOT CHANGE ANY AC'S
FOR STATE OF CARRY EXCEPT AC3
ITERATE TEST ROUTINE

LDA A3:
ELDA1 3,0,SNC,ELDA
ELEFS 3,0,SNC,ELDA
SETUP 100.
JSR @ENTIN
100.
RANDR ONE,TSTTOP
RAND
JSR @ENTRA
JSR @RNADR
ONE
TSTTOP
MOV 0,0
ADCC 3+103,3+103
MOV 3+103,3+103
MOV 3+103,3+103
STA 3,+3
ADC 3,3
ELDA 3,0,0
MOV 3,3,SNC
ERROR 3+103,3+103,2R 1
COM# 3+103,3+103,2R 1
COM# 3+103,3+103,2R 1
ERROR 3+103,3+103,2R 1
COM# 3+103,3+103,2R 1
ERROR 3+103,3+103,2R 1
LOOP
JSR @ENTLO
ITERATE TEST ROUTINE

```



```

16868 ECL31
01
02          SETUP 500.
03          JSR @ENTIN
04          500.
05          RNDADR ONE,TSTTOP
06          RAND
07          JSR @ENTRA
08          JSR @RNADR
09          ONE
10          TSTTOP
11          STA @+11.
12          JSR @+1
13          ADDI @+3
14          ADD @+8
15          LDA 1,TSTTOP
16          SUBZ4 @+1,SNC
17          JMP LAB1
18          MOV @+3
19          LDA 1,@+3
20          ELDA 2,@+1
21          SUB# 1,@+3ZR
22          ERROR
23          LOOP
24          JSR @ENTLO
25
26          03173 086386
27
INITIALIZE TEST.
PC(AC0)=RANDOM #
PGT ADDRESS IN THE RANGE
PC(ONE) AND C(TSTTOP)
PSAVE IN ELDA INSTRUCTION
P
PADD PC OF ELDA+1 TO
PRNDM ADDR
P
PADDR> TSTTOP, GET NEW ADDR
PSET UP THIS ADDR IN ACS
PAC1 = C(RNDM ADDR)+ PC OF (ELDA+1
PAPPR ELDA,AC2 MUST BE
PC(RNDM ADDR)+PC OF (ELDA+1)
PITERATE TEST ROUTINE
P
LDAB01
01
02          03123 086386
03          03124 086386
04          03124 086386
05          03127 086333
06          03130 086334
07          03131 049484
08          03132 111800
09          03133 085980
10          03134 13667F
11          03136 136414
12          03143 086386
13
14          SETUP 500.
15          JSR @ENTIN
16          500.
17          RNDADR ONE,TSTTOP
18          RAND
19          JSR @ENTRA
20          JSR @RNADR
21          ONE
22          TSTTOP
23          STA @+4
24          MOV @+2
25          LDA 1,@+2
26          ELDA 3,@+8
27          SUB# 1,@+3ZR
28          ERROR
29          LOOP
30          JSR @ENTLO
31
32          03173 086386
33
INITIALIZE TEST.
PC(AC0)=RANDOM #
PGT ADDRESS IN THE RANGE
PC(ONE) AND C(TSTTOP)
P
PSAVE THIS ADDR IN AC2
PAC1 = C(RNDM ADDR)
PEXECUTE ELDA
P
PACS MUST BE= C(RNDM ADDR)
P
PITERATE TEST ROUTINE
P
LDAB11
01
02          03144 086386
03          03145 086764
04          03146 086318
05          03147 086311
06          03150 086333
07          03151 086334
08          03152 049413
09          03153 084461
10          03154 177778
11          03156 086311
12          03157 024234
13          03158 186433
14          03161 086765
15          03162 115688
16          03163 025460
17          03164 132478
18          03166 132414
19          03173 086386
20
21          SETUP 500.
22          JSR @ENTIN
23          500.
24          RNDADR ONE,TSTTOP
25          RAND
26          JSR @ENTRA
27          JSR @RNADR
28          ONE
29          TSTTOP
30          STA @+11.
31          JSR @+1
32          ADDI @+3
33          ADD @+8
34          LDA 1,TSTTOP
35          SUBZ4 @+1,SNC
36          JMP LAB1
37          MOV @+3
38          LDA 1,@+3
39          ELDA 2,@+1
40          SUB# 1,@+3ZR
41          ERROR
42          LOOP
43          JSR @ENTLO
44
45          03173 086386
46
ITERATE TEST ROUTINE
P

```

```

18071 ECL31
01
02 LDAB2: SETUP 500.
03 JSR #ENTIN
04 500.
05 RNDADR ONE,TSTTOP
06 RAND
07 JSR #ENTRA
08 JSR #RNADR
09 ONE
10 TSTTOP
11 HLV 0
12 MOV 0,2
13 STA 0,45
14 ADD 0,0
15 MOV 0,3
16 LDA 1,0,3
17 ELDA 0,0,2
18
19 SUB# 0,1,5ZR
20 ERROR
21 LOOP
22 JSR #ENTLO
23

INITIALIZE TEST.

IC(AC0)=RANDOM #
ICGET ADDRESS IN THE RANGE
IC(ONE) AND C(TSTTOP)

ISAVE (RNDM ADDR/2) IN ELDA
IAND INDEX REGISTER AC2
I
IAC0 =OFFSET+C(AC2)
ISET AC3=OFFSET+C(AC2)
IAC1 IS C(OFFSET+C(AC2))
IACB MUST BE EQ. TO AC1

ITERATE TEST ROUTINE

18072 ECL31
01
02 LDAB3: SETUP 500.
03 JSR #ENTIN
04 500.
05 RNDADR ONE,TSTTOP
06 RAND
07 JSR #ENTRA
08 JSR #RNADR
09 ONE
10 TSTTOP
11 HLV 0
12 MOV 0,3
13 STA 0,45
14 MOVOL
15 LDA 1,0,2
16 AIT 1,3
17 ELDA 0,0,3
18
19 SUB# 0,1,5ZR
20 ERROR
21 LOOP
22 JSR #ENTLO
23

INITIALIZE TEST.

IC(AC0)=RANDOM #
ICGET ADDRESS IN THE RANGE
IC(ONE) AND C(TSTTOP)

ISAVE (RNDM ADDR/2) IN ELDA
IAND INDEX REGISTER ACS
I
IAC1=C(OFFSET+C(ACS))
IAPTER ELDA.
IACB MUST BE EQ. TO
IC(OFFSET+C(ACS))

ITERATE TEST ROUTINE

```





```

10076 ECL31
01
02 03371 086305 LDACC1 SETUP 100.
03 03372 086144 JSR #ENTIN 100.
04 03373 086234 RNDADR ONE,TSTTOP
05
06 03374 086310 RAND
07 03375 086311 JSR #ENTRA
08 03376 086311 JSR #RNADR
09 03377 086234 ONE
10 03378 086234 TSTTOP
11 03379 086376 HLVL 0,2
12 03380 111140 LDA 2,0,2
13 03381 081000 MOV#2,2,SNC
14 03382 151015 JMP LACS
15 03383 086770 SUBZ# 1,TSTTOP
16 03384 082434 LDA 2,1,SNC
17 03385 146433 JMP LACS
18 03386 086765 MOV 0,3
19 03387 110000 ADI 1,3
20 03388 114810 ADI 0,0
21 03389 103248 STA 0,+3
22 03390 044403 LDA 1,0,2
23 03391 020000 ELDA 0,0,3
24 03392 123470 SUB# 0,1,SZR
25 03393 100000 ERROR
26 03394 106414 LOOP
27
28 03395 086306 JSR #ENTLO
29
30
31
INITIALIZE TEST.
IC(AC0)=RANDOM #
I GET ADDRESS IN THE RANGE
IC(ONE) AND C(TSTTOP)
FAC0=(RNDM ADDR/2)
FAC2=(RNDM ADDR+1)
I SHAKE SURE AC2 IS NOT ZERO.
I SHAKE SURE THAT (RNDM ADDR+1)
I DOES NOT INDIRECT
I SHAKE SURE AC3 IS NOT ZERO.
I
I SHAKE (RNDM ADDR/2+BIT 15) IN AC
I
FAC3=C(C(OFFSET+C(AC2)))
I ADD THE INDIRECT BIT TO OFFSET
I
I FAC1 SHOULD BE=
I
IC(C(OFFSET+C(AC2)))
ITERATE TEST ROUTINE

```

```

10075 ECL31
01
02 03371 086305 LDACC1 SETUP 100.
03 03372 086144 JSR #ENTIN 100.
04 03373 086234 RNDADR ONE,TSTTOP
05
06 03374 086310 RAND
07 03375 086311 JSR #ENTRA
08 03376 086311 JSR #RNADR
09 03377 086234 ONE
10 03378 086234 TSTTOP
11 03379 086376 HLVL 0,2
12 03380 111140 LDA 2,0,2
13 03381 081000 MOV#2,2,SNC
14 03382 151015 JMP LACS
15 03383 086770 SUBZ# 1,TSTTOP
16 03384 082434 LDA 2,1,SNC
17 03385 146433 JMP LACS
18 03386 086765 MOV 0,3
19 03387 110000 ADI 1,3
20 03388 114810 ADI 0,0
21 03389 103248 STA 0,+3
22 03390 044403 LDA 1,0,2
23 03391 020000 ELDA 0,0,3
24 03392 123470 SUB# 0,1,SZR
25 03393 100000 ERROR
26 03394 106414 LOOP
27
28 03395 086306 JSR #ENTLO
29
30
31
INITIALIZE TEST.
IC(AC0)=RANDOM #
I GET ADDRESS IN THE RANGE
IC(ONE) AND C(TSTTOP)
FAC0=(RNDM ADDR/2)
FAC2=(RNDM ADDR+1)
I SHAKE SURE THAT C(AC2)
I DOES NOT INDIRECT
I SHAKE SURE AC3 IS NOT ZERO.
I
I SHAKE (RNDM ADDR/2+BIT 15) IN AC
I
FAC3=C(C(OFFSET+C(AC2)))
I ADD THE INDIRECT BIT TO OFFSET
I
I FAC1 SHOULD BE=
I
IC(C(OFFSET+C(AC2)))
ITERATE TEST ROUTINE

```



|             |              |                               |             |              |                               |
|-------------|--------------|-------------------------------|-------------|--------------|-------------------------------|
| 18079 ECL31 | LOAD21       | ELDA2 2,LAD2                  | 18080 ECL31 | LOAD31       | ELDA2 3,LAD3                  |
| 01          | SETUP 100,   | JSR #ENTIN                    | 01          | SETUP 100,   | JSR #ENTIN                    |
| 02          | 100,         | LOBFAD                        | 02          | 100,         | LOBFAD                        |
| 03          | 03542 003305 | RNDADR LOWBF,LOBFU            | 03          | 03610 003305 | RNDADR LOWBF,LOBFU            |
| 04          | 03543 000144 | RAND                          | 04          | 03614 000144 | RAND                          |
| 05          |              | JSR #ENTRA                    | 05          |              | JSR #ENTRA                    |
| 06          |              | JSR #RNADR                    | 06          |              | JSR #RNADR                    |
| 07          |              | LOBF                          | 07          |              | LOBF                          |
| 08          | 03544 003310 | LOBFAD                        | 08          | 03615 003310 | LOBFAD                        |
| 09          | 03545 003311 | RAND                          | 09          | 03616 003311 | RAND                          |
| 10          | 03546 003315 | JSR #ENTRA                    | 10          | 03617 003315 | JSR #ENTRA                    |
| 11          | 03547 003312 | JSR #RNADR                    | 11          | 03620 003312 | JSR #RNADR                    |
| 12          |              | LOBF                          | 12          | 03621 040251 | LOBF                          |
| 13          | 03550 040251 | STA 0,LDSVM                   | 13          |              | STA 0,LDSVM                   |
| 14          |              | HIBFAD                        | 14          |              | HIBFAD                        |
| 15          |              | RNDADR HIBBF,HIBFU            | 15          |              | RNDADR HIBBF,HIBFU            |
| 16          |              | RAND                          | 16          | 03622 003310 | RAND                          |
| 17          | 03551 003310 | JSR #ENTRA                    | 17          | 03623 003311 | JSR #ENTRA                    |
| 18          | 03552 003311 | JSR #RNADR                    | 18          | 03624 003313 | JSR #RNADR                    |
| 19          | 03553 003315 | HIGBF                         | 19          | 03625 003316 | HIGBF                         |
| 20          | 03554 003316 | HIBFU                         | 20          | 03626 103240 | HIBFU                         |
| 21          | 03555 103240 | ADDR 0,0                      | 21          | 03627 040252 | ADDR 0,0                      |
| 22          | 03556 040252 | STA 0,LDSV1                   | 22          |              | STA 0,LDSV1                   |
| 23          |              | HIBFAD                        | 23          |              | HIBFAD                        |
| 24          |              | RNDADR HIBBF,HIBFU            | 24          |              | RNDADR HIBBF,HIBFU            |
| 25          | 03557 003310 | RAND                          | 25          | 03630 003310 | RAND                          |
| 26          | 03558 003311 | JSR #ENTRA                    | 26          | 03631 003311 | JSR #ENTRA                    |
| 27          | 03559 003315 | JSR #RNADR                    | 27          | 03632 003315 | JSR #RNADR                    |
| 28          | 03560 003316 | HIGBF                         | 28          | 03633 003316 | HIGBF                         |
| 29          | 03561 024252 | HIBFU                         | 29          | 03634 024252 | HIBFU                         |
| 30          | 03562 024252 | LDA                           | 30          | 03635 127220 | LDA                           |
| 31          | 03564 127220 | ADDR                          | 31          | 03637 000771 | ADDR                          |
| 32          | 03566 000771 | SUB#                          | 32          | 03640 111000 | SUB#                          |
| 33          | 03567 111000 | MOV 0,2                       | 33          | 03641 034252 | MOV 0,2                       |
| 34          | 03570 034052 | LDA 3,LDSV1                   | 34          | 03642 050000 | LDA 3,LDSV1                   |
| 35          | 03571 050000 | STA 3,0,2                     | 35          | 03643 172220 | STA 3,0,2                     |
| 36          | 03572 172220 | STA 3,3                       | 36          | 03644 030251 | STA 3,3                       |
| 37          | 03573 030251 | LDA 2,LDSVM                   | 37          | 03645 051400 | LDA 2,LDSVM                   |
| 38          | 03574 051400 | LDA 2,0,3                     | 38          | 03646 025000 | LDA 2,0,3                     |
| 39          | 03575 025000 | LDA 1,0,2                     | 39          | 03647 110222 | LDA 1,0,2                     |
| 40          | 03576 110222 | MOVZ 0,2,0ZC                  | 40          | 03650 140000 | MOVZ 0,2,0ZC                  |
| 41          | 03577 110010 | FALSO STORE (RNDM ADDR 3)/2   | 41          | 03651 140010 | FALSO STORE (RNDM ADDR 3)/2   |
| 42          | 03578 140010 | FALSO STORE (RNDM ADDR 3) AND | 42          | 03652 103240 | FALSO STORE (RNDM ADDR 3) AND |
| 43          | 03580 143370 | IN (RNDM ADDR 1) IN           | 43          | 03653 040402 | IN (RNDM ADDR 1) IN           |
| 44          | 03581 040402 | IN ELDA INSTRUCTION           | 44          | 03654 103240 | IN ELDA INSTRUCTION           |
| 45          | 03582 040402 | HLV 0,0                       | 45          | 03655 100000 | HLV 0,0                       |
| 46          | 03583 100000 | STA 0,0,2                     | 46          | 03656 106414 | STA 0,0,2                     |
| 47          | 03584 106414 | ELDA 0,0,2                    | 47          |              | ELDA 0,0,2                    |
| 48          | 03585 106414 | SUB# 0,1,0ZR                  | 48          |              | SUB# 0,1,0ZR                  |
| 49          |              | ERROR                         | 49          |              | ERROR                         |
| 50          |              | LOOP                          | 50          | 03656 003306 | LOOP                          |
| 51          | 03512 003306 | JSR #ENTLO                    | 51          |              | JSR #ENTLO                    |

INITIALIZE TEST.

IC(AC0)=RANDOM #

IGET ADDRESS IN THE RANGE  
IC(LOWBF) AND C(LOBFU)

IC(LOBV0)=ADDR FROM LOW BUFF

IC(AC0)=RANDOM #

IGET ADDRESS IN THE RANGE  
IC(HIGBF) AND C(HIBFU)

IC(LDSV1)=ADDR FROM HIGH  
BUFF WITH INDIRECT BIT SET

IC(AC0)=RANDOM #

IGET ADDRESS IN THE RANGE  
IC(HIGBF) AND C(HIBFU)

SHAKE SURE THAT RNDM NOS.  
FROM HIBUFF ARE DIFFERENT

FALSO STORE (RNDM ADDR 2)  
IN (RNDM ADDR 3) AND  
(RNDM ADDR 2)

STORE (RNDM ADDR 3)/2  
IN AC2

IN ELDA INSTRUCTION  
FALSO STORE (RNDM ADDR 3) AND  
IN ELDA INSTRUCTION

FALSO MUST BE#

IC(RNDM ADDR 1)

ITERATE TEST ROUTINE

10881 ECL31

```

01
02
03
04
05
06
07
08 03664 004401
09 03665 054210
10
11 03666 006305
12 03667 000144
13
14
15 03670 000310
16 03671 000311
17 03672 000310
18 03673 000312
19 03674 048410
20
21
22 03675 000310
23 03676 048253
24 03677 101040
25 03678 100000
26 03679 131000
27 03680 100000
28 03681 140070
29 03682 100000
30 03683 100000
31 03684 100000
32 03718 124014
33
34 03717 150014
35
36 03724 174014
37 03731 024253
38 03732 100414
39
40
41
42 03737 000306
43

```

```

; TESTING 'ESTA' -LONG STA INSTRUCTION
;
;

```

```

STA08: ESTAI 0,0,0NC,LOBFAD

```

```

TSTLOC
JBR *+1
STA 3,HELP
SETUP 100.
JBR #ENTIN
100.
;INITIALIZE TEST.

```

```

RAND
RNDADR LOMB,LOBFU
JBR #ENTRA
JBR #RNADR
LOMBF
LOBFU
STA 0,+.0.
;IC(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(LOMBF) AND C(LOBFU)
;

```

```

;IC(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(LOMBF) AND C(LOBFU)
;
JBR #ENTRA
JBR #RNADR
LOMBF
LOBFU
STA 0,+.0.
;IC(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(LOMBF) AND C(LOBFU)
;

```

```

;IC(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(LOMBF) AND C(LOBFU)
;
JBR #ENTRA
JBR #RNADR
LOMBF
LOBFU
STA 0,+.0.
;IC(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(LOMBF) AND C(LOBFU)
;

```

10882 ECL31

```

01
02
03
04 03740 004401
05 03741 054210
06
07 03742 006305
08 03743 000144
09
10
11 03744 000310
12 03745 000311
13 03746 000311
14 03747 000311
15 03748 000310
16 03750 048410
17
18 03751 000310
19 03752 048253
20 03753 100000
21 03754 150000
22 03755 150000
23 03756 161000
24 03757 140070
25 03758 000000
26 03761 120002
27
28 03766 150014
29
30 03773 174014
31
32 04000 100014
33
34 04006 030253
35 04006 132414
36
37
38 04013 000306
39
40
41
42
43

```

```

STA08: ESTAI 1,1,0ZC,HIBFAD

```

```

TSTLOC
JBR *+1
STA 3,HELP
SETUP 100.
JBR #ENTIN
100.
;INITIALIZE TEST.

```

```

RAND
RNDADR HIGBF,HIBFU
JBR #ENTRA
JBR #RNADR
HIGBF
HIBFU
STA 0,+.0.
;IC(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(HIGBF) AND C(HIBFU)
;

```

```

;IC(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(HIGBF) AND C(HIBFU)
;
JBR #ENTRA
JBR #RNADR
HIGBF
HIBFU
STA 0,+.0.
;IC(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(HIGBF) AND C(HIBFU)
;

```

```

;IC(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(HIGBF) AND C(HIBFU)
;
JBR #ENTRA
JBR #RNADR
HIGBF
HIBFU
STA 0,+.0.
;IC(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(HIGBF) AND C(HIBFU)
;

```

```

;ITERATE TEST ROUTINE

```

18884 ECL31

```
01 STAAB1 ESTAI 2,0,8NC,HIBFAD  
02 TSTLOC  
03  
04 04014 04401 *(ADDR OF TEST+1) IS  
05 04015 044216 *HELP  
06 *1  
07 04016 040305 SETUP 100,  
08 04017 040144 JSR #ENTIN  
09  
10 HIBFAD  
11 RNDADR HIBGF,HIBFU  
12 RAND  
13 04020 040310 IC(AC0)=RANDOM #  
14 04021 040311 IGET ADDR IN THE RANGE  
15 04022 040313 HIBGF #RNADR IC(HIBGF) AND C(HIBFU)  
16 04023 040316 HIBFU  
17 04024 040410 STA 0,+,0,  
18 RAND  
19 04025 040310 JSR #ENTRA IC(AC0)=RANDOM #  
20 04026 040253 STA 0,STLCR  
21 04027 110400 MOV 0,2  
22 04030 170000 ADC 2+183,2+183  
23 04031 151000 MOV 2+183,2+183  
24 04032 150000 MOV 2+283,2+383  
25 04033 152070 ESTA 2,0,0  
26 04035 151003 NOV 2,2,8NC ICARRY MUST BE UNCHANGED  
27 ERROR  
28 04042 174014 COM# 2+183,2+183,SZR I  
29 ERROR  
30 04047 180014 COM# 2+283,2+283,SZR I  
31 ERROR  
32 04054 184014 COM# 2+383,2+383,SZR I  
33 ERROR  
34 04061 044203 LOA 2+183,STLCR  
35 04062 164414 SUB# 2,2+183,SZR I  
36 ERROR  
37 LOOP  
38 04067 040306 JSR #ENTLO ITERATE TEST ROUTINE  
39
```

18884 ECL31

```
01 STAAB1 ESTAI 3,2,8ZC,LOBFAD  
02 TSTLOC  
03  
04 04070 04401 *(ADDR OF TEST+1) IS  
05 04071 044216 *HELP  
06 *1  
07 04072 040305 SETUP 100,  
08 04073 040144 JSR #ENTIN  
09  
10 LOBFAD  
11 RNDADR LOBF,LOBFU  
12 RAND  
13 04074 040310 IC(AC0)=RANDOM #  
14 04075 040311 IGET ADDR IN THE RANGE  
15 04077 040312 LOBF IC(LOBF) AND C(LOBFU)  
16 04100 040410 STA 0,+,0,  
17 RAND  
18 04101 040310 JSR #ENTRA IC(AC0)=RANDOM #  
19 04102 040253 STA 0,STLCR  
20 04103 110000 MOV 0,2  
21 04104 190000 ADC 3+183,3+183  
22 04105 190000 MOV 3+183,3+183  
23 04106 151000 MOV 3+283,3+283  
24 04107 150070 ESTA 3,0,0  
25 04108 040000 NOV 3,2,8ZC ICARRY MUST BE UNCHANGED  
26 04111 175002 ERROR  
27 COM# 3+183,3+183,SZR I  
28 ERROR  
29 04116 180014 COM# 3+283,3+283,SZR I  
30 ERROR  
31 04123 124014 COM# 3+383,3+383,SZR I  
32 04130 150014 ERROR  
33 LOA 3+183,STLCR  
34 04135 040203 SUB# 3,3+183,SZR I  
35 04136 162414 ERROR  
36 LOOP  
37 04143 040306 JSR #ENTLO ITERATE TEST ROUTINE  
38  
39
```

10005 ECL31  
01

10006 ECL31  
01  
02 04144 004401  
03 04145 054216  
04 04146 006305  
05 04147 000144  
06 04148 006310  
07 04149 006311  
08 04150 006312  
09 04151 006313  
10 04152 006314  
11 04153 006315  
12 04154 040203  
13 04155 040404  
14 04156 006310  
15 04157 040204  
16 04158 140070  
17 04159 006000  
18 04160 006203  
19 04161 006204  
20 04162 102414  
21 04163 006204  
22 04164 102414  
23  
24  
25  
26  
27 04171 006306

STAGE:

TESTLOC

JSR \*1  
STA S\*HELP  
SETUP 100.  
JSR \*ENTIN  
100.  
MIBFAD  
RNDADR MIBGF,MIBFU  
RAND  
JSR \*ENTRA  
JSR \*RNDADR  
MIBGF  
MIBFU  
STA 0,STLCR  
STA 0,++4  
RAND  
JSR \*ENTRA  
STA 0,STLC1  
ESTA 0,0,0  
LDA 1,0,STLCR  
LDA 2,STLC1  
SUB# 1,0,500  
ERROR  
LOOP  
JBR \*ENTLO

IC(ADDR OF TEST\*1) IS  
\*SAVED IN LOC. HELP  
INITIALIZE TEST.

IC(AC0)=RANDOM #  
IC(ADDR IN THE RANGE  
IC(MIBGF) AND C(MIBFU)  
JSAVE RNDH ADDR IN STLC0  
FALSO IN ESTA

IC(AC0)=RANDOM #  
JSAVE RNDH DATA IN STLC1  
FAND IN C(STLCR)

JAC1 MUST BE EQ. TO  
RNDH DATA  
!

JITERATE TEST ROUTINE

```

10088 ECL31
01
02
03 04172 004001
04 04173 004210
05
06 04174 006305
07 04175 000144
08
09
10
11 04176 006310
12 04177 006311
13 04200 000313
14 04201 000316
15 04202 040253
16 04203 100470
17 000000
18 04205 102400
19 04206 040404
20
21 04207 006310
22 04210 100000
23 04211 100470 04011
24
25 04213 002253
26 04214 112414
27
28
29 04221 006306
30

```

STAB11  
 TSTLOC  
 JSR \*+1  
 STA 3,HELP  
 SETUP 100,  
 JSR #ENTIN  
 100,  
 HIBFAD  
 RNDADR HIGBF,HIBFU  
 RAND  
 JSR #ENTRA  
 JSR #RNADR  
 HIGBF  
 HIBFU  
 STA 0,STLCO  
 ELEF 1,0401...1  
 SUB 1,0  
 STA 0,0401+1  
 RAND  
 JSR #ENTRA  
 MOV 0,1  
 ESTA 1,0,1  
 LOA 2,0STLCO  
 SUBW 0,0,SXP  
 ERROR  
 LOOP  
 JSR #ENTLO

! (ADDR OF TEST+1) IS  
 ! SAVED IN LOC. HELP  
 ! INITIALIZE TEST.  
 !  
 ! C(A0)=RANDOM #  
 ! GET ADDRESS IN THE RANGE  
 ! C(HIGBF) AND C(HIBFU)  
 !  
 ! SAVE RNDM ADDR IN STLCO  
 !  
 ! SAVE (RNDM ADDR-PC OF  
 ! TEST+1) IN ESTA INSTRUCTION  
 !  
 ! C(A0)=RANDOM #  
 ! C(A0)=RANDOM DATA  
 ! STORE RNDM DATA IN C(STLCO)  
 !  
 ! AC2 MUST BE EQ. TO  
 ! RANDOM DATA  
 !  
 ! ITERATE TEST ROUTINE



```

10000 ECL31
01
02
03
04 04253 004401
05 04254 054216
06
07 04255 006305
08 04256 000144
09
10
11
12 04257 006310
13 04260 006311
14 04261 006315
15 04262 006312
16 04263 006253
17
18 04264 006310
19 04265 100000
20 04266 006253
21 04267 110222
22 04270 114810
23 04271 143370
24 04272 040402
25 04273 147478
26
27 04275 022203
28 04276 106414
29
30
31 04303 006306

```

```

10000 ECL31
01
02
03
04 04222 004401
05 04223 054216
06
07 04224 006305
08 04225 000144
09
10
11
12 04226 006310
13 04227 006311
14 04230 006313
15 04231 006316
16 04232 040253
17
18 04233 006310
19 04234 100000
20 04235 006253
21 04236 110222
22 04237 110010
23 04240 143370
24 04241 040402
25 04242 147070
26
27 04244 022203
28 04245 106414
29
30
31 04252 006306

```

```

          STAB2: 2,HIBFAD
          ESTAR
          TSTLOC
          JSR
          STA
          SETUP
          JSR
          LDB
          RNDADR
          RAND
          JSR
          LDA
          MOVZR
          ADI
          HLV
          STA
          ESTA
          LDA
          SUBM
          ERROR
          LOOP
          JSR

```

```

          STAB3: 3,LOBFAD
          ESTAR
          TSTLOC
          JSR
          STA
          SETUP
          JSR
          LDB
          RNDADR
          RAND
          JSR
          LDA
          MOVZR
          ADI
          HLV
          STA
          ESTA
          LDA
          SUBM
          ERROR
          LOOP
          JSR

```

```

          STAB2: 2,HIBFAD
          ESTAR
          TSTLOC
          JSR
          STA
          SETUP
          JSR
          LDB
          RNDADR
          RAND
          JSR
          LDA
          MOVZR
          ADI
          HLV
          STA
          ESTA
          LDA
          SUBM
          ERROR
          LOOP
          JSR

```

```

          STAB3: 3,LOBFAD
          ESTAR
          TSTLOC
          JSR
          STA
          SETUP
          JSR
          LDB
          RNDADR
          RAND
          JSR
          LDA
          MOVZR
          ADI
          HLV
          STA
          ESTA
          LDA
          SUBM
          ERROR
          LOOP
          JSR

```

10091 ECL31

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

NOW CHECK "ESTA" SINGLE LEVEL INDIRECT
STAC0: ESTA3 LOBFAD,HIBFAD,SAC00
TSYLOC
JSP *+1
STA 3,HELP
SETUP 100.
JSP #ENTIN
INITIALIZE TEST.
IC(AC0)=RANDOM #
JSET ADDRESS IN THE RANGE
IC(L0BF) AND C(HIBFU)
IC(RNDM ADDR 1)=(RNDM ADDR 2)
RNDADR HIBF,HIBFU
RND
JSP #ENTRA
JSP #RNADR
LOBF
LOBFU
STA 0,STLC0
LDA 1,STLC0
SUB# 1,0,SNR
JMP SAC00
STA 0,STLC0
RND
JSP #ENTRA
MOV 0,1
LDA 0,STLC0
IORI 100000,0
STA 0,+4
SBI 2,0
STA 0,+4
ESTA 1,00,1
ELDA 2,00,0
SUB# 1,2,SZR
ERROR
LOOP
JSP #ENTLO
ITERATE TEST ROUTINE

```

10092 ECL31

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

HIBFAD,LOBFAD,SAC10
TSYLOC
JSP *+1
STA 3,HELP
SETUP 100.
JSP #ENTIN
INITIALIZE TEST.
IC(AC0)=RANDOM #
JSET ADDRESS IN THE RANGE
IC(HIBF) AND C(HIBFU)
IC(RNDM ADDR 1)=(RNDM ADDR 2)
RNDADR LOBF,LOBFU
RND
JSP #ENTRA
JSP #RNADR
LOBF
LOBFU
LDA 1,STLC0
SUB# 1,0,SNR
JMP SAC10
STA 0,STLC0
RND
JSP #ENTRA
MOV 0,1
LDA 0,STLC0
ELEF 2,SAC1-..1
SUB 2,0
IORI 100000,0
STA 0,+4
SBI 2,0
STA 0,+4
ESTA 1,00,1
ELDA 3,00,1
SUB# 1,3,SZR
ERROR
LOOP
JSP #ENTLO
ITERATE TEST ROUTINE

```

10093 ECL31

```

01 ESTA4 LOBFAD,HIBFAD,SAC20,2
02 ESTA3 LOBFAD,HIBFAD,SAC20
03 TESTLOC
04 JSR *+1
05 STA S,HELP
06 SETUP 100.
07 JSR *ENTIN
08 *+1
09 *+1
10 *+1
11 RNDADR LOBF,LOBFU
12 RAND
13 JSR *ENTRA
14 JSR *RNADR
15 LOBF
16 LOBFU
17 STA *STLCR
18 HIBFAD
19 RNDADR HIGBF,HIBFU
20 RAND
21 JSR *ENTRA
22 JSR *RNADR
23 HIGBF
24 HIBFU
25 LDA 1,STLCR
26 SUB# 1,0,SNR
27 JMP SAC20
28 STA *STLCR
29 RAND
30 JSR *ENTRA
31 JSR *RNADR
32 LDA *STLCR
33 MOVZR 0,2,SEC FAC2 IS
34 ADI 1,2
35 HLV 0
36 *OR1 100000,0
37 100000
38 STA 0,+3
39 STA 0,+4
40 ESTA 1,00,2
41
42 ELDA 0,00,2
43
44 SUB# 1,0,5ZR
45 ERROR
46 LOOP
47 JSR *ENTLO

```

10094 ECL31

```

01 ESTA4 HIBFAD,HIBFAD,SAC30,3
02 ESTA3 HIBFAD,HIBFAD,SAC30
03 TESTLOC
04 JSR *+1
05 STA S,HELP
06 SETUP 100.
07 JSR *ENTIN
08 *+1
09 *+1
10 *+1
11 RNDADR HIGBF,HIBFU
12 RAND
13 JSR *ENTRA
14 JSR *RNADR
15 HIGBF
16 HIBFU
17 STA *STLCR
18 HIBFAD
19 RNDADR HIGBF,HIBFU
20 RAND
21 JSR *ENTRA
22 JSR *RNADR
23 HIGBF
24 HIBFU
25 LDA 1,STLCR
26 SUB# 1,0,SNR
27 JMP SAC30
28 STA *STLCR
29 RAND
30 JSR *ENTRA
31 JSR *RNADR
32 LDA *STLCR
33 MOVZR 0,3,SEC FAC3 IS
34 ADI 1,3
35 HLV 0
36 *OR1 100000,0
37 100000
38 STA 0,+3
39 STA 0,+4
40 ESTA 1,00,3
41
42 ELDA 0,00,3
43
44 SUB# 1,0,5ZR
45 ERROR
46 LOOP
47 JSR *ENTLO

```

10095 ECL31

```

01 ESTA4 HIBFAD,HIBFAD,SAC30,3
02 ESTA3 HIBFAD,HIBFAD,SAC30
03 TESTLOC
04 JSR *+1
05 STA S,HELP
06 SETUP 100.
07 JSR *ENTIN
08 *+1
09 *+1
10 *+1
11 RNDADR HIGBF,HIBFU
12 RAND
13 JSR *ENTRA
14 JSR *RNADR
15 HIGBF
16 HIBFU
17 STA *STLCR
18 HIBFAD
19 RNDADR HIGBF,HIBFU
20 RAND
21 JSR *ENTRA
22 JSR *RNADR
23 HIGBF
24 HIBFU
25 LDA 1,STLCR
26 SUB# 1,0,SNR
27 JMP SAC30
28 STA *STLCR
29 RAND
30 JSR *ENTRA
31 JSR *RNADR
32 LDA *STLCR
33 MOVZR 0,3,SEC FAC3 IS
34 ADI 1,3
35 HLV 0
36 *OR1 100000,0
37 100000
38 STA 0,+3
39 STA 0,+4
40 ESTA 1,00,3
41
42 ELDA 0,00,3
43
44 SUB# 1,0,5ZR
45 ERROR
46 LOOP
47 JSR *ENTLO

```

10895 ECL31

```

01 07 04026 004401
02 08 04527 054216
03 09 04530 006305
04 10 04531 008144
05 11 04532 006310
06 12 04533 006311
07 13 04534 006312
08 14 04535 006313
09 15 04536 040253
10 16 04537 006310
11 17 04540 006311
12 18 04541 006313
13 19 04542 006316
14 20 04543 103240
15 21 04544 040254
16 22 04545 006310
17 23 04546 006311
18 24 04547 006313
19 25 04548 006316
20 26 04549 103240
21 27 04550 006310
22 28 04551 040253
23 29 04552 103240
24 30 04553 006313
25 31 04554 110000
26 32 04555 103240
27 33 04556 040254
28 34 04557 040254
29 35 04558 050000
30 36 04559 172250
31 37 04560 030253
32 38 04561 051400
33 39 04562 006310
34 40 04563 110000
35 41 04564 100000
36 42 04565 100000
37 43 04570 006305
38 44 04571 102414
39 45 04572 006306
40 46 04573 006306
41 47 04574 006306
42 48 04575 006306
43 49 04576 006306
44 50 04577 006306
45 51 04578 006306
46 52 04579 006306
47 53 04580 006306
48 54 04581 006306
49 55 04582 006306
50 56 04583 006306
51 57 04584 006306
52 58 04585 006306
53 59 04586 006306
54 60 04587 006306
55 61 04588 006306
56 62 04589 006306
57 63 04590 006306

```

10896 ECL31

```

01 10896 ECL31
02 07 04577 004401
03 08 04578 054216
04 09 04579 006305
05 10 04580 008144
06 11 04581 006310
07 12 04582 006311
08 13 04583 006312
09 14 04584 006313
10 15 04585 040253
11 16 04586 006310
12 17 04587 006311
13 18 04588 006313
14 19 04589 103240
15 20 04590 040254
16 21 04591 006310
17 22 04592 006311
18 23 04593 006313
19 24 04594 103240
20 25 04595 040254
21 26 04596 006310
22 27 04597 006311
23 28 04598 006313
24 29 04599 103240
25 30 04600 040254
26 31 04601 006310
27 32 04602 006311
28 33 04603 006313
29 34 04604 103240
30 35 04605 040254
31 36 04606 006310
32 37 04607 006311
33 38 04608 006313
34 39 04609 103240
35 40 04610 040254
36 41 04611 006310
37 42 04612 006311
38 43 04613 006313
39 44 04614 103240
40 45 04615 040254
41 46 04616 006310
42 47 04617 006311
43 48 04618 006313
44 49 04619 103240
45 50 04620 040254
46 51 04621 006310
47 52 04622 006311
48 53 04623 006313
49 54 04624 103240
50 55 04625 040254
51 56 04626 006310
52 57 04627 006311
53 58 04628 006313
54 59 04629 103240
55 60 04630 040254
56 61 04631 006310
57 62 04632 006311
58 63 04633 006313
59 64 04634 103240
60 65 04635 040254
61 66 04636 006310
62 67 04637 006311
63 68 04638 006313
64 69 04639 103240
65 70 04640 040254
66 71 04641 006310
67 72 04642 006311
68 73 04643 006313
69 74 04644 103240
70 75 04645 040254
71 76 04646 006310
72 77 04647 006311
73 78 04648 006313
74 79 04649 103240
75 80 04650 040254
76 81 04651 006310
77 82 04652 006311
78 83 04653 006313
79 84 04654 103240
80 85 04655 040254
81 86 04656 006310
82 87 04657 006311
83 88 04658 006313
84 89 04659 103240
85 90 04660 040254
86 91 04661 006310
87 92 04662 006311
88 93 04663 006313
89 94 04664 103240
90 95 04665 040254
91 96 04666 006310
92 97 04667 006311
93 98 04668 006313
94 99 04669 103240
95 100 04670 040254
96 101 04671 006310
97 102 04672 006311
98 103 04673 006313
99 104 04674 103240
100 105 04675 040254

```

| 10897 ECL31 | STAD2:       | ESTAS  | 2, S, AD2    | STAD3: | ESTAS  | 3, S, AD3    |
|-------------|--------------|--------|--------------|--------|--------|--------------|
| 01          |              | TSTLOC |              |        | TSTLOC |              |
| 02          |              | JSR    | *+1          |        | JSR    | *+1          |
| 03          |              | SETUP  | 3, HELP      |        | SETUP  | 3, HELP      |
| 04          | 04653 004401 | JSR    | ENTIN        |        | JSR    | ENTIN        |
| 05          | 04654 054210 | 100,   |              |        | 100,   |              |
| 06          | 04655 063305 | RAND   |              |        | RAND   |              |
| 07          | 04656 060144 | JSR    | ENTRA        |        | JSR    | ENTRA        |
| 08          | 04657 066310 | STA    | 0, STSV0     |        | STA    | 0, STSV0     |
| 09          | 04658 048255 | LOBFAD | LOMBF, LOBFU |        | LOBFAD | LOMBF, LOBFU |
| 10          | 04659 048255 | RAND   |              |        | RAND   |              |
| 11          | 04660 066310 | JSR    | ENTRA        |        | JSR    | ENTRA        |
| 12          | 04661 066311 | LOMBF  | LOBFU        |        | LOMBF  | LOBFU        |
| 13          | 04662 066312 | LDA    | 1, STLC0     |        | LDA    | 1, STLC0     |
| 14          | 04663 066313 | JMP    | SAD2         |        | JMP    | SAD2         |
| 15          | 04664 066314 | ADDR   | 0, 0         |        | ADDR   | 0, 0         |
| 16          | 04665 066315 | STA    | 0, STLC1     |        | STA    | 0, STLC1     |
| 17          | 04666 066316 | HISFAD | HISBF, HISFU |        | HISFAD | HISBF, HISFU |
| 18          | 04667 066317 | RAND   |              |        | RAND   |              |
| 19          | 04668 066318 | JSR    | ENTRA        |        | JSR    | ENTRA        |
| 20          | 04669 066319 | LOMBF  | LOBFU        |        | LOMBF  | LOBFU        |
| 21          | 04670 066320 | LDA    | 1, STLC0     |        | LDA    | 1, STLC0     |
| 22          | 04671 066321 | JMP    | SAD3         |        | JMP    | SAD3         |
| 23          | 04672 066322 | ADDR   | 0, 0         |        | ADDR   | 0, 0         |
| 24          | 04673 066323 | STA    | 0, STLC1     |        | STA    | 0, STLC1     |
| 25          | 04674 066324 | HISFAD | HISBF, HISFU |        | HISFAD | HISBF, HISFU |
| 26          | 04675 066325 | RAND   |              |        | RAND   |              |
| 27          | 04676 066326 | JSR    | ENTRA        |        | JSR    | ENTRA        |
| 28          | 04677 066327 | LOMBF  | LOBFU        |        | LOMBF  | LOBFU        |
| 29          | 04678 066328 | LDA    | 1, STLC0     |        | LDA    | 1, STLC0     |
| 30          | 04679 066329 | JMP    | SAD3         |        | JMP    | SAD3         |
| 31          | 04680 066330 | ADDR   | 0, 0         |        | ADDR   | 0, 0         |
| 32          | 04681 066331 | STA    | 0, STLC1     |        | STA    | 0, STLC1     |
| 33          | 04682 066332 | HISFAD | HISBF, HISFU |        | HISFAD | HISBF, HISFU |
| 34          | 04683 066333 | RAND   |              |        | RAND   |              |
| 35          | 04684 066334 | JSR    | ENTRA        |        | JSR    | ENTRA        |
| 36          | 04685 066335 | LOMBF  | LOBFU        |        | LOMBF  | LOBFU        |
| 37          | 04686 066336 | LDA    | 1, STLC0     |        | LDA    | 1, STLC0     |
| 38          | 04687 066337 | JMP    | SAD3         |        | JMP    | SAD3         |
| 39          | 04688 066338 | ADDR   | 0, 0         |        | ADDR   | 0, 0         |
| 40          | 04689 066339 | STA    | 0, STLC1     |        | STA    | 0, STLC1     |
| 41          | 04690 066340 | HISFAD | HISBF, HISFU |        | HISFAD | HISBF, HISFU |
| 42          | 04691 066341 | RAND   |              |        | RAND   |              |
| 43          | 04692 066342 | JSR    | ENTRA        |        | JSR    | ENTRA        |
| 44          | 04693 066343 | LOMBF  | LOBFU        |        | LOMBF  | LOBFU        |
| 45          | 04694 066344 | LDA    | 1, STLC0     |        | LDA    | 1, STLC0     |
| 46          | 04695 066345 | JMP    | SAD3         |        | JMP    | SAD3         |
| 47          | 04696 066346 | ADDR   | 0, 0         |        | ADDR   | 0, 0         |
| 48          | 04697 066347 | STA    | 0, STLC1     |        | STA    | 0, STLC1     |
| 49          | 04698 066348 | HISFAD | HISBF, HISFU |        | HISFAD | HISBF, HISFU |
| 50          | 04699 066349 | RAND   |              |        | RAND   |              |
| 51          | 04700 066350 | JSR    | ENTRA        |        | JSR    | ENTRA        |
| 52          | 04701 066351 | LOMBF  | LOBFU        |        | LOMBF  | LOBFU        |
| 53          | 04702 066352 | LDA    | 1, STLC0     |        | LDA    | 1, STLC0     |
| 54          | 04703 066353 | JMP    | SAD3         |        | JMP    | SAD3         |
| 55          | 04704 066354 | ADDR   | 0, 0         |        | ADDR   | 0, 0         |
| 56          | 04705 066355 | STA    | 0, STLC1     |        | STA    | 0, STLC1     |
| 57          | 04706 066356 | HISFAD | HISBF, HISFU |        | HISFAD | HISBF, HISFU |

10898 ECL31

01

02

03 04730 004401

04 04731 054210

05 04732 066305

06 04733 000144

07 04734 066310

08 04735 048255

09 04736 066310

10 04737 066311

11 04740 066315

12 04741 066312

13 04742 048253

14 04743 066310

15 04744 066311

16 04745 066315

17 04746 066312

18 04747 048253

19 04750 106415

20 04751 066772

21 04752 103240

22 04753 048254

23 04754 066310

24 04755 066311

25 04756 066315

26 04757 066316

27 04760 111800

28 04761 034254

29 04762 055000

30 04763 177220

31 04764 066253

32 04766 051400

33 04766 115222

34 04767 114810

35 04770 143370

36 04771 103240

37 04772 048403

38 04773 024205

39 04774 147470

40 04775 100000

41 04776 022253

42 04777 106414

43 04778 066306

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

10899 ECL31

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

10900 ECL31

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

10100 ECL31

```

01
02
03
04
05
06
07 05005 084401
08 05005 054215
09
10 05007 056305
11 05010 080144
12 05011 080200
13 05010 040003
14 05013 050315
15 05014 054313
16 05016 054255
17
18 05016 050310
19 05017 054255
20 05020 143070
21
22 05022 143470
23
24 05024 151400
25 05025 175400
26 05026 054255
27 05027 014253
28 05030 080700
29 05031 054255
30 05032 044253
31 05033 054315
32 05034 050313
33 05035 131470
34 05035 131470
35 05037 137470
36 05000 050000
37 05041 185414
38
39 05045 014253
40 05047 080700
41
42 05050 056305
43

```

```

; MISC TEST OF "ELDA" AND "ESTA"
;
STA01
TSTLOC
JSR #+1
STA 3,HELP
SETUP 100,
JSR #ENTIN
LDA 0,0100
STA 0,STL00
LDA 2,L0MBF
LDA 3,HIGBF
STA 3,STBY0
RAND
JSR #ENTRA
LDA 3,STBY0
ESTA 0,0,2
ESTA 0,0,3
INC 2,2
STA 3,STBY0
DBZ STL00
JMP SA00
LDA 1,0100
STA 1,STL00
LDA 3,L0MBF
LDA 2,HIGBF
ELDA 0,0,3
SAE001 ELDA 1,0,2
ELDA 1,0,2
SUBW 0,1,02R
DSZ STL00
JMP SA000
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE

```

10100 ECL31

```

01
02
03 05051 084401
04 05052 054215
05
06 05053 056305
07 05054 080144
08 05055 050315
09 05056 034313
10 05057 103070
11 05058 080002
12 05061 157470
13 05062 080002
14 05063 177470
15 05065 050002
16 05065 173070
17 05065 080002
18 05067 153070
19 05068 080000
20 05071 157470
21 05072 080000
22 05073 133070
23 05075 080000
24 05075 137470
25 05077 050000
26 05077 112414
27
28 05104 136414
29
30
31 05111 056305
32

```

```

STA01
TSTLOC
JSR #+1
STA 3,HELP
SETUP 100,
JSR #ENTIN
LDA 2,L0MBF
LDA 3,HIGBF
ELEF 0,2,2
ELEF 1,2,3
ELEF 3,2,3
ELEF 2,2,2
ESTA 2,0,2
ESTA 3,0,3
ELDA 2,0,2
ELDA 3,0,3
SUBW 0,2,02R
SUBW 1,3,02R
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE
; (ADDR OF TEST+1) IS
; SAVED IN LOC. HELP
; INITIALIZE TEST.
; MISC TEST OF ELDA,ESTA
; AND ELEF
;

```

```

10102 ECL31
01
02
03
04 05107 004401      STA231
05 05170 054216      J3R      +1
06 05171 006305      STA      3,HELP
07 05172 000144      SETUP   100.
08 05173 000310      J3R #ENTIN
09 05174 111000      RAND
10 05175 176470      J3R #ENTRA
11 05176 000021      MOV     0,2
12 05177 153470      ELEV   3,SAE3=-1,1
13 05178 000000      ESTA   2,0,3
14 05179 154710      XCH    2,3
15 05180 157070      ESTA   3,1,2
16 05181 000001      ELDA   0,0,2
17 05182 125070      ELDA   1,1,2
18 05183 000000      SUB#   0,1,SZR
19 05184 127070      LOOP
20 05185 106414      ERROR
21 05186 000306      J3R #ENTLO
22 05187 000403      JMP    +3
23 05188 000002      .BLK   2
24
25
26
27
28
29

```

```

10101 ECL31
01
02
03
04 05112 004401      STA221
05 05113 054216      J3R      +1
06 05114 006305      STA      3,HELP
07 05115 000144      SETUP   100.
08 05116 000310      J3R #ENTIN
09 05117 000310      RAND
10 05118 040444      J3R #ENTRA
11 05119 040444      STA      0,SAE2
12 05120 100510      XOR     0,0
13 05121 122470      ELDA   0,SAE2=-1,1
14 05122 000041      ESTA
15 05123 142470      ESTA   0,SAE2=-1
16 05124 000040      MOV     0,2
17 05125 110000      MOV     2,3
18 05126 150000      ESTA   2,SAE2=1,-1,1
19 05127 102470      ESTA
20 05128 000030      ESTA   3,SAE2+2,-1,1
21 05129 156470      ELEV   3,SAE2=-1,1
22 05130 000034      ELDA   2,0,3
23 05131 176470      ELDA   1,1,3
24 05132 000027      SUB#   2,1,SZR
25 05133 134470      ERROR
26 05134 000000      ELDA   0,2,3
27 05135 107470      ELEV   2,SAE2=-1,1,1
28 05136 000001      ELDA   3,3,2
29 05137 107470      SUB#   0,3,SZR
30 05138 146414      LOOP
31 05139 123470      ERROR
32 05140 000002      ELDA   0,2,3
33 05141 172470      ELEV   2,SAE2=-1,1,1
34 05142 000012      ELDA   3,3,2
35 05143 137070      SUB#   0,3,SZR
36 05144 000003      LOOP
37 05145 116414      ERROR
38
39
40
41
42
43

```

```

I(ADDR OF TEST+1) IS
I(SAVED IN LOC. HELP
I(INITIALIZE TEST.
I(CCAC0)RANDOM #
I(MISC TEST OF ELDA,ESTA
I(AND ELEV
I(ITERATE TEST ROUTINE

```

```

I(ADDR OF TEST+1) IS
I(SAVED IN LOC. HELP
I(INITIALIZE TEST.
I(CCAC0)RANDOM #
I(MISC TEST OF ELEV,ELDA AND
I(ESTA
I(ITERATE TEST ROUTINE

```

10103 ECL31

```

01
02
03
04
05
06
07 08221 084481
08 08222 084810
09 08223 086314
10
11
12 08224 086305
13 08225 086144
14
15 08226 086310
16 08227 110480
17 08230 144400
18 08231 147370
19 08232 138840
20 08233 188470
21
22 08235 110814
23
24 08242 186414
25
26 08247 181083
27
28
29 08254 086306

```

```

; CHECKING OUT "EJMP" -LONG JMP INSTRUCTION
;

```

```

JMP00:
JSR TSTLOC
STA *+1
JSR @JFILL
JSR @ENTER
SETUP 100.
JSR @ENTIN
100.
RNDADR HIGBF, HIBFU
RAND
JSR @ENTRA
NEG 0,2
NEG 2,1
HLV 1
MOV 1,3
EJMP 1,1
ADDW 0,2,SIZ
ERROR
SUBW 1,3,SIZ
ERROR
MOV 0,8,SNC
ERROR
JSR @ENTLO
ITERATE TEST ROUTINE

```

10104 ECL31

```

01
02
03
04 08265 084401
05 08266 084210
06 08267 086314
07
08
09 08268 086305
10 08269 086144
11
12
13
14 08262 086310
15 08263 086311
16 08264 080313
17 08265 080316
18 08266 111000
19 08267 084207
20 08270 043008
21 08271 156470
22
23 08273 044208
24 08274 040402
25 08275 102070
26
27
28
29 08303 086306

```

```

JMP00:
EJMP,HIGBFAD,JP00
TSTLOC
STA *+1
JSR @JFILL
JSR @ENTER
SETUP 100.
JSR @ENTIN
100.
RNDADR HIGBF, HIBFU
RAND
JSR @ENTRA
HIGBF @RNDADR
HIBFU
MOV 0,2
LOA 1,JMLC1
STA 1,0,2
EJMP 1,JP00,+.2,1
STA 1,JMLC1
STA 0,+.2
EJMP 0,0
ERROR
JSR @ENTLO
ITERATE TEST ROUTINE

```

```

; (ADDR OF TEST+1) IS
; SAVED IN LOC. HELP
; FILL SCRATCH AREA WITH
; JSR @ENTER
; INITIALIZE TEST.
; (AC0)=RANDOM #
; GET ADDRESS IN THE RANGE
; C(HIGBF) AND C(HIBFU)
; (AC0)=AC0+RANDOM ADDR
; (JMP @JMLC1) IS STORED IN,
; RANDOM ADDRESS.
; (ADDR OF (EJMP+3)) IS
; STORED IN JMLC1 FOR
; PROPER RETURN
;

```



```

10106 ECL31
01
02
03
04 05333 004401
05 05334 054216
06 05335 006314
07
08
09 05336 006305
10 05337 000144
11
12
13
14 05340 006310
15 05341 006311
16 05342 000313
17 05343 000316
18 05344 115000
19 05345 024257
20 05346 045400
21 05347 100470
22
23 05351 044260
24 05352 106470
25 05353 000004
26 05354 122400
27 05355 048402
28 05356 102470 JP021
29
30
31
32 05364 006306

```

```

10105 ECL31
01
02
03
04 05330 004401
05 05331 054216
06 05332 006314
07
08
09 05333 006305
10 05334 000144
11
12
13
14 05337 006310
15 05338 006311
16 05339 000313
17 05340 000316
18 05341 115000
19 05342 024257
20 05343 045400
21 05344 100470
22
23 05348 044260
24 05349 106470
25 05350 000004
26 05351 122400
27 05352 048402
28 05353 102470 JP011
29
30
31
32 05361 006306

```

```

10105 ECL31
01
02
03
04 05330 004401
05 05331 054216
06 05332 006314
07
08
09 05333 006305
10 05334 000144
11
12
13
14 05337 006310
15 05338 006311
16 05339 000313
17 05340 000316
18 05341 115000
19 05342 024257
20 05343 045400
21 05344 100470
22
23 05348 044260
24 05349 106470
25 05350 000004
26 05351 122400
27 05352 048402
28 05353 102470 JP011
29
30
31
32 05361 006306

```

```

JMP02:
EJMP2 EJMP,JP02
TSTLOC
JSR 0,1
STA 3,HELP
PJFILL
JSR #ENTER
SETUP 100.
JSR #ENTIN
RNDADR HIGBF,HIBFU
RAND
JSR #ENTRA
RNDADR HIGBF,HIBFU
MOV 0,3
LDA 1,JMLCA
STA 1,0,3
EJMP 1,JP02-->2,1
STA 1,JMLC1
EJMP 1,JP02-->1
SUB 1,0
STA 0,2
EJMP 0,1
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE

```

```

JMP01:
EJMP1 EJMP,LOBFAD,JP01
TSTLOC
JSR 0,1
STA 3,HELP
PJFILL
JSR #ENTER
SETUP 100.
JSR #ENTIN
RNDADR LOWBF,LOBFU
RAND
JSR #ENTRA
RNDADR LOWBF,LOBFU
MOV 0,2
LDA 1,JMLCA
STA 1,0,2
EJMP 1,JP01-->2,1
STA 1,JMLC1
EJMP 0,2
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE

```

```

JMP01:
EJMP1 EJMP,LOBFAD,JP01
TSTLOC
JSR 0,1
STA 3,HELP
PJFILL
JSR #ENTER
SETUP 100.
JSR #ENTIN
RNDADR LOWBF,LOBFU
RAND
JSR #ENTRA
RNDADR LOWBF,LOBFU
MOV 0,2
LDA 1,JMLCA
STA 1,0,2
EJMP 1,JP01-->2,1
STA 1,JMLC1
EJMP 0,2
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE

```

18107 ECL31

|    |        |        |  |  |  |
|----|--------|--------|--|--|--|
| 01 | JMP03: | TSTLOC |  |  |  |
| 02 |        | JSR    |  |  |  |
| 03 | 05365  | 064401 |  |  |  |
| 04 | 05365  | 054210 |  |  |  |
| 05 | 05367  | 056314 |  |  |  |
| 06 |        |        |  |  |  |
| 07 | 05370  | 056385 |  |  |  |
| 08 | 05371  | 056144 |  |  |  |
| 09 |        | 180.   |  |  |  |
| 10 |        | 00     |  |  |  |
| 11 | 05372  | 182470 |  |  |  |
| 12 |        | 058002 |  |  |  |
| 13 |        | ERROR  |  |  |  |
| 14 |        | .ENDC  |  |  |  |
| 15 | 05400  | 182470 |  |  |  |
| 16 |        | 058002 |  |  |  |
| 17 |        | ERROR  |  |  |  |
| 18 | 05406  | 182470 |  |  |  |
| 19 |        | 058002 |  |  |  |
| 20 |        | ERROR  |  |  |  |
| 21 |        | .ENDC  |  |  |  |
| 22 | 05414  | 182470 |  |  |  |
| 23 |        | 058002 |  |  |  |
| 24 |        | ERROR  |  |  |  |
| 25 |        | .ENDC  |  |  |  |
| 26 | 05422  | 182470 |  |  |  |
| 27 |        | 058002 |  |  |  |
| 28 |        | ERROR  |  |  |  |
| 29 |        | .ENDC  |  |  |  |
| 30 | 05430  | 182470 |  |  |  |
| 31 |        | 058002 |  |  |  |
| 32 |        | ERROR  |  |  |  |
| 33 |        | .ENDC  |  |  |  |
| 34 | 05436  | 182470 |  |  |  |
| 35 |        | 058002 |  |  |  |
| 36 |        | ERROR  |  |  |  |
| 37 |        | .ENDC  |  |  |  |
| 38 | 05444  | 182470 |  |  |  |
| 39 |        | 058002 |  |  |  |
| 40 |        | ERROR  |  |  |  |
| 41 |        | .ENDC  |  |  |  |
| 42 | 05452  | 182470 |  |  |  |
| 43 |        | 058002 |  |  |  |
| 44 |        | ERROR  |  |  |  |
| 45 |        | .ENDC  |  |  |  |
| 46 | 05460  | 182470 |  |  |  |
| 47 |        | 058002 |  |  |  |
| 48 |        | ERROR  |  |  |  |
| 49 |        | .ENDC  |  |  |  |
| 50 | 05466  | 182470 |  |  |  |
| 51 |        | 058002 |  |  |  |
| 52 |        | ERROR  |  |  |  |
| 53 |        | .ENDC  |  |  |  |
| 54 | 05474  | 182470 |  |  |  |
| 55 |        | 058002 |  |  |  |
| 56 |        | ERROR  |  |  |  |
| 57 |        | .ENDC  |  |  |  |
| 58 | 05582  | 182470 |  |  |  |
| 59 |        | 058002 |  |  |  |
| 60 |        | ERROR  |  |  |  |

0100 ECL31

|    |            |     |  |  |                      |
|----|------------|-----|--|--|----------------------|
| 01 | ERROR      |     |  |  |                      |
| 02 | .ENDC      |     |  |  |                      |
| 03 | EJMP       |     |  |  |                      |
| 04 |            | 2,1 |  |  | JUMP TO PC OF EJMP+3 |
| 05 | ERROR      |     |  |  |                      |
| 06 | .ENDC      |     |  |  |                      |
| 07 | EJMP       |     |  |  |                      |
| 08 |            | 2,1 |  |  | JUMP TO PC OF EJMP+3 |
| 09 | ERROR      |     |  |  |                      |
| 10 | .ENDC      |     |  |  |                      |
| 11 | EJMP       |     |  |  |                      |
| 12 |            | 2,1 |  |  | JUMP TO PC OF EJMP+3 |
| 13 | ERROR      |     |  |  |                      |
| 14 | .ENDC      |     |  |  |                      |
| 15 | LOOP       |     |  |  |                      |
| 16 | JSR #ENTLO |     |  |  |                      |
| 17 |            |     |  |  | ITERATE TEST ROUTINE |

```

10110 ECL31
01
02
03
04 05365 004401      JMPB51      EJM3      HIBFAD,JPB5,3
05 05366 054216      TSLOC
06 05367 066314      JBR      *+1      ! (ADDR OF TEST+1) IS
07                                JBR      *PJFILL    ! FILL SCRATCH AREA WITH
08                                SETUP 100.      !JSR *ENTER
09 05370 066305      JSR *ENTIN      ! INITIALIZE TEST.
10 05371 066144      100.
11
12
13
14 05372 066310      RNDADR HIGB*,HIBFU
15 05373 066311      RAND
16 05374 066313      JBR *ENTRA      !C(ACB)*RANDOM #
17 05375 066316      HIBF      !GET ADDRESS IN THE RANGE
18 05376 111800      MOV      !C(HIBBF) AND C(HIBFU)
19 05377 024207      LDA      !ACB=AC2*RNDR ADDRESS
20 05380 045900      STA      ! (JMP *JMLC1) IS STORED
21 05381 166470      ELEF     !IN RANDOM ADDRESS
22                                !ADDR OF (EJMP*3) IS STORED
23 05383 044260      STA      !IN JMLC1 FOR GOOD RETURN
24 05384 113222      MOVZR   !ADDR/2)+BIT 15 IS
25 05385 114510      ADI     !STORED IN AC3 AND
26 05386 143370      HLV     !
27 05387 040482      STA      ! (RNDR ADDR/2) IS STORED
28 05318 183470      EJMP    !IN EJMP INSTRUCTION
29                                !
30
31
32 05316 066306      ERROR
      LOOP
      JSR *ENTLO      !ITERATE TEST ROUTINE

```

```

10100 ECL31
01
02
03
04 05333 004401      JMPB41      EJM3      LOBFAD,JPB4,2
05 05334 054216      TSLOC
06 05335 066314      JBR      *+1      ! (ADDR OF TEST+1) IS
07                                JBR      *PJFILL    ! FILL SCRATCH AREA WITH
08                                SETUP 100.      !JSR *ENTER
09 05336 066305      JSR *ENTIN      ! INITIALIZE TEST.
10 05337 066144      100.
11
12
13
14 05338 066310      RNDADR LOBF*,LOBFU
15 05339 066311      RAND
16 05340 066313      JBR *ENTRA      !C(ACB)*RANDOM #
17 05341 066316      LOBF     !GET ADDRESS IN THE RANGE
18 05342 111800      MOV      !C(LOBF) AND C(LOBFU)
19 05343 111800      LDA      !ACB=AC2*RNDR ADDRESS
20 05344 024207      STA      ! (JMP *JMLC1) IS STORED
21 05345 045900      STA      !IN RANDOM ADDRESS
22 05346 166470      ELEF     !IN JMLC1 FOR GOOD RETURN
23 05347 066311      STA      !ADDR/2)+BIT 15 IS
24 05348 044260      MOVZR   !STORED IN AC2 AND
25 05349 113222      ADI     !
26 05350 118910      HLV     ! (RNDR ADDR/2) IS STORED
27 05351 143370      STA      !IN EJMP INSTRUCTION
28 05352 040482      EJMP    !
29                                !
30
31
32 05354 066306      ERROR
      LOOP
      JSR *ENTLO      !ITERATE TEST ROUTINE

```

```

18111 ECL31
01 JMSNBC=1
02 EJMP4
03 TSTLOC
04 JSR *+1
05 STA S,HELP
06 JPFILL
07 JSR CENTER
08 SETUP 100.
09 JSR CENTER
10 100.
11 HIBFAD
12 RNDADR HIGBF,HIBFU
13 RAND
14 JSR CENTER
15 JSR CENTER
16 HIGBF
17 HIBFU
18 STA
19 HIBFAD
20 RNDADR HIGBF,HIBFU
21 RAND
22 JSR CENTER
23 JSR CENTER
24 HIGBF
25 HIBFU
26 LDA
27 SUB#
28 JMP
29 STA
30 MOV
31 LDA
32 STA
33 ELEF
34 STA
35 LDA
36 MOV
37 STA
38 ELEF
39 .NOLOC
40 IDRI
41 STA
42 EJMP
43
44
45
46
47
48
49
50

18112 ECL31
01 JMSNBC=1
02 EJMP4
03 TSTLOC
04 JSR *+1
05 STA S,HELP
06 JPFILL
07 JSR CENTER
08 SETUP 100.
09 JSR CENTER
10 100.
11 HIBFAD
12 RNDADR HIGBF,HIBFU
13 RAND
14 JSR CENTER
15 JSR CENTER
16 HIGBF
17 HIBFU
18 STA
19 HIBFAD
20 RNDADR HIGBF,HIBFU
21 RAND
22 JSR CENTER
23 JSR CENTER
24 HIGBF
25 HIBFU
26 LDA
27 SUB#
28 JMP
29 STA
30 MOV
31 LDA
32 STA
33 ELEF
34 STA
35 LDA
36 MOV
37 STA
38 ELEF
39 .NOLOC
40 IDRI
41 STA
42 EJMP
43
44
45
46
47
48
49
50

18113 ECL31
01 JMSNBC=0
02 EJMP4
03 TSTLOC
04 JSR *+1
05 STA S,HELP
06 JPFILL
07 JSR CENTER
08 SETUP 100.
09 JSR CENTER
10 100.
11 HIBFAD
12 RNDADR HIGBF,HIBFU
13 RAND
14 JSR CENTER
15 JSR CENTER
16 HIGBF
17 HIBFU
18 STA
19 HIBFAD
20 RNDADR HIGBF,HIBFU
21 RAND
22 JSR CENTER
23 JSR CENTER
24 HIGBF
25 HIBFU
26 LDA
27 SUB#
28 JMP
29 STA
30 MOV
31 LDA
32 STA
33 ELEF
34 STA
35 LDA
36 MOV
37 STA
38 ELEF
39 .NOLOC
40 IDRI
41 STA
42 EJMP
43
44
45
46
47
48
49
50

18114 ECL31
01 JMSNBC=0
02 EJMP4
03 TSTLOC
04 JSR *+1
05 STA S,HELP
06 JPFILL
07 JSR CENTER
08 SETUP 100.
09 JSR CENTER
10 100.
11 HIBFAD
12 RNDADR HIGBF,HIBFU
13 RAND
14 JSR CENTER
15 JSR CENTER
16 HIGBF
17 HIBFU
18 STA
19 HIBFAD
20 RNDADR HIGBF,HIBFU
21 RAND
22 JSR CENTER
23 JSR CENTER
24 HIGBF
25 HIBFU
26 LDA
27 SUB#
28 JMP
29 STA
30 MOV
31 LDA
32 STA
33 ELEF
34 STA
35 LDA
36 MOV
37 STA
38 ELEF
39 .NOLOC
40 IDRI
41 STA
42 EJMP
43
44
45
46
47
48
49
50

```

```

CHECKING SINGLE LEVEL INDIRECT "EJMP"
JMSNBC=0
EJMP4 EJMP,HIBFAD,JPCB,LOBFAD,0,JPCB0
TSTLOC
JSR *+1
STA S,HELP
JPFILL
JSR CENTER
SETUP 100.
JSR CENTER
100.
HIBFAD
RNDADR HIGBF,HIBFU
RAND
JSR CENTER
JSR CENTER
HIGBF
HIBFU
STA
LOBFAD
RNDADR LOBF,HIBFU
RAND
JSR CENTER
JSR CENTER
LOBF
LOBFU
LDA
SUB#
JMP
STA
MOV
LDA
RNDADR
STORE ADDR OF (EJMP+3) IN
JPCB0=+2.1
JMLC1
JMLC2
JMLC3
JPCB0
JPCB1
JPCB2
JPCB3
JPCB4
JPCB5
JPCB6
JPCB7
JPCB8
JPCB9
JPCB10
JPCB11
JPCB12
JPCB13
JPCB14
JPCB15
JPCB16
JPCB17
JPCB18
JPCB19
JPCB20
JPCB21
JPCB22
JPCB23
JPCB24
JPCB25
JPCB26
JPCB27
JPCB28
JPCB29
JPCB30
JPCB31
JPCB32
JPCB33
JPCB34
JPCB35
JPCB36
JPCB37
JPCB38
JPCB39
JPCB40
JPCB41
JPCB42
JPCB43
JPCB44
JPCB45
JPCB46
JPCB47
JPCB48
JPCB49
JPCB50

```

```

JMSNBC=1
EJMP4
TSTLOC
JSR *+1
STA S,HELP
JPFILL
JSR CENTER
SETUP 100.
JSR CENTER
100.
HIBFAD
RNDADR HIGBF,HIBFU
RAND
JSR CENTER
JSR CENTER
HIGBF
HIBFU
STA
HIBFAD
RNDADR HIGBF,HIBFU
RAND
JSR CENTER
JSR CENTER
HIGBF
HIBFU
STA
SUB#
JMP
STA
MOV
LDA
RNDADR
STORE ADDR OF (EJMP+3) IN
JPCB1=+2.1
JMLC1
JMLC2
JMLC3
JPCB1
JPCB2
JPCB3
JPCB4
JPCB5
JPCB6
JPCB7
JPCB8
JPCB9
JPCB10
JPCB11
JPCB12
JPCB13
JPCB14
JPCB15
JPCB16
JPCB17
JPCB18
JPCB19
JPCB20
JPCB21
JPCB22
JPCB23
JPCB24
JPCB25
JPCB26
JPCB27
JPCB28
JPCB29
JPCB30
JPCB31
JPCB32
JPCB33
JPCB34
JPCB35
JPCB36
JPCB37
JPCB38
JPCB39
JPCB40
JPCB41
JPCB42
JPCB43
JPCB44
JPCB45
JPCB46
JPCB47
JPCB48
JPCB49
JPCB50

```

```

JMSNBC=0
EJMP4
TSTLOC
JSR *+1
STA S,HELP
JPFILL
JSR CENTER
SETUP 100.
JSR CENTER
100.
HIBFAD
RNDADR HIGBF,HIBFU
RAND
JSR CENTER
JSR CENTER
HIGBF
HIBFU
STA
LOBFAD
RNDADR LOBF,HIBFU
RAND
JSR CENTER
JSR CENTER
LOBF
LOBFU
LDA
SUB#
JMP
STA
MOV
LDA
RNDADR
STORE ADDR OF (EJMP+3) IN
JPCB0=+2.1
JMLC1
JMLC2
JMLC3
JPCB0
JPCB1
JPCB2
JPCB3
JPCB4
JPCB5
JPCB6
JPCB7
JPCB8
JPCB9
JPCB10
JPCB11
JPCB12
JPCB13
JPCB14
JPCB15
JPCB16
JPCB17
JPCB18
JPCB19
JPCB20
JPCB21
JPCB22
JPCB23
JPCB24
JPCB25
JPCB26
JPCB27
JPCB28
JPCB29
JPCB30
JPCB31
JPCB32
JPCB33
JPCB34
JPCB35
JPCB36
JPCB37
JPCB38
JPCB39
JPCB40
JPCB41
JPCB42
JPCB43
JPCB44
JPCB45
JPCB46
JPCB47
JPCB48
JPCB49
JPCB50

```

```

JMSNBC=0
EJMP4
TSTLOC
JSR *+1
STA S,HELP
JPFILL
JSR CENTER
SETUP 100.
JSR CENTER
100.
HIBFAD
RNDADR HIGBF,HIBFU
RAND
JSR CENTER
JSR CENTER
HIGBF
HIBFU
STA
LOBFAD
RNDADR LOBF,HIBFU
RAND
JSR CENTER
JSR CENTER
LOBF
LOBFU
LDA
SUB#
JMP
STA
MOV
LDA
RNDADR
STORE ADDR OF (EJMP+3) IN
JPCB0=+2.1
JMLC1
JMLC2
JMLC3
JPCB0
JPCB1
JPCB2
JPCB3
JPCB4
JPCB5
JPCB6
JPCB7
JPCB8
JPCB9
JPCB10
JPCB11
JPCB12
JPCB13
JPCB14
JPCB15
JPCB16
JPCB17
JPCB18
JPCB19
JPCB20
JPCB21
JPCB22
JPCB23
JPCB24
JPCB25
JPCB26
JPCB27
JPCB28
JPCB29
JPCB30
JPCB31
JPCB32
JPCB33
JPCB34
JPCB35
JPCB36
JPCB37
JPCB38
JPCB39
JPCB40
JPCB41
JPCB42
JPCB43
JPCB44
JPCB45
JPCB46
JPCB47
JPCB48
JPCB49
JPCB50

```

|       |        |                                 |                     |                                 |        |       |        |                                 |        |
|-------|--------|---------------------------------|---------------------|---------------------------------|--------|-------|--------|---------------------------------|--------|
| 10113 | ECL31  | JMSKc2                          | JMPC21              | JMSKc3                          | JMPC31 | 10114 | ECL31  | JMSKc3                          | JMPC31 |
| 01    | 000002 | EJMP,LOBPAD,JPC2,HISFAD,2,JPC20 | EJMP4               | EJMP,HISFAD,JPC3,LOBFAD,3,JPC30 | 000003 | 01    | 000003 | EJMP,HISFAD,JPC3,LOBFAD,3,JPC30 | 000003 |
| 02    |        | TSTLOC                          | TSTLOC              | TSTLOC                          |        | 02    |        | TSTLOC                          |        |
| 03    |        | (ADDR OF TEST+3) IS             | (ADDR OF TEST+3) IS | (ADDR OF TEST+3) IS             |        | 03    |        | (ADDR OF TEST+3) IS             |        |
| 04    | 05730  | S*HELP                          | S*HELP              | S*HELP                          |        | 04    | 05730  | S*HELP                          |        |
| 05    | 05731  | 0*FILL                          | 0*FILL              | 0*FILL                          |        | 05    | 05731  | 0*FILL                          |        |
| 06    | 05732  | 0*FILL                          | 0*FILL              | 0*FILL                          |        | 06    | 05732  | 0*FILL                          |        |
| 07    |        | SETUP 100.                      | SETUP 100.          | SETUP 100.                      |        | 07    |        | SETUP 100.                      |        |
| 08    | 05733  | JSR 0ENTIN                      | JSR 0ENTIN          | JSR 0ENTIN                      |        | 08    | 05733  | JSR 0ENTIN                      |        |
| 09    | 05734  | LOBFAD                          | LOBFAD              | LOBFAD                          |        | 09    | 05734  | LOBFAD                          |        |
| 10    | 05735  | RNDADR LOWBF,LOBFU              | RNDADR LOWBF,LOBFU  | RNDADR HISBF,HISFU              |        | 10    | 05735  | RNDADR HISBF,HISFU              |        |
| 11    |        | RAND                            | RAND                | RAND                            |        | 11    |        | RAND                            |        |
| 12    |        | JSR 0ENTRA                      | JSR 0ENTRA          | JSR 0ENTRA                      |        | 12    |        | JSR 0ENTRA                      |        |
| 13    | 05736  | 0*RNADR                         | 0*RNADR             | 0*RNADR                         |        | 13    | 05736  | 0*RNADR                         |        |
| 14    | 05737  | LOBF                            | LOBF                | HISBF                           |        | 14    | 05737  | LOBF                            |        |
| 15    | 05738  | LOBFU                           | LOBFU               | HISFU                           |        | 15    | 05738  | LOBFU                           |        |
| 16    | 05740  | STA 0,JMLC2                     | STA 0,JMLC2         | STA 0,JMLC2                     |        | 16    | 05740  | STA 0,JMLC2                     |        |
| 17    | 05741  | HISFAD                          | HISFAD              | LOBFAD                          |        | 17    | 05741  | HISFAD                          |        |
| 18    |        | RNDADR LOWBF,LOBFU              | RNDADR LOWBF,LOBFU  | RNDADR LOWBF,LOBFU              |        | 18    |        | RNDADR LOWBF,LOBFU              |        |
| 19    |        | RAND                            | RAND                | RAND                            |        | 19    |        | RAND                            |        |
| 20    |        | JSR 0ENTRA                      | JSR 0ENTRA          | JSR 0ENTRA                      |        | 20    |        | JSR 0ENTRA                      |        |
| 21    | 05742  | 0*RNADR                         | 0*RNADR             | 0*RNADR                         |        | 21    | 05742  | 0*RNADR                         |        |
| 22    | 05743  | LOBF                            | LOBF                | LOBF                            |        | 22    | 05743  | LOBF                            |        |
| 23    | 05744  | LOBFU                           | LOBFU               | LOBFU                           |        | 23    | 05744  | LOBFU                           |        |
| 24    | 05745  | LOA                             | LOA                 | LOA                             |        | 24    | 05745  | LOA                             |        |
| 25    | 05746  | 1,JMLC2                         | 1,JMLC2             | 1,JMLC2                         |        | 25    | 05746  | 1,JMLC2                         |        |
| 26    | 05747  | 1,0,0NR                         | 1,0,0NR             | 1,0,0NR                         |        | 26    | 05747  | 1,0,0NR                         |        |
| 27    | 05748  | JPC20                           | JPC20               | JPC30                           |        | 27    | 05748  | JPC30                           |        |
| 28    | 05750  | 0,JMLC2                         | 0,JMLC2             | 0,JMLC2                         |        | 28    | 05750  | 0,JMLC2                         |        |
| 29    | 05751  | 0*RNADR                         | 0*RNADR             | 0*RNADR                         |        | 29    | 05751  | 0*RNADR                         |        |
| 30    | 05752  | LOBF                            | LOBF                | LOBF                            |        | 30    | 05752  | LOBF                            |        |
| 31    | 05753  | LOBFU                           | LOBFU               | LOBFU                           |        | 31    | 05753  | LOBFU                           |        |
| 32    | 05754  | STA 1,0,2                       | STA 1,0,2           | STA 1,0,2                       |        | 32    | 05754  | STA 1,0,2                       |        |
| 33    | 05755  | 1,JPC20,+02,1                   | 1,JPC20,+02,1       | 1,JPC30,+02,1                   |        | 33    | 05755  | 1,JPC30,+02,1                   |        |
| 34    |        | STA                             | STA                 | STA                             |        | 34    |        | STA                             |        |
| 35    | 05757  | 1,JMLC1                         | 1,JMLC1             | 1,JMLC1                         |        | 35    | 05757  | 1,JMLC1                         |        |
| 36    | 05758  | 0*RNADR                         | 0*RNADR             | 0*RNADR                         |        | 36    | 05758  | 0*RNADR                         |        |
| 37    | 05761  | 1,1,022                         | 1,1,022             | 1,1,022                         |        | 37    | 05761  | 1,1,022                         |        |
| 38    | 05762  | 1,1,022                         | 1,1,022             | 1,1,022                         |        | 38    | 05762  | 1,1,022                         |        |
| 39    | 05763  | 1,4,3,37                        | 1,4,3,37            | 1,4,3,37                        |        | 39    | 05763  | 1,4,3,37                        |        |
| 40    |        | NOLOC                           | NOLOC               | NOLOC                           |        | 40    |        | NOLOC                           |        |
| 41    | 05764  | 103770                          | 103770              | 103770                          |        | 41    | 05764  | 103770                          |        |
| 42    |        | STA                             | STA                 | STA                             |        | 42    |        | STA                             |        |
| 43    | 05766  | 0*RNADR                         | 0*RNADR             | 0*RNADR                         |        | 43    | 05766  | 0*RNADR                         |        |
| 44    | 05767  | 103770                          | 103770              | 103770                          |        | 44    | 05767  | 103770                          |        |
| 45    |        | ERROR                           | ERROR               | ERROR                           |        | 45    |        | ERROR                           |        |
| 46    |        | LOOP                            | LOOP                | LOOP                            |        | 46    |        | LOOP                            |        |
| 47    |        | JSR 0ENTLO                      | JSR 0ENTLO          | JSR 0ENTLO                      |        | 47    |        | JSR 0ENTLO                      |        |
| 48    | 05775  | 006306                          | 006306              | 006306                          |        | 48    | 05775  | 006306                          |        |

ITERATE TEST ROUTINE

ITERATE TEST ROUTINE

ITERATE TEST ROUTINE

ITERATE TEST ROUTINE

ITERATE TEST ROUTINE

ITERATE TEST ROUTINE

0115 ECL31

```
01 00000 JMP001  ; CHECKING MULTI-LEVEL INDIRECT *EJMP*
02 00044 00401   ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
03 00045 05421   ; TSTLOC  ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
04 00046 000314  ; JBR *+1 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
05 00047 000305  ; STA 3,HELP ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
06 00048 00014   ; JBR *JFILL ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
07 00049 00010   ; SETUP 100. ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
08 00050 00010   ; JBR *ENTIN ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
09 00051 000310  ; 100. ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
10 00052 000311  ; RNDADR LONBF,LOBFU ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
11 00053 000315  ; RAND ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
12 00054 000312  ; JBR *ENTRA ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
13 00055 000310  ; JBR *ENTRA ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
14 00056 000311  ; LONBF *RNADR ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
15 00057 000315  ; LONBF ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
16 00058 000312  ; STA 0,JMLC2 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
17 00059 000310  ; HISFAD ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
18 00060 000311  ; RNDADR HISBP,HISBU ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
19 00061 000315  ; RAND ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
20 00062 000310  ; JBR *ENTRA ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
21 00063 000311  ; HISBP *RNADR ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
22 00064 000315  ; LDA 1,JMLC2 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
23 00065 000312  ; SUB# 1,0,SNR ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
24 00066 000310  ; JBR *JMLC2 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
25 00067 000311  ; JMP JPD00 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
26 00068 000315  ; STA 0,JMLC3 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
27 00069 000310  ; TORI 1000000,0 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
28 00070 000311  ; STA 0,JMLC2 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
29 00071 000315  ; HISBP *RNADR ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
30 00072 000310  ; LDA 1,JMLC2 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
31 00073 000315  ; SUB# 1,0,SNR ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
32 00074 000312  ; JBR *JMLC3 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
33 00075 000310  ; STA 0,JMLC2 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
34 00076 000311  ; TORI 1000000,0 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
35 00077 000315  ; STA 0,JMLC2 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
36 00078 000310  ; RNDADR LONBF,LOBFU ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
37 00079 000311  ; RAND ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
38 00080 000315  ; JBR *ENTRA ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
39 00081 000310  ; HISBP *RNADR ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
40 00082 000315  ; LDA 1,JMLC3 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
41 00083 000312  ; SUB# 0,1,SNR ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
42 00084 000310  ; JBR *JMLC3 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
43 00085 000311  ; MOV 0,2 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
44 00086 000315  ; LDA 1,JMLC2 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
45 00087 000312  ; LDA 1,JMLC2 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
46 00088 000310  ; JBR *JMLC2 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
47 00089 000311  ; STA 0,JMLC3 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
48 00090 000315  ; LDA 1,JMLC2 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
49 00091 000310  ; JBR *JMLC2 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
50 00092 000311  ; STA 0,JMLC3 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
51 00093 000315  ; MOV 0,2 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
52 00094 000310  ; LDA 1,JMLC2 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
53 00095 000311  ; LDA 1,JMLC2 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
54 00096 000315  ; STA 1,0,2 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
55 00097 000310  ; ELEV 1,JP000,+,R,1 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
56 00098 000311  ; STA 1,JMLC1 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
57 00099 000315  ; LDA 0,JMLC2 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
58 00100 000310  ; *NOLOC 0 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
59 00101 000311  ; TORI 1000000,0 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
60 00102 000315  ; TORI 1000000,0 ; JMSNCS8 EJMP,LOBFAD,JP001,HISFAD,0,LOBFAD,JP000,JP001
```

```
0116 ECL31
01 1000000
02 00115 040482  STA 0,+,2
03 00116 102070  JMP001
04 00117 1000000
05 00118 1000000
06 00119 1000000
07 00120 000306  ; ITERATE TEST ROUTINE
```

```
0116 ECL31
01 1000000
02 00115 040482  STA 0,+,2
03 00116 102070  JMP001
04 00117 1000000
05 00118 1000000
06 00119 1000000
07 00120 000306  ; ITERATE TEST ROUTINE
```

```
0116 ECL31
01 1000000
02 00115 040482  STA 0,+,2
03 00116 102070  JMP001
04 00117 1000000
05 00118 1000000
06 00119 1000000
07 00120 000306  ; ITERATE TEST ROUTINE
```

```

0116 ECL31
01 100000
02
03
04 06210 000306
ITERATE TEST ROUTINE

ERROR
LOAD
JSR #ENTLO

10117 ECL31
000001 JPD11
JMSK=1
EJMP5 EJM#HISFAD,JPD1,HISFAD,1,HISFAD,JPD10,JPD11
TSYLOC
JSR #*1
STA S,HELP
JSR #JFILL
SETUP 100,
JSR #ENTIN,
100,
HISFAD
RNDADR HISGF,HISFU
RAND
JSR #ENTRA
JSR #ORADR
HISGF
HISFU
STA #JMLC2
HISFAD
RNDADR HISGF,HISFU
RAND
JSR #ENTRA
JSR #ORADR
HISGF
HISFU
LDA 1,JMLC2
SUB# 1,0,SNR
JMP JPD10
STA #JMLC3
LORI 100000,0
STA #JMLC2
HISFAD
RNDADR HISGF,HISFU
RAND
JSR #ENTRA
JSR #ORADR
HISGF
HISFU
LDA 1,JMLC3
SUB# 1,0,SNR
JMP JPD11
STA #JMLC3
MOV #0
LDA 1,JMLC3
STA 1,0,2
EJMP 1,JPD10-0,1
STA 1,JMLC1
LDA #JMLC2
EJMP 1,JPD10-0,1
SUB 1,0
NOLOC
LORI 100000,0
STA #*2
EJMP #0,1

01 000001 JPD11
02
03
04 06125 004081
05 06126 004016
06 06127 006314
07
08
09 06130 006305
10 06131 006144
11
12
13
14 06132 006310
15 06133 006311
16 06134 006313
17 06135 006310
18 06136 006261
19
20
21 06137 006310
22 06140 006311
23 06141 006313
24 06142 006310
25 06143 004261
26 06144 102415
27 06145 00072
28 06146 00252
29 06147 103770
30 06148 100000
31 06151 042261
32
33
34
35
36 06152 006310
37 06153 006311
38 06154 006313
39 06155 006310
40 06156 004262
41 06157 106415
42 06158 00072
43 06161 004261
44 06162 102415
45 06163 00072
46 06164 042262
47 06165 111000
48 06166 004262
49 06167 00000
50 06170 106470
51 060014
52 06172 044260
53 06173 026261
54 06174 106470
55 06176 102400
56 060000
57 06177 100770
58 100000
59 06001 040402
60 06002 106470 JPD11

```

0120 ECL31  
01 06266 180878 JPD21  
02  
03  
04  
05 06274 066306  
EJMP 00.2  
ERROR  
LOOP  
JSR 0ENTLO  
ITERATE TEST ROUTINE

08110 ECL31  
000002 JPD21: JMSMC=2  
EJMP  
TSTLOC  
JSR  
STA  
JSR  
\*+1  
S,HELP  
\*FILL  
\*ENTER  
\*INITIALIZE TEST.  
JPD20: L0SFAD, JPD08, L0SFAD, JPD08, JPD21  
\*+1  
S,HELP  
\*FILL  
\*ENTER  
\*INITIALIZE TEST.  
JPD21: L0SFAD, JPD08, L0SFAD, JPD08, JPD21  
\*+1  
S,HELP  
\*FILL  
\*ENTER  
\*INITIALIZE TEST.

01 06266 180878 JPD21  
02  
03  
04  
05 06274 066306  
EJMP 00.2  
ERROR  
LOOP  
JSR 0ENTLO  
ITERATE TEST ROUTINE

```

01 06266 180878 JPD21
02
03
04
05 06274 066306
EJMP 00.2
ERROR
LOOP
JSR 0ENTLO
ITERATE TEST ROUTINE

08110 ECL31
000002 JPD21: JMSMC=2
EJMP
TSTLOC
JSR
STA
JSR
*+1
S,HELP
*FILL
*ENTER
*INITIALIZE TEST.
JPD20: L0SFAD, JPD08, L0SFAD, JPD08, JPD21
*+1
S,HELP
*FILL
*ENTER
*INITIALIZE TEST.
JPD21: L0SFAD, JPD08, L0SFAD, JPD08, JPD21
*+1
S,HELP
*FILL
*ENTER
*INITIALIZE TEST.
01 06266 180878 JPD21
02
03
04
05 06274 066306
EJMP 00.2
ERROR
LOOP
JSR 0ENTLO
ITERATE TEST ROUTINE

```

01 06266 180878 JPD21  
02  
03  
04  
05 06274 066306  
EJMP 00.2  
ERROR  
LOOP  
JSR 0ENTLO  
ITERATE TEST ROUTINE



```

10121 ECL31
01 00000 JMPD31
02 EJMP #3
03 TS1LOC
04 00275 00401
05 00276 054210
06 00277 006314
07
08
09 00300 006305
10 00301 006104
11
12
13
14 00302 006310
15 00303 006311
16 00304 000313
17 00305 000310
18 00306 000301
19
20
21
22 00307 006310
23 00310 006311
24 00311 000310
25 00312 000310
26 00313 004901
27 00314 102415
28 00315 000772
29 00316 040020
30 00317 103770
31 100000
32 00321 042201
33
34
35
36 00332 006310
37 00333 006311
38 00334 000310
39 00335 000312
40 00336 024002
41 00337 100415
42 00338 000772
43 00331 024201
44 00332 102415
45 00333 000772
46 00334 042202
47 00335 110000
48 00336 024002
49 00337 040000
50 00340 100470
51
52 00342 044000
53 00343 000310
54 00344 110022
55 00345 114010
56 00346 143370
57 00347 000000
58 00347 103770
59 100000
60 00351 040402

JMSNC=3
EJMP #HISFAD,JPD31,LOBFAD,3,LOBFAD,JPD30,JPD31
TS1LOC
JSR #1
JRD 3,HELP
JRD #FILL
JRD #ENTER
SETUP 100.
JRD #ENTIN
100.
HISFAD
RNDADR HISOF,HISFU
RND
JRD #ENTRA
JRD #RNADR
HISBF
HISFU
STA 0,JMLC2
LOBFAD
RNDADR LOHBF,LOBFU
RND
JRD #ENTRA
JRD #RNADR
LOBBF
LOBFU
LDA 1,JMLC2
SUBW 1,0,SNR
JRD #JPD3B
STA 0,JMLC3
TORI 100000.0
STA 0,JMLC2
RNDADR LOHBF,LOBFU
RND
JRD #ENTRA
JRD #RNADR
LOBBF
LOBFU
LDA 1,JMLC3
SUBW 1,0,SNR
JRD #JPD31
JRD #JMLC2
LDA 1,JMLC2
SUBW 1,0,SNR
JRD #JPD31
JRD #JMLC3
MOV #02
LDA 1,JMLC0
STA 1,0,2
ELEV 1,JPD30,-02,1
STA 1,JMLC1
LDA 0,JMLC2
MOVW 0,J,02C 1(RNDH
ADI 0
MOV 0
INLOC 0
TORI 100000.0
STA 0,J,+2

0122 ECL31
01 00352 103470 JPD31
02 100000
03
04
05 00350 006305
ITERATE TEST ROUTINE

EJMP #0,3
ERROR
LOOP
JRD #ENTLO

```

18123 ECL31

```

01
02
03
04
05
06
07 06361 084401
08 06362 054216
09 06363 086314
10
11
12 06364 086305
13 06365 086144
14
15 06366 086316
16 06367 119400
17 06370 144400
18 06371 176048
19 06372 106470
20 06373 086001
21 06374 113614
22
23 06401 147614
24
25 06406 181903
26
27 06413 174815
28
29
30 06420 086306
31

```

```

; JSRAB:
; CHECKING OUT "EJSR" - LONG JSR INSTRUCTION
;

```

```

TSTLOC
JSR      ;+1
STA      3,HELP
JSR      0,JFILL
100.
SETUP 100.
JSR #ENTN
RANO
JSR #ENTRA
NEG 0,2
NEG 2,1
ADDO 3,3
EJSR 1,1
ADD# 0,2,SZR
ERROR
ADD# 2,1,SZR
ERROR
MOV 0,0,SNR
ERROR
COM# 3,3,SNR
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE

```

18124 ECL31

```

01
02 06421 084401
03 06422 054216
04 06423 086314
05
06
07 06424 086305
08 06425 086144
09 06426 176000
10 06427 182470
11 06428 086003
12 06431 186470 JRA11
13 06433 086001
14 06433 116414
15
16
17 06440 086306
18

```

```

TSTLOC
JSR      ;+1
STA      3,HELP
JSR      0,JFILL
100.
SETUP 100.
JSR #ENTN
ADDO 3,3
EJSR 0,JRA1-;+1,1
EJSR 1,1
SUB# 0,3,SZR
ERROR
LOOP
JSR #ENTLO
ITERATE TEST ROUTINE

```

```

; (ADDR OF TEST+1) IS
; SAVED IN LOC. HELP
; FILL SCRATCH AREA WITH
; JSR #ENTER
; INITIALIZE TEST.
; AFTER EXECUTING EJSR,
; FAC3 MUST BE EQ. TO
; FADDR OF (JRA1+2)
;

```

```

18126 ECL31
01
02
03
04 06441 064401
05 06442 064210
06 06443 066314
07
08
09 06444 066305
10 06445 066144
11
12
13
14 06446 066318
15 06447 066311
16 06448 066315
17 06451 066312
18 06452 111000
19 06453 024257
20 06454 045000
21 06455 166470
22 06456 066065
23 06457 044258
24 06460 048482
25 06461 166670
26
27
28
29 06467 066306
30
31

JNR001
EJMP1 EJSR,LOSFAD,JR00
TSTLOC
JBR
STA 3,HELP
JBR 0,JFILL
SETUP 100,
JBR 0,ENTIN
100,
LOSFAD
RNDADR LOSF,LOSFU
RAND
JBR 0,ENTRA
JBR 0,RNADR
LOSF
LOSFU
MOV 0,2
LDA 1,JMLCR
STA 1,0,2
ELEF 1,JR00,-2,1
STA 1,JMLCI
STA 0,0,2
EJSR 0,0
ERROR
LOOP
JBR 0,ENTLO

JNR011
EJMP2 EJSR,JR01
TSTLOC
JBR
STA 3,HELP
JBR 0,JFILL
SETUP 100,
JBR 0,ENTIN
100,
RNDADR MIBGF,MIBFU
RAND
JBR 0,ENTRA
JBR 0,RNADR
MIBGF
MIBFU
MOV 0,3
LDA 1,JMLCR
STA 1,0,3
ELEF 1,JR01,-2,1
STA 1,JMLCI
ELEF 1,JR01,-2,1
SUB 1,0
STA 0,0,2
EJSR 0,1
ERROR
LOOP
JBR 0,ENTLO

JNR013
J(ADDR OF TEST+1) IS
J(SAVED IN LOC. HELP
J(FILL SCRATCH AREA WITH
J(JSR 0,CENTER
J(INITIALIZE TEST.
J(C(AC0)=RANDOM #
J(GET ADDRESS IN THE RANGE
J(C(MIBGF) AND C(MIBFU)
J(AC0=ACS=RANDOM ADDRESS
J(STORE (JMP 0,JMLCI) IN
J(RANDOM ADDRESS
J(ADDR OF (EJSR+3) IS STORED
J(IN JMLCI FOR PROPER RETURN
J(RANDOM ADDR= PC OF EJSR+1)
J(THIS STORED IN EJSR
J(INSTRUCTION
J(ITERATE TEST ROUTINE

```

10187 ECL31

JR821

EJRS, HIBFAD, JR82.2

```

01 00000 000000
02 78TLOC
03 JSR *+1
04 STA 3*HELP
05 JSR *JFILL
06
07 SETUP 100.
08 JSR *ENTIN
09 HIBFAD
10
11 RNDADR HIBF, HIBFU
12 RAND
13 JSR *ENTRA
14 JSR *RNADR
15 HIBF
16 HIBFU
17 MOV 0,2
18 LDA 1,JMLCR
19 STA 1,0,2
20 ELEM 1,JR82,+,2,1
21
22 STA 1,JMLC1
23 MOVZR 0,2,3ZC I(RNDM
24 ADI 1,2
25 HLV 0
26 STA 0,+,2
27 EJRS
28
29 ERROR
30 LOOP
31 JSR *ENTLO
ITERATE TEST ROUTINE

```

10188 ECL31

JR831

EJRS, LOBFAD, JR83.3

```

01 00000 000000
02 78TLOC
03 JSR *+1
04 STA 3*HELP
05 JSR *JFILL
06
07 SETUP 100.
08 JSR *ENTIN
09 LOBFAD
10
11 RNDADR LOBF, LOBFU
12 RAND
13 JSR *ENTRA
14 JSR *RNADR
15 LOBF
16 LOBFU
17 MOV 0,2
18 LDA 1,JMLCR
19 STA 1,0,2
20 ELEM 1,JR83,+,2,1
21
22 STA 1,JMLC1
23 MOVZR 0,3,3ZC I(RNDM
24 ADI 1,3
25 HLV 0
26 STA 0,+,2
27 EJRS
28
29 ERROR
30 LOOP
31 JSR *ENTLO
ITERATE TEST ROUTINE

```

10189 ECL31

JR831

EJRS, LOBFAD, JR83.3

```

01 00000 000000
02 78TLOC
03 JSR *+1
04 STA 3*HELP
05 JSR *JFILL
06
07 SETUP 100.
08 JSR *ENTIN
09 LOBFAD
10
11 RNDADR LOBF, LOBFU
12 RAND
13 JSR *ENTRA
14 JSR *RNADR
15 LOBF
16 LOBFU
17 MOV 0,2
18 LDA 1,JMLCR
19 STA 1,0,2
20 ELEM 1,JR83,+,2,1
21
22 STA 1,JMLC1
23 MOVZR 0,3,3ZC I(RNDM
24 ADI 1,3
25 HLV 0
26 STA 0,+,2
27 EJRS
28
29 ERROR
30 LOOP
31 JSR *ENTLO
ITERATE TEST ROUTINE

```



|             |        |       |         |                                   |             |        |       |         |                                   |
|-------------|--------|-------|---------|-----------------------------------|-------------|--------|-------|---------|-----------------------------------|
| 10131 ECL31 | 000002 | JRC2: | JMSHC#2 | EJSR,LOBFAD,JRC2,LOBFAD,2,JRC20   | 10132 ECL31 | 000003 | JRC3: | JMSHC#3 | EJSR,HIBFAD,JRC3,LOBFAD,3,JRC30   |
| 01          | 000002 | JRC2: | EJMPA   |                                   | 01          | 000003 | JRC3: | EJMPA   |                                   |
| 02          | 000002 | JRC2: | TSTLOC  |                                   | 02          | 000003 | JRC3: | TSTLOC  |                                   |
| 03          | 000002 | JRC2: | JSR     | +1                                | 03          | 000003 | JRC3: | JSR     | +1                                |
| 04          | 000002 | JRC2: | STA     | 3,HELP                            | 04          | 000003 | JRC3: | STA     | 3,HELP                            |
| 05          | 000002 | JRC2: | JSR     | 0,FILL                            | 05          | 000003 | JRC3: | JSR     | 0,FILL                            |
| 06          | 000002 | JRC2: | SETUP   | 100,                              | 06          | 000003 | JRC3: | SETUP   | 100,                              |
| 07          | 000002 | JRC2: | JSR     | 0,ENTIN                           | 07          | 000003 | JRC3: | JSR     | 0,ENTIN                           |
| 08          | 000002 | JRC2: | LOBFAD  |                                   | 08          | 000003 | JRC3: | LOBFAD  |                                   |
| 09          | 000002 | JRC2: | RNDADR  | LCHBF,LOBFU                       | 09          | 000003 | JRC3: | RNDADR  | LCHBF,LOBFU                       |
| 10          | 000002 | JRC2: | RAND    |                                   | 10          | 000003 | JRC3: | RAND    |                                   |
| 11          | 000002 | JRC2: | JSR     | 0,ENTRA                           | 11          | 000003 | JRC3: | JSR     | 0,ENTRA                           |
| 12          | 000002 | JRC2: | JSR     | 0,RNADR                           | 12          | 000003 | JRC3: | JSR     | 0,RNADR                           |
| 13          | 000002 | JRC2: | LOBF    |                                   | 13          | 000003 | JRC3: | LOBF    |                                   |
| 14          | 000002 | JRC2: | STA     | 0,JMLC2                           | 14          | 000003 | JRC3: | STA     | 0,JMLC2                           |
| 15          | 000002 | JRC2: | LOBFAD  |                                   | 15          | 000003 | JRC3: | LOBFAD  |                                   |
| 16          | 000002 | JRC2: | RNDADR  | LCHBF,LOBFU                       | 16          | 000003 | JRC3: | RNDADR  | LCHBF,LOBFU                       |
| 17          | 000002 | JRC2: | RAND    |                                   | 17          | 000003 | JRC3: | RAND    |                                   |
| 18          | 000002 | JRC2: | JSR     | 0,ENTRA                           | 18          | 000003 | JRC3: | JSR     | 0,ENTRA                           |
| 19          | 000002 | JRC2: | JSR     | 0,RNADR                           | 19          | 000003 | JRC3: | JSR     | 0,RNADR                           |
| 20          | 000002 | JRC2: | LOBF    |                                   | 20          | 000003 | JRC3: | LOBF    |                                   |
| 21          | 000002 | JRC2: | STA     | 0,JMLC2                           | 21          | 000003 | JRC3: | STA     | 0,JMLC2                           |
| 22          | 000002 | JRC2: | LOBFAD  |                                   | 22          | 000003 | JRC3: | LOBFAD  |                                   |
| 23          | 000002 | JRC2: | RNDADR  | LCHBF,LOBFU                       | 23          | 000003 | JRC3: | RNDADR  | LCHBF,LOBFU                       |
| 24          | 000002 | JRC2: | RAND    |                                   | 24          | 000003 | JRC3: | RAND    |                                   |
| 25          | 000002 | JRC2: | JSR     | 0,ENTRA                           | 25          | 000003 | JRC3: | JSR     | 0,ENTRA                           |
| 26          | 000002 | JRC2: | JSR     | 0,RNADR                           | 26          | 000003 | JRC3: | JSR     | 0,RNADR                           |
| 27          | 000002 | JRC2: | LOBF    |                                   | 27          | 000003 | JRC3: | LOBF    |                                   |
| 28          | 000002 | JRC2: | LDA     | 1,JMLC2                           | 28          | 000003 | JRC3: | LDA     | 1,JMLC2                           |
| 29          | 000002 | JRC2: | SUB#    | 1,0,SNR                           | 29          | 000003 | JRC3: | SUB#    | 1,0,SNR                           |
| 30          | 000002 | JRC2: | JMP     | JRC30                             | 30          | 000003 | JRC3: | JMP     | JRC30                             |
| 31          | 000002 | JRC2: | STA     | 0,JJMLC2                          | 31          | 000003 | JRC3: | STA     | 0,JJMLC2                          |
| 32          | 000002 | JRC2: | MOV     | 0,2                               | 32          | 000003 | JRC3: | MOV     | 0,2                               |
| 33          | 000002 | JRC2: | LDA     | 1,JMLC0                           | 33          | 000003 | JRC3: | LDA     | 1,JMLC0                           |
| 34          | 000002 | JRC2: | STA     | 1,0,2                             | 34          | 000003 | JRC3: | STA     | 1,0,2                             |
| 35          | 000002 | JRC2: | ELEF    | 1,JRC2-+2,1                       | 35          | 000003 | JRC3: | ELEF    | 1,JRC2-+2,1                       |
| 36          | 000002 | JRC2: | STA     |                                   | 36          | 000003 | JRC3: | STA     |                                   |
| 37          | 000002 | JRC2: | LDA     | 1,JMLC1                           | 37          | 000003 | JRC3: | LDA     | 1,JMLC1                           |
| 38          | 000002 | JRC2: | MOVZR   | 0,2,52C 1((RNDM ADDR 1)/2)+BIT 15 | 38          | 000003 | JRC3: | MOVZR   | 0,2,52C 1((RNDM ADDR 1)/2)+BIT 15 |
| 39          | 000002 | JRC2: | ADI     | 1,2                               | 39          | 000003 | JRC3: | ADI     | 1,2                               |
| 40          | 000002 | JRC2: | HLV     | 0                                 | 40          | 000003 | JRC3: | HLV     | 0                                 |
| 41          | 000002 | JRC2: | NDLOC   | 0                                 | 41          | 000003 | JRC3: | NDLOC   | 0                                 |
| 42          | 000002 | JRC2: | IDRI    | 100000,0                          | 42          | 000003 | JRC3: | IDRI    | 100000,0                          |
| 43          | 000002 | JRC2: | STA     | 0,+2                              | 43          | 000003 | JRC3: | STA     | 0,+2                              |
| 44          | 000002 | JRC2: | EJSR    | 00,2                              | 44          | 000003 | JRC3: | EJSR    | 00,2                              |
| 45          | 000002 | JRC2: | ERROR   |                                   | 45          | 000003 | JRC3: | ERROR   |                                   |
| 46          | 000002 | JRC2: | LOOP    |                                   | 46          | 000003 | JRC3: | LOOP    |                                   |
| 47          | 000002 | JRC2: | JSR     | 0,ENTLO                           | 47          | 000003 | JRC3: | JSR     | 0,ENTLO                           |
| 48          | 000002 | JRC2: |         |                                   | 48          | 000003 | JRC3: |         |                                   |

(ADDR OF TEST+1) IS  
 SAVED IN LOC. HELP  
 FILL SCRATCH AREA WITH  
 JSR #ENTER  
 INITIALIZE TEST.  
 IC(AC0)=RANDOM #  
 JGET ADDRESS IN THE RANGE  
 IC(HIBF) AND C(HIBFU)  
 IC(JMLC0)=RNDM ADDR 1  
 IC(AC0)=RANDOM #  
 JGET ADDRESS IN THE RANGE  
 IC(L0WB) AND C(L0BFU)  
 1  
 1  
 IC(RNDM ADDR 1)=RNDM ADDR 2  
 IACBAC=RNDM ADDR 2  
 ISTORE JUMP #JMLC1 IN  
 RNDM ADDR 2  
 ISTORE ADDR OF (EJSR+3) IN  
 JMLC1 FOR GOOD RETURN  
 IACB=RNDM ADDR 1  
 ADDR 1)/2)+BIT 15  
 IIS STORED IN ACS AND  
 IACB=(RNDM ADDR 1)/2  
 IADD THE INDIRECT BIT TO  
 IACB AND STORE IN EJSR  
 I  
 ERROR  
 LOOP  
 JSR #ENTLO  
 ITERATE TEST ROUTINE

10133 ECL31

```

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

```

```

? CHECKING MULTI-LEVEL INDIRECT "EJSR"
?
JRS00: JMSK=0      JMSK=0
EJSR,LOSFAD,JRO0,LOSFAD,*,HISFAD,JR00B,JR001
TS1LOC          ?
?1             ?
3*HELP         ?
PJFILL        ?
JSR #ENTER    ?

INITIALIZE TEST.

JRS00:
RANDR      LOBF,LOBFU
JSR #ENTR  ?
#RNADR    ?
#LOBF    ?
#LOBFU   ?
STA 0,JMLC2
LOSFAD   LOBF,LOBFU
RANDR    LOBF,LOBFU
RAND      ?
JSR #ENTR ?
#RNADR   ?
#LOBF    ?
#LOBFU   ?
LDA 1,JMLC2
SUB# 10,SNR
JMP JR000
STA 0,JMLC3
IDRI 100000,0

STA 0,JMLC2
HISFAD  HISF, HISFU
RANDR   HISF, HISFU
RAND    ?
JSR #ENTR ?
#RNADR  ?
#HISF   ?
#HISFU  ?
LDA 1,JMLC3
SUB# 0,1,SNR
JMP JR001
LDA 1,JMLC2
SUB# 10,SNR
JMP JR001
STA 0,JJMLC3
MOV #0,2
LDA 1,JMLC0
STA 1,0,2
ELEF 1,JR00B-*,2,1

?FOR PROPER RETURN
?ACB# RNDM ADDR 1
?ADD INDIRECT BIT TO ACB

```

```

W134 ECL31
01      100000
02      W7104 040402
03      W7105 100070 JR00:
04      100000
05      ?
06      ?
07      W7113 006306

```

```

STA      0,*,*2
EJSR    #0,0
ERROR
LOOP
JSR #ENTLO
ITERATE TFST ROUTINE

```

```

RAND STORE IN EJSR

```





0138 ECL31  
 01 07255 107070 JRD21  
 02  
 03  
 04  
 05 07253 000300  
 EJSR 00,2  
 ERROR  
 LOOP  
 JSR #ENTLO  
 ITERATE TEST ROUTINE

|    |        |        |            |   |
|----|--------|--------|------------|---|
| 01 | 080002 | JRD21  | JMSUC=2    | EJSR,LOBFAD,JRD2,LOBFAD,2,HIFAD,JRD26,JRD21 |
| 02 |        |        | TS1LOC     |   |
| 03 |        |        | JSR #ENTRA | I(ADDR OF TEST+1) IS                        |
| 04 | 07200  | 064401 | STA        | ISAVED IN LOC. HELP                         |
| 05 | 07201  | 054215 | JSR        | IFILL SCRATCH AREA WITH                     |
| 06 | 07202  | 066314 | JSR        | JSR #ENTER                                  |
| 07 |        |        | SETUP      | INITIALIZE TEST.                            |
| 08 | 07203  | 066305 | JSR #ENTIN |   |
| 09 | 07204  | 050144 | LOBFAD     |   |
| 10 |        |        | RNDADR     | LOBF,LOBFU                                  |
| 11 |        |        | RAND       |   |
| 12 |        |        | JSR #ENTRA | IC(AC0)=RANDOM #                            |
| 13 | 07205  | 066310 | JSR #ENTRA | ISET ADDRESS IN THE RANGE                   |
| 14 | 07206  | 066311 | LOBF       | IC(LOBF) AND C(LOBFU)                       |
| 15 | 07207  | 066315 | LOBFU      |   |
| 16 | 07210  | 066312 | STA        | 0,JMLC2                                     |
| 17 | 07211  | 048061 | LOBFAD     | LOBF,LOBFU                                  |
| 18 |        |        | RNDADR     | LOBF,LOBFU                                  |
| 19 |        |        | RAND       |   |
| 20 |        |        | JSR #ENTRA | IC(AC0)=RANDOM #                            |
| 21 | 07212  | 066310 | JSR #ENTRA | ISET ADDRESS IN THE RANGE                   |
| 22 | 07213  | 066311 | LOBF       | IC(LOBF) AND C(LOBFU)                       |
| 23 | 07214  | 066315 | LOBFU      |   |
| 24 | 07215  | 066312 | LDA        | 0,JMLC2                                     |
| 25 | 07216  | 054201 | SUB#       | 1,0,SNR                                     |
| 26 | 07217  | 152415 | JMP        | JRD21                                       |
| 27 | 07218  | 066312 | STA        | 0,JMLC3                                     |
| 28 | 07220  | 066312 | TORI       | 100000,0                                    |
| 29 | 07221  | 048062 | STA        | 0,JMLC2                                     |
| 30 | 07222  | 103778 | HIFAD      | HIFB,HIFBU                                  |
| 31 | 07223  | 106000 | RNDADR     | HIFB,HIFBU                                  |
| 32 | 07224  | 048061 | RAND       |   |
| 33 |        |        | JSR #ENTRA | IC(AC0)=RANDOM #                            |
| 34 |        |        | HIFB       | ISET ADDRESS IN THE RANGE                   |
| 35 |        |        | HIFBU      | IC(HIFB) AND C(HIFBU)                       |
| 36 | 07225  | 066310 | LOA        | 1,JMLC3                                     |
| 37 | 07226  | 066311 | SUB#       | 0,1,SNR                                     |
| 38 | 07227  | 066313 | JMP        | JRD21                                       |
| 39 | 07230  | 066310 | LDA        | 1,JMLC2                                     |
| 40 | 07231  | 054202 | SUB#       | 1,0,SNR                                     |
| 41 | 07232  | 106435 | JMP        | JRD21                                       |
| 42 | 07233  | 066312 | STA        | 0,JMLC3                                     |
| 43 | 07234  | 054201 | NOV        | 0,2   |
| 44 | 07235  | 152415 | LDA        | 1,JMLC3                                     |
| 45 | 07236  | 066312 | STA        | 1,0,2                                       |
| 46 | 07237  | 048062 | ELF        | 1,JRD2,0,2,1                                |
| 47 | 07240  | 111800 | STA        | 1,JMLC1                                     |
| 48 | 07241  | 054207 | LDA        | 0,JMLC2                                     |
| 49 | 07242  | 048060 | NOVZR      | 0,2,0,2C I(RNDM                             |
| 50 | 07243  | 106478 | ADI        | ADDR 1)/2+ BIT 15                           |
| 51 | 066814 | 066814 | HLV        | 0   |
| 52 | 07245  | 044200 | NOLOC      | 0   |
| 53 | 07246  | 020201 | TORI       | 100000,0                                    |
| 54 | 07247  | 111822 | STA        | 0,0,2                                       |
| 55 | 07250  | 110016 |            |   |
| 56 | 07251  | 143370 |            |   |
| 57 | 07251  | 066800 |            |   |
| 58 | 07252  | 103770 |            |   |
| 59 | 07253  | 106000 |            |   |
| 60 | 07254  | 048062 |            |   |

```

18139 ECL31
01 00000 JRD31 JMSKCS
02 00000 EJSR,LOBFAD,JRDS,LOBFAD,J,LOBFAD,JRD36,JRD31
03 TSTLOC
04 07264 06481 JSR +1 I(ADDR OF TEST+1) IS
05 07265 054016 STA S,HELP I(SET ADDRESS IN LOC. HELP
06 07266 066314 JBR @JILL I(FILL SCRATCH AREA WITH
07 JBR PENTER I(JBR PENTER)
08 SETUP 100. INITIALIZE TEST.
09 JBR @ENTIN
10 100.
11 HIFAD
12 RNDADR HIBF,HISFU
13 RAND
14 07271 066310 JBR @ENTRA I(C(ACB)*RANDOM #
15 07272 066311 JBR @RNADR I(SET ADDRESS IN THE RANGE
16 07273 066313 HIBF I(C(HIGB) AND C(HISFU))
17 07274 066316 HIBFU
18 07275 06261 STA 0,JMLC2 I(C(JMLC2)*RNDM ADDR 1
19 LOBFAD
20 RNDADR LOHBF,LOBFU
21 RAND
22 07276 066310 JBR @ENTRA I(C(ACB)*RANDOM #
23 07277 066311 JBR @RNADR I(SET ADDRESS IN THE RANGE
24 07300 066315 LOHBF I(C(LOHBF) AND C(LOBFU))
25 07301 066310 LOBFU
26 07302 06261 LDA I,JMLC2
27 07303 12415 SUB# I,0,SNR
28 07304 066772 JMP JRD36 I(JMLC3)*RNDM ADDR 2
29 07305 066772 JMP JRD36 I(C(RNDM ADDR 1)*
30 07306 103779 IOR1 100000,0
31 100000
32 07310 06261 STA 0,JMLC2 I(*RNDM ADDR 2)
33 LOBFAD
34 RNDADR LOHBF,LOBFU
35 RAND
36 07311 066310 JBR @ENTRA I(C(ACB)*RANDOM #
37 07312 066311 JBR @RNADR I(SET ADDRESS IN THE RANGE
38 07313 066315 LOHBF I(C(LOHBF) AND C(LOBFU))
39 07314 066312 LOBFU
40 07315 06262 LDA I,JMLC3
41 07316 10415 SUB# I,1,SNR
42 07317 066772 JMP JRD31 I(JMLC2
43 07300 06481 LDA I,JMLC2 I(JMLC2
44 07301 12415 SUB# I,0,SNR
45 07302 066772 JMP JRD31 I(JBR+3) IS STORED IN JMLC1
46 07303 06262 STA 0,JMLC3 I(RNDM ADDR 0)*RNDM ADDR 3
47 07304 11000 MOV 0,2 I(ACB)*RNDM ADDR 3
48 07305 06481 LDA I,JMLC0 I(JMP @JMLC1) IS STORED IN
49 07306 06680 STA 1,0,2 I(RNDM ADDR 3 AND ADDR
50 07307 10470 ELEF I(EJBR+3) IS STORED IN JMLC1
51 066014
52 07331 04400 STA I,JMLC1 I(OR PROPER RETURN
53 07330 06001 LDA 0,JMLC2 I(ACB)*RNDM ADDR 1
54 07333 10222 MOVZR 0,J,SEC I(RNDM ADDR 1)* BIT 15
55 07334 14010 ADI 1,3 I(5 IN ACS AND ACS=
56 07335 143370 HLY 0 I(RNDM ADDR 1)*2
57 066000 I,NOLC 0
58 07336 103770 IOR1 100000,0 I(ADD INDIRECT BIT TO ACS
59 100000 I(AND STORE IN EJBR
60 07340 040400

```

```

0140 ECL31
01 07341 107470 JRD31 EJSR 00,3
02 100000 ERROR
03 LOOP
04 JBR @ENTLO I(ITERATE TEST ROUTINE
05 07347 066306
06

```

10141 ECL31

```

01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
PASSIN
DORR
PASS
9,9,9,9
P,ASSVL
P,ASSIN
READS
MOVLM
JMP
ELDA
ANDI
NOV
PCKI
JSR
PASHB
NOVZ
LDA
JSR
LDA
NOV
JMP
DST
JMP
TORST
LDA
LDA
STA
JMP
CALL
REL
EXTR:
10194 014094
02 0301 008437
04 0302 010283
05 0303 011061
06 0304 003077
07 0305 008285
08 0306 040884
09 0307 000477
10 0308 011112
11 0309 000493
12 0310 000493
13 0311 022470
14 000034
15 03364 143770
16 004080
17 03366 011084
18 03367 000486
19
20 03370 000822
21 03371 001455
22 03372 000820
23 03373 024883
24 03374 000825
25
26 03375 034010=PCKI:
27 03376 021400
28 03377 011085
29 07400 000410
30 07401 015483
31 07402 000406
32 07403 000877
33 07404 021403
34 07405 030484
35 07406 041776
36 07407 001406
37 07410 000821
38 07411 001172
39
40
41
42
43
44

```

\*\*\* END OF TEST ROUTINES \*\*\*

10142 ECL31

```

01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
*****EGGS & DIRT DATA BLOCKS*****
EGGS:
05 07412 000000 AUTO: 0
06 07413 000000 DEV: 0
07 07414 000000 CATSH: 0
08 07415 000000 PCNT: 0
09 07416 000000 RTRN: 0
10 07417 000000 SAREG: 0
11
12 07420 047503
13 054520
14 044522
15 044107
16 024124
17 024503
18 043604
19 026183
20 034461
21 032067
22 033464
23 025065
24 033667
25 07435 040101 ALL RIGHTS RESERVED/
26 020114
27 044522
28 044107
29 051524
30 051940
31 051505
32 051100
33 040820
34 000104
35
36 07447 141705 DIRTS .TXTE !ECLIPSE313!
37 144714
38 051508
39 031705
40 031661
41 000000
42 07405 000000
43 07406 000000
44 07407 175772
45 07408 000000
46 07409 000000
47 07410 000000
48 07411 000000
49 07412 000000
50
51 07465 007465 PREEND:
52
53
54
55
56
57 07466 000144 MURUFF: .BLK 100.
58
59
60

```

RELOCATABLE UPPER BUFFER USED IN ABOVE TEST

0143 ECL31

.END DTOSB

01 000000  
 02 00013-000240  
 03 177770  
 04 000212  
 05 104400  
 06 022000  
 07 000377  
 08 000000  
 09 000011  
 10 030031  
 11 000144  
 12 000200  
 13 000400  
 14 001777  
 15 000300  
 16 000270  
 17 000010

0144 ECL31

AC0 000220  
 AC1 000227  
 AC2 000230  
 AC3 000231

\*\*00000 TOTAL ERRORS, 00000 PASS 1 ERRORS

|        |        |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|--------|
| 26/16  | 30/08  | 30/30  | 31/21  | 31/34  | 32/05  | 32/28  |
| 32/39  | 31/22  | 31/35  | 32/15  | 32/30  | 32/46  | 35/04  |
| 35/18  | 31/23  | 31/36  | 32/16  | 32/32  | 32/47  | 35/05  |
| 35/19  | 32/34  | 32/39  | 42/35  | 42/37  | 42/39  | 42/41  |
| 43/21  | 43/23  | 43/25  | 43/27  | 44/21  | 44/23  | 44/25  |
| 44/27  | 45/21  | 45/23  | 45/25  | 45/27  | 46/17  | 47/17  |
| 48/17  | 49/17  | 50/22  | 51/22  | 52/22  | 53/22  | 54/21  |
| 53/21  | 54/20  | 54/20  | 54/20  | 54/20  | 54/20  | 54/21  |
| 55/21  | 55/20  | 54/20  | 55/27  | 55/29  | 55/31  | 55/33  |
| 56/22  | 56/24  | 56/26  | 56/28  | 57/22  | 57/24  | 57/26  |
| 57/28  | 58/22  | 58/24  | 58/26  | 58/28  | 58/30  | 58/32  |
| 59/22  | 59/24  | 59/26  | 59/28  | 59/30  | 59/32  | 59/34  |
| 60/22  | 60/24  | 60/26  | 60/28  | 60/30  | 60/32  | 60/34  |
| 61/22  | 61/24  | 61/26  | 61/28  | 61/30  | 61/32  | 61/34  |
| 62/22  | 62/24  | 62/26  | 62/28  | 62/30  | 62/32  | 62/34  |
| 63/22  | 63/24  | 63/26  | 63/28  | 63/30  | 63/32  | 63/34  |
| 64/22  | 64/24  | 64/26  | 64/28  | 64/30  | 64/32  | 64/34  |
| 65/22  | 65/24  | 65/26  | 65/28  | 65/30  | 65/32  | 65/34  |
| 66/22  | 66/24  | 66/26  | 66/28  | 66/30  | 66/32  | 66/34  |
| 67/22  | 67/24  | 67/26  | 67/28  | 67/30  | 67/32  | 67/34  |
| 68/22  | 68/24  | 68/26  | 68/28  | 68/30  | 68/32  | 68/34  |
| 69/22  | 69/24  | 69/26  | 69/28  | 69/30  | 69/32  | 69/34  |
| 70/22  | 70/24  | 70/26  | 70/28  | 70/30  | 70/32  | 70/34  |
| 71/22  | 71/24  | 71/26  | 71/28  | 71/30  | 71/32  | 71/34  |
| 72/22  | 72/24  | 72/26  | 72/28  | 72/30  | 72/32  | 72/34  |
| 73/22  | 73/24  | 73/26  | 73/28  | 73/30  | 73/32  | 73/34  |
| 74/22  | 74/24  | 74/26  | 74/28  | 74/30  | 74/32  | 74/34  |
| 75/22  | 75/24  | 75/26  | 75/28  | 75/30  | 75/32  | 75/34  |
| 76/22  | 76/24  | 76/26  | 76/28  | 76/30  | 76/32  | 76/34  |
| 77/22  | 77/24  | 77/26  | 77/28  | 77/30  | 77/32  | 77/34  |
| 78/22  | 78/24  | 78/26  | 78/28  | 78/30  | 78/32  | 78/34  |
| 79/22  | 79/24  | 79/26  | 79/28  | 79/30  | 79/32  | 79/34  |
| 80/22  | 80/24  | 80/26  | 80/28  | 80/30  | 80/32  | 80/34  |
| 81/22  | 81/24  | 81/26  | 81/28  | 81/30  | 81/32  | 81/34  |
| 82/22  | 82/24  | 82/26  | 82/28  | 82/30  | 82/32  | 82/34  |
| 83/22  | 83/24  | 83/26  | 83/28  | 83/30  | 83/32  | 83/34  |
| 84/22  | 84/24  | 84/26  | 84/28  | 84/30  | 84/32  | 84/34  |
| 85/22  | 85/24  | 85/26  | 85/28  | 85/30  | 85/32  | 85/34  |
| 86/22  | 86/24  | 86/26  | 86/28  | 86/30  | 86/32  | 86/34  |
| 87/22  | 87/24  | 87/26  | 87/28  | 87/30  | 87/32  | 87/34  |
| 88/22  | 88/24  | 88/26  | 88/28  | 88/30  | 88/32  | 88/34  |
| 89/22  | 89/24  | 89/26  | 89/28  | 89/30  | 89/32  | 89/34  |
| 90/22  | 90/24  | 90/26  | 90/28  | 90/30  | 90/32  | 90/34  |
| 91/22  | 91/24  | 91/26  | 91/28  | 91/30  | 91/32  | 91/34  |
| 92/22  | 92/24  | 92/26  | 92/28  | 92/30  | 92/32  | 92/34  |
| 93/22  | 93/24  | 93/26  | 93/28  | 93/30  | 93/32  | 93/34  |
| 94/22  | 94/24  | 94/26  | 94/28  | 94/30  | 94/32  | 94/34  |
| 95/22  | 95/24  | 95/26  | 95/28  | 95/30  | 95/32  | 95/34  |
| 96/22  | 96/24  | 96/26  | 96/28  | 96/30  | 96/32  | 96/34  |
| 97/22  | 97/24  | 97/26  | 97/28  | 97/30  | 97/32  | 97/34  |
| 98/22  | 98/24  | 98/26  | 98/28  | 98/30  | 98/32  | 98/34  |
| 99/22  | 99/24  | 99/26  | 99/28  | 99/30  | 99/32  | 99/34  |
| 100/22 | 100/24 | 100/26 | 100/28 | 100/30 | 100/32 | 100/34 |
| 101/22 | 101/24 | 101/26 | 101/28 | 101/30 | 101/32 | 101/34 |
| 102/22 | 102/24 | 102/26 | 102/28 | 102/30 | 102/32 | 102/34 |
| 103/22 | 103/24 | 103/26 | 103/28 | 103/30 | 103/32 | 103/34 |
| 104/22 | 104/24 | 104/26 | 104/28 | 104/30 | 104/32 | 104/34 |
| 105/22 | 105/24 | 105/26 | 105/28 | 105/30 | 105/32 | 105/34 |
| 106/22 | 106/24 | 106/26 | 106/28 | 106/30 | 106/32 | 106/34 |
| 107/22 | 107/24 | 107/26 | 107/28 | 107/30 | 107/32 | 107/34 |
| 108/22 | 108/24 | 108/26 | 108/28 | 108/30 | 108/32 | 108/34 |
| 109/22 | 109/24 | 109/26 | 109/28 | 109/30 | 109/32 | 109/34 |
| 110/22 | 110/24 | 110/26 | 110/28 | 110/30 | 110/32 | 110/34 |
| 111/22 | 111/24 | 111/26 | 111/28 | 111/30 | 111/32 | 111/34 |
| 112/22 | 112/24 | 112/26 | 112/28 | 112/30 | 112/32 | 112/34 |
| 113/22 | 113/24 | 113/26 | 113/28 | 113/30 | 113/32 | 113/34 |
| 114/22 | 114/24 | 114/26 | 114/28 | 114/30 | 114/32 | 114/34 |
| 115/22 | 115/24 | 115/26 | 115/28 | 115/30 | 115/32 | 115/34 |
| 116/22 | 116/24 | 116/26 | 116/28 | 116/30 | 116/32 | 116/34 |
| 117/22 | 117/24 | 117/26 | 117/28 | 117/30 | 117/32 | 117/34 |
| 118/22 | 118/24 | 118/26 | 118/28 | 118/30 | 118/32 | 118/34 |
| 119/22 | 119/24 | 119/26 | 119/28 | 119/30 | 119/32 | 119/34 |
| 120/22 | 120/24 | 120/26 | 120/28 | 120/30 | 120/32 | 120/34 |
| 121/22 | 121/24 | 121/26 | 121/28 | 121/30 | 121/32 | 121/34 |
| 122/22 | 122/24 | 122/26 | 122/28 | 122/30 | 122/32 | 122/34 |
| 123/22 | 123/24 | 123/26 | 123/28 | 123/30 | 123/32 | 123/34 |
| 124/22 | 124/24 | 124/26 | 124/28 | 124/30 | 124/32 | 124/34 |
| 125/22 | 125/24 | 125/26 | 125/28 | 125/30 | 125/32 | 125/34 |
| 126/22 | 126/24 | 126/26 | 126/28 | 126/30 | 126/32 | 126/34 |
| 127/22 | 127/24 | 127/26 | 127/28 | 127/30 | 127/32 | 127/34 |
| 128/22 | 128/24 | 128/26 | 128/28 | 128/30 | 128/32 | 128/34 |
| 129/22 | 129/24 | 129/26 | 129/28 | 129/30 | 129/32 | 129/34 |
| 130/22 | 130/24 | 130/26 | 130/28 | 130/30 | 130/32 | 130/34 |

AUTO 007412  
 BABL 000275  
 BABY 000265  
 BEC 000707  
 BEGIN 001570  
 BONAD 000200  
 BUFP 000204  
 CAL 001144  
 CAL0 000242  
 CAL1 000243  
 CAL2 000244  
 CALL 001614  
 CATM 007414  
 CHAR 001372  
 CHAR1 001421  
 CHAR2 001430  
 CHAR3 001435  
 CHAR4 001447  
 CHARE 000235  
 CHORZ 000236  
 CHR8V 001454  
 CRY 000232  
 D100 000255  
 DEV 007415  
 DIRT 007447  
 DOROR 007410  
 DT080 000200  
 EGG8 007412  
 EMALT 000377  
 EJNI1 001021  
 EJM2 001067

MC

MC





0140 ECL31

JPD1 006282 117/58 117/54 117/60  
 JPD10 006137 117/19 117/28  
 JPD11 006152 117/33 117/42 117/45  
 JPD2 006886 119/58 119/54 126/01  
 JPD80 006223 119/19 119/28  
 JPD21 006236 119/33 119/42 119/45  
 JPD3 006352 121/50 121/54 122/01  
 JPD38 006387 121/19 121/28  
 JPD31 006322 121/33 121/42 121/45  
 JRA1 006431 124/10 124/12  
 JRB0 006451 125/21 125/23  
 JRB1 006513 126/20 126/23  
 JRB2 006577 127/20 127/27  
 JRB5 006577 128/20 128/27  
 JRC0 006642 129/38 129/42  
 JRC80 006620 129/24 129/33  
 JRC1 006710 130/53 130/57  
 JRC2 006756 131/33 131/37  
 JRC3 006731 131/19 131/28  
 JRC38 006724 132/33 132/37  
 JRC38 006777 132/19 132/28  
 JRD0 007195 133/55 133/59 134/03  
 JRD88 007845 133/24 133/33  
 JRD1 007888 133/58 133/47 133/50  
 JRD1 007171 135/58 135/54 135/60  
 JRD10 007126 135/19 135/28  
 JRD11 007141 135/33 135/42 135/45  
 JRD2 007595 137/50 137/54 136/01  
 JRD58 007512 137/19 137/28  
 JRD1 007525 137/33 137/42 137/45  
 JRD5 007531 139/58 139/54 140/01  
 JRD38 007276 139/19 139/28  
 JRD31 007311 139/53 139/42 139/45  
 JRA0 006351 123/86  
 JRA1 006421 124/01  
 JRB0 006441 125/02  
 JRB1 006478 126/01  
 JRB2 006522 127/01  
 JRB3 006554 128/01  
 JRB5 006886 129/06  
 JRC1 006851 130/01  
 JRC2 006717 131/01  
 JRC3 006765 132/01  
 JRD0 007033 133/06  
 JRD1 007114 135/01  
 JRD2 007200 137/01  
 JRD3 007264 139/01  
 JCLW 001244 29/10 29/22 32/44 48/59  
 LAB1 003146 70/85 70/18  
 LAC1 003246 73/89 73/18 73/21  
 LAC10 003325 74/10 74/21 74/24  
 LAC19 003325 74/06 74/10 74/20  
 LAC2 003337 75/05 75/10 75/17  
 LAC3 003373 76/05 76/10 76/16  
 LAD0 003441 77/26 77/35  
 LAD1 003532 78/32 78/44  
 LAD18 003587 78/22 78/31

0150 ECL31

LAD2 003557 78/23 79/33 79/41  
 LAD3 003636 80/22 80/32 80/38  
 LDA0 002707 85/06  
 LAA1 002752 86/01  
 LAA2 003015 87/01  
 LAA3 003060 88/01  
 LDAB0 003123 89/02  
 LDAB1 003144 90/02  
 LDAB2 003174 91/02  
 LDAB3 003220 92/02  
 LDACP 003244 93/06  
 LDAC1 003275 94/02  
 LDAC2 003336 95/02  
 LDAC3 003371 96/02  
 LDAD0 003424 97/06  
 LDAD1 003472 98/02  
 LDAD2 003542 99/01  
 LDAD3 003613 00/01  
 LDALC 000850 26/37 74/09 74/15 78/12 78/29 78/41  
 LD8V0 000261 26/38 77/16 77/33 78/38 79/38 79/41  
 LD8V1 000252 29/13 79/38 80/12 80/37 80/48 80/53 80/58 80/59 80/62 79/30  
 LD8V1 000252 25/39 77/25 77/48 78/01 78/38 79/22 79/30  
 LD8V1 000252 79/39 80/21 80/29 80/34  
 LEFAB 001071 42/16  
 LEFA1 001634 43/01  
 LEFA2 001677 44/01  
 LEFAS 001742 45/01  
 LEFB0 002085 46/01  
 LEFB1 002026 47/01  
 LEFB2 002047 48/01  
 LEFB3 002070 49/01  
 LEFC0 002111 50/01  
 LEFC1 002137 51/01  
 LEFC2 002166 52/01  
 LEFC3 002213 53/01  
 LEFD0 002241 54/02  
 LEFE0 002265 55/02  
 LEFF0 002311 56/06  
 LEFF1 002341 57/02  
 LEFF2 002375 58/02  
 LEFF3 002433 59/02  
 LEFG0 002471 60/06  
 LEFG1 002522 61/02  
 LEFG2 002552 62/02  
 LEFH0 002615 63/02  
 LEFH8 002650 64/06  
 LEFLC 000246 26/36 60/21 60/24 61/24 62/16 62/36  
 LEFL0 000245 63/18 63/36 60/19 60/22 62/16 62/22  
 LFC0 002113 63/16 63/22  
 LFC1 002141 60/05 60/13  
 LFC2 002167 51/08 51/13  
 LFC3 002215 52/05 52/13  
 LFD0 002245 53/05 53/13  
 LFE0 002271 54/08 54/13  
 LFF0 002313 55/06 55/13  
 LFF1 002343 56/09 56/18 57/05 57/20

0151 ECL31

0152 ECL31

|     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| LF2 | 00377 | 58/05 | 58/19 | 58/23 | 58/14 | 59/07 | 59/14 | 59/08 | 51/13 | 60/05 | 61/13 | 62/08 | 63/13 | 64/18 | 65/23 | 66/28 | 67/33 | 68/38 | 69/43 | 70/48 | 71/53 | 72/58 | 73/63 | 74/68 | 75/73 | 76/78 | 77/83 | 78/88 | 79/93 | 80/98 | 81/03 | 82/08 | 83/13 | 84/18 | 85/23 | 86/28 | 87/33 | 88/38 | 89/43 | 90/48 | 91/53 | 92/58 | 93/63 | 94/68 | 95/73 | 96/78 | 97/83 | 98/88 | 99/93 | 100/98 | 101/03 | 102/08 | 103/13 | 104/18 | 105/23 | 106/28 | 107/33 | 108/38 | 109/43 | 110/48 | 111/53 | 112/58 | 113/63 | 114/68 | 115/73 | 116/78 | 117/83 | 118/88 | 119/93 | 120/98 | 121/03 | 122/08 | 123/13 | 124/18 | 125/23 | 126/28 | 127/33 | 128/38 | 129/43 | 130/48 | 131/53 | 132/58 | 133/63 | 134/68 | 135/73 | 136/78 | 137/83 | 138/88 | 139/93 | 140/98 | 141/03 | 142/08 | 143/13 | 144/18 | 145/23 | 146/28 | 147/33 | 148/38 | 149/43 | 150/48 | 151/53 | 152/58 | 153/63 | 154/68 | 155/73 | 156/78 | 157/83 | 158/88 | 159/93 | 160/98 | 161/03 | 162/08 | 163/13 | 164/18 | 165/23 | 166/28 | 167/33 | 168/38 | 169/43 | 170/48 | 171/53 | 172/58 | 173/63 | 174/68 | 175/73 | 176/78 | 177/83 | 178/88 | 179/93 | 180/98 | 181/03 | 182/08 | 183/13 | 184/18 | 185/23 | 186/28 | 187/33 | 188/38 | 189/43 | 190/48 | 191/53 | 192/58 | 193/63 | 194/68 | 195/73 | 196/78 | 197/83 | 198/88 | 199/93 | 200/98 | 201/03 | 202/08 | 203/13 | 204/18 | 205/23 | 206/28 | 207/33 | 208/38 | 209/43 | 210/48 | 211/53 | 212/58 | 213/63 | 214/68 | 215/73 | 216/78 | 217/83 | 218/88 | 219/93 | 220/98 | 221/03 | 222/08 | 223/13 | 224/18 | 225/23 | 226/28 | 227/33 | 228/38 | 229/43 | 230/48 | 231/53 | 232/58 | 233/63 | 234/68 | 235/73 | 236/78 | 237/83 | 238/88 | 239/93 | 240/98 | 241/03 | 242/08 | 243/13 | 244/18 | 245/23 | 246/28 | 247/33 | 248/38 | 249/43 | 250/48 | 251/53 | 252/58 | 253/63 | 254/68 | 255/73 | 256/78 | 257/83 | 258/88 | 259/93 | 260/98 | 261/03 | 262/08 | 263/13 | 264/18 | 265/23 | 266/28 | 267/33 | 268/38 | 269/43 | 270/48 | 271/53 | 272/58 | 273/63 | 274/68 | 275/73 | 276/78 | 277/83 | 278/88 | 279/93 | 280/98 | 281/03 | 282/08 | 283/13 | 284/18 | 285/23 | 286/28 | 287/33 | 288/38 | 289/43 | 290/48 | 291/53 | 292/58 | 293/63 | 294/68 | 295/73 | 296/78 | 297/83 | 298/88 | 299/93 | 300/98 | 301/03 | 302/08 | 303/13 | 304/18 | 305/23 | 306/28 | 307/33 | 308/38 | 309/43 | 310/48 | 311/53 | 312/58 | 313/63 | 314/68 | 315/73 | 316/78 | 317/83 | 318/88 | 319/93 | 320/98 | 321/03 | 322/08 | 323/13 | 324/18 | 325/23 | 326/28 | 327/33 | 328/38 | 329/43 | 330/48 | 331/53 | 332/58 | 333/63 | 334/68 | 335/73 | 336/78 | 337/83 | 338/88 | 339/93 | 340/98 | 341/03 | 342/08 | 343/13 | 344/18 | 345/23 | 346/28 | 347/33 | 348/38 | 349/43 | 350/48 | 351/53 | 352/58 | 353/63 | 354/68 | 355/73 | 356/78 | 357/83 | 358/88 | 359/93 | 360/98 | 361/03 | 362/08 | 363/13 | 364/18 | 365/23 | 366/28 | 367/33 | 368/38 | 369/43 | 370/48 | 371/53 | 372/58 | 373/63 | 374/68 | 375/73 | 376/78 | 377/83 | 378/88 | 379/93 | 380/98 | 381/03 | 382/08 | 383/13 | 384/18 | 385/23 | 386/28 | 387/33 | 388/38 | 389/43 | 390/48 | 391/53 | 392/58 | 393/63 | 394/68 | 395/73 | 396/78 | 397/83 | 398/88 | 399/93 | 400/98 | 401/03 | 402/08 | 403/13 | 404/18 | 405/23 | 406/28 | 407/33 | 408/38 | 409/43 | 410/48 | 411/53 | 412/58 | 413/63 | 414/68 | 415/73 | 416/78 | 417/83 | 418/88 | 419/93 | 420/98 | 421/03 | 422/08 | 423/13 | 424/18 | 425/23 | 426/28 | 427/33 | 428/38 | 429/43 | 430/48 | 431/53 | 432/58 | 433/63 | 434/68 | 435/73 | 436/78 | 437/83 | 438/88 | 439/93 | 440/98 | 441/03 | 442/08 | 443/13 | 444/18 | 445/23 | 446/28 | 447/33 | 448/38 | 449/43 | 450/48 | 451/53 | 452/58 | 453/63 | 454/68 | 455/73 | 456/78 | 457/83 | 458/88 | 459/93 | 460/98 | 461/03 | 462/08 | 463/13 | 464/18 | 465/23 | 466/28 | 467/33 | 468/38 | 469/43 | 470/48 | 471/53 | 472/58 | 473/63 | 474/68 | 475/73 | 476/78 | 477/83 | 478/88 | 479/93 | 480/98 | 481/03 | 482/08 | 483/13 | 484/18 | 485/23 | 486/28 | 487/33 | 488/38 | 489/43 | 490/48 | 491/53 | 492/58 | 493/63 | 494/68 | 495/73 | 496/78 | 497/83 | 498/88 | 499/93 | 500/98 | 501/03 | 502/08 | 503/13 | 504/18 | 505/23 | 506/28 | 507/33 | 508/38 | 509/43 | 510/48 | 511/53 | 512/58 | 513/63 | 514/68 | 515/73 | 516/78 | 517/83 | 518/88 | 519/93 | 520/98 | 521/03 | 522/08 | 523/13 | 524/18 | 525/23 | 526/28 | 527/33 | 528/38 | 529/43 | 530/48 | 531/53 | 532/58 | 533/63 | 534/68 | 535/73 | 536/78 | 537/83 | 538/88 | 539/93 | 540/98 | 541/03 | 542/08 | 543/13 | 544/18 | 545/23 | 546/28 | 547/33 | 548/38 | 549/43 | 550/48 | 551/53 | 552/58 | 553/63 | 554/68 | 555/73 | 556/78 | 557/83 | 558/88 | 559/93 | 560/98 | 561/03 | 562/08 | 563/13 | 564/18 | 565/23 | 566/28 | 567/33 | 568/38 | 569/43 | 570/48 | 571/53 | 572/58 | 573/63 | 574/68 | 575/73 | 576/78 | 577/83 | 578/88 | 579/93 | 580/98 | 581/03 | 582/08 | 583/13 | 584/18 | 585/23 | 586/28 | 587/33 | 588/38 | 589/43 | 590/48 | 591/53 | 592/58 | 593/63 | 594/68 | 595/73 | 596/78 | 597/83 | 598/88 | 599/93 | 600/98 | 601/03 | 602/08 | 603/13 | 604/18 | 605/23 | 606/28 | 607/33 | 608/38 | 609/43 | 610/48 | 611/53 | 612/58 | 613/63 | 614/68 | 615/73 | 616/78 | 617/83 | 618/88 | 619/93 | 620/98 | 621/03 | 622/08 | 623/13 | 624/18 | 625/23 | 626/28 | 627/33 | 628/38 | 629/43 | 630/48 | 631/53 | 632/58 | 633/63 | 634/68 | 635/73 | 636/78 | 637/83 | 638/88 | 639/93 | 640/98 | 641/03 | 642/08 | 643/13 | 644/18 | 645/23 | 646/28 | 647/33 | 648/38 | 649/43 | 650/48 | 651/53 | 652/58 | 653/63 | 654/68 | 655/73 | 656/78 | 657/83 | 658/88 | 659/93 | 660/98 | 661/03 | 662/08 | 663/13 | 664/18 | 665/23 | 666/28 | 667/33 | 668/38 | 669/43 | 670/48 | 671/53 | 672/58 | 673/63 | 674/68 | 675/73 | 676/78 | 677/83 | 678/88 | 679/93 | 680/98 | 681/03 | 682/08 | 683/13 | 684/18 | 685/23 | 686/28 | 687/33 | 688/38 | 689/43 | 690/48 | 691/53 | 692/58 | 693/63 | 694/68 | 695/73 | 696/78 | 697/83 | 698/88 | 699/93 | 700/98 | 701/03 | 702/08 | 703/13 | 704/18 | 705/23 | 706/28 | 707/33 | 708/38 | 709/43 | 710/48 | 711/53 | 712/58 | 713/63 | 714/68 | 715/73 | 716/78 | 717/83 | 718/88 | 719/93 | 720/98 | 721/03 | 722/08 | 723/13 | 724/18 | 725/23 | 726/28 | 727/33 | 728/38 | 729/43 | 730/48 | 731/53 | 732/58 | 733/63 | 734/68 | 735/73 | 736/78 | 737/83 | 738/88 | 739/93 | 740/98 | 741/03 | 742/08 | 743/13 | 744/18 | 745/23 | 746/28 | 747/33 | 748/38 | 749/43 | 750/48 | 751/53 | 752/58 | 753/63 | 754/68 | 755/73 | 756/78 | 757/83 | 758/88 | 759/93 | 760/98 | 761/03 | 762/08 | 763/13 | 764/18 | 765/23 | 766/28 | 767/33 | 768/38 | 769/43 | 770/48 | 771/53 | 772/58 | 773/63 | 774/68 | 775/73 | 776/78 | 777/83 | 778/88 | 779/93 | 780/98 | 781/03 | 782/08 | 783/13 | 784/18 | 785/23 | 786/28 | 787/33 | 788/38 | 789/43 | 790/48 | 791/53 | 792/58 | 793/63 | 794/68 | 795/73 | 796/78 | 797/83 | 798/88 | 799/93 | 800/98 | 801/03 | 802/08 | 803/13 | 804/18 | 805/23 | 806/28 | 807/33 | 808/38 | 809/43 | 810/48 | 811/53 | 812/58 | 813/63 | 814/68 | 815/73 | 816/78 | 817/83 | 818/88 | 819/93 | 820/98 | 821/03 | 822/08 | 823/13 | 824/18 | 825/23 | 826/28 | 827/33 | 828/38 | 829/43 | 830/48 | 831/53 | 832/58 | 833/63 | 834/68 | 835/73 | 836/78 | 837/83 | 838/88 | 839/93 | 840/98 | 841/03 | 842/08 | 843/13 | 844/18 | 845/23 | 846/28 | 847/33 | 848/38 | 849/43 | 850/48 | 851/53 | 852/58 | 853/63 | 854/68 | 855/73 | 856/78 | 857/83 | 858/88 | 859/93 | 860/98 | 861/03 | 862/08 | 863/13 | 864/18 | 865/23 | 866/28 | 867/33 | 868/38 | 869/43 | 870/48 | 871/53 | 872/58 | 873/63 | 874/68 | 875/73 | 876/78 | 877/83 | 878/88 | 879/93 | 880/98 | 881/03 | 882/08 | 883/13 | 884/18 | 885/23 | 886/28 | 887/33 | 888/38 | 889/43 | 890/48 | 891/53 | 892/58 | 893/63 | 894/68 | 895/73 | 896/78 | 897/83 | 898/88 | 899/93 | 900/98 | 901/03 | 902/08 | 903/13 | 904/18 | 905/23 | 906/28 | 907/33 | 908/38 | 909/43 | 910/48 | 911/53 | 912/58 | 913/63 | 914/68 | 915/73 | 916/78 | 917/83 | 918/88 | 919/93 | 920/98 | 921/03 | 922/08 | 923/13 | 924/18 | 925/23 | 926/28 | 927/33 | 928/38 | 929/43 | 930/48 | 931/53 | 932/58 | 933/63 | 934/68 | 935/73 | 936/78 | 937/83 | 938/88 | 939/93 | 940/98 | 941/03 | 942/08 | 943/13 | 944/18 | 945/23 | 946/28 | 947/33 | 948/38 | 949/43 | 950/48 | 951/53 | 952/58 | 953/63 | 954/68 | 955/73 | 956/78 | 957/83 | 958/88 | 959/93 | 960/98 | 961/03 | 962/08 | 963/13 | 964/18 | 965/23 | 966/28 | 967/33 | 968/38 | 969/43 | 970/48 | 971/53 | 972/58 | 973/63 | 974/68 | 975/73 | 976/78 | 977/83 | 978/88 | 979/93 | 980/98 | 981/03 | 982/08 | 983/13 | 984/18 | 985/23 | 986/28 | 987/33 | 988/38 | 989/43 | 990/48 | 991/53 | 992/58 | 993/63 | 994/68 | 995/73 | 996/78 | 997/83 | 998/88 | 999/93 | 1000/98 | 1001/03 | 1002/08 | 1003/13 | 1004/18 | 1005/23 | 1006/28 | 1007/33 | 1008/38 | 1009/43 | 1010/48 | 1011/53 | 1012/58 | 1013/63 | 1014/68 | 1015/73 | 1016/78 | 1017/83 | 1018/88 | 1019/93 | 1020/98 | 1021/03 | 1022/08 | 1023/13 | 1024/18 | 1025/23 | 1026/28 | 1027/33 | 1028/38 | 1029/43 | 1030/48 | 1031/53 | 1032/58 | 1033/63 | 1034/68 | 1035/73 | 1036/78 | 1037/83 | 1038/88 | 1039/93 | 1040/98 | 1041/03 | 1042/08 | 1043/13 | 1044/18 | 1045/23 | 1046/28 | 1047/33 | 1048/38 | 1049/43 | 1050/48 | 1051/53 | 1052/58 | 1053/63 | 1054/68 | 1055/73 | 1056/78 | 1057/83 | 1058/88 | 1059/93 | 1060/98 | 1061/03 | 1062/08 | 1063/13 | 1064/18 | 1065/23 | 1066/28 | 1067/33 | 1068/38 | 1069/43 | 1070/48 | 1071/53 | 1072/58 | 1073/63 | 1074/68 | 1075/73 | 1076/78 | 1077/83 | 1078/88 | 1079/93 | 1080/98 | 1081/03 | 1082/08 | 1083/13 | 1084/18 | 1085/23 | 1086/28 | 1087/33 | 1088/38 | 1089/43 | 1090/48 | 1091/53 | 1092/58 | 1093/63 | 1094/68 | 1095/73 | 1096/78 | 1097/83 | 1098/88 | 1099/93 | 1100/98 | 1101/03 | 1102/08 | 1103/13 | 1104/18 | 1105/23 | 1106/28 | 1107/33 | 1108/38 | 1109/43 | 1110/48 | 1111/53 | 1112/58 | 1113/63 | 1114/68 | 1115/73 | 1116/78 | 1117/83 | 1118/88 | 1119/93 | 1120/98 | 1121/03 | 1122/08 | 1123/13 | 1124/18 | 1125/23 | 1126/28 | 1127/33 | 1128/38 | 1129/43 | 1130/48 | 1131/53 | 1132/58 | 1133/63 | 1134/68 | 1135/73 | 1136/78 | 1137/83 | 1138/88 | 1139/93 | 1140/98 | 1141/03 | 1142/08 | 1143/13 | 1144/18 | 1145/23 | 1146/28 | 1147/33 | 1148/38 | 1149/43 | 1150/48 | 1151/53 | 1152/58 | 1153/63 | 1154/68 |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|







LISTING

096-000225-01

PROGRAM

EXERCISER FOR ECLIPSE  
PART 9

TAPE

095-000246-01

ABSTRACT

'ECLIPSE32' IS AN EXERCISER PROGRAM USED TO TEST THE RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF THE ECLIPSE COMPUTER. 'ECLIPSE32' EXERCISES THE DOUBLE WORD INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES OF ITS RELIABLE OPERATION.



0001 ECL32 MACRO REV 03.00 16:13:05 08/06/76  
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

10002 ECL32  
02  
03  
04  
05  
06  
07

TITLE ECL32  
:ECLIPSE32  
:ECLIPSE32 - CONTINUATION OF ECLIPSE31  
:PART 9 OF EXERCISER FOR ECLIPSE  
:

\*\*\*\*\*  
: NAME: ECLIPSE32.SR PART NUMBER: 0900-000645  
:  
: DESCRIPTION: ECLIPSE EXERCISER, PART 9  
:  
: REVISION HISTORY:  
: REV. DATE  
: 00 12/20/74  
: 01 08/06/76  
:  
: COPYRIGHT (C) DATA GENERAL CORPORATION, 1974, 1976  
: ALL RIGHTS RESERVED.  
:\*\*\*\*\*

10003 ECL32

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

```

EXERCISER FOR ECLIPSE: PART 9

PROGRAM NAME  
-----  
ECLIPSE32  
GENERAL DESCRIPTION  
-----  
'ECLIPSE32' IS AN EXERCISER PROGRAM USED TO TEST THE  
RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF  
THE ECLIPSE COMPUTER. 'ECLIPSE32' EXERCISES THE DOUBLE  
WORD INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES  
OF ITS RELIABLE OPERATION.

THE INSTRUCTIONS EXERCISED ARE AS FOLLOWS:

EISZ/EDSZ AND DSRA

LOCATIONS 200 TO 205 IN PAGE 0 ARE FIXED FOR ECLIPSE32  
PROGRAM.

LOCATION 203 KEEPS TRACK OF NUMBER OF PASSES RUN  
THROUGH ECLIPSE32 PROGRAM.

LOCATION 201 KEEPS TRACK OF THE TEST RUNNING AT  
PRESENT AND IS USEFUL FOR DEBUG WHEN LOOPING  
OCCURS IN THE PROGRAM.

LOCATION 202 CONTAINS THE STARTING ADDRESS OF  
ECLIPSE32 PROGRAM.

LOCATION 200 IS USED BY DTOS.

LOCATION 204 KEEPS TRACK OF INTERNAL PASS COUNT  
WHICH IS FIXED BY LOCATION 205.

FIRST PASS THROUGH ECLIPSE32 TEST WILL RUN SUPERFAST.  
NEXT PASSES WILL RUN SLOWER AS EACH TEST IS RUN SEVERAL  
TIMES TO RUN ALL RANDOM NUMBER COMBINATIONS.

MACHINE REQUIREMENTS  
-----  
ECLIPSE PROCESSOR  
4K READ-WRITE MEMORY  
CONSOLE EQUIPMENT

10004 ECL32

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

```

SWITCH SETTINGS  
-----  
THIS PROGRAM USES DATA SWITCHES AS FOLLOWS

SW"0" - USE CONTENTS OF "SWRES" IF 0  
USE DATA SWITCHES IF 1  
SW"1" - LOOP ON FAILING TEST IF 0  
PROCEED TO NEXT TEST IF 1  
SW"2" - OUTPUT TO ITY IF 0  
INHIBIT PRINTING TO ITY IF 1  
SW"3" - DO NOT PRINT % ERRORS IF 0  
PRINT FAILURE RATE IF 1  
SW"4" - PRINT PASS COUNT IF 0  
INHIBIT PRINTING PASS COUNT IF 1  
SW"5" - INHIBIT OUTPUT TO LINE PRINTER IF 0  
OUTPUT TO LINE PRINTER IF 1

EDTOS PRESUMES THAT ALL SWITCHES ARE 0. RETURN IS  
MADE TO THE OPER SYS UPON AN ERROR AFTER FIRST PRINTING  
THE FAILURE REPORT. IN STAND ALONE MODE, THE OPERATOR  
IS ASKED TO PRESET THESE SWITCHES (THE MACHINE HALTS  
FOR HIM TO DO SO) AND PRESS CONTINUE. THIS PRESET VALUE  
CAN BE CHANGED AT ANY TIME BY SETTING DATA SWITCH "0"  
TO A ONE. THE PROGRAM WILL THEN FOLLOW ACTUAL SETTINGS.

STAND ALONE STARTING ADDRESS = 200  
IF 'CAT' OR 'KITTEN' WAS LOADED FROM DTOS AND RESTR  
WAS NEEDED, THEN USE AS FOLLOWS:  
STARTING ADDR = 170 (FOR START WITH NO 'CAT')  
STARTING ADDR = 171 (FOR START WITH 'CAT')

MONITOR LOCATION 203 TO CHECK THE CURRENT PASS COUNT

MONITOR LOCATION X6000 TO MAKE SURE THAT 'CAT' OR  
'KITTEN' IS RUNNING. IN CASES WHERE PROGRAM IS  
STARTED WITH 'CAT' OR 'KITTEN' LOCATION X6000 WILL SHOW  
A PATTERN CHANGING FROM ZEROS TO ALL ONES  
TO AN INC/SWAP PATTERN.

(X% THE NUMBER OF THE HIGHEST MEMORY MODULE IN THE  
SYSTEM AND MAY BE A VALUE 0 - 7)

10005 ECL32

```
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
```

SWITCH SETTINGS  
-----  
THIS PROGRAM USES THE DATA SWITCHES AS FOLLOWS:  
SW\*0\* - USE CONTENTS OF "SWREG" IF 0  
SW\*1\* - LOOP ON FAILING TEST IF 0  
SW\*2\* - PROCEED TO NEXT TEST IF 1  
SW\*3\* - DO NOT PRINT IF 1  
SW\*4\* - DO NOT PRINT \* ERRORS IF 0  
SW\*5\* - PRINT FAILURE RATE IF 1

EDTOS PRESUMES THAT ALL SWITCHES ARE 0. RETURN IS  
MADE TO THE OPER SYS UPON AN ERROR AFTER FIRST PRINTING  
THE FAILURE REPORT. IN STAND ALONE MODE, THE OPERATOR  
IS ASKED TO PRESET THESE SWITCHES (THE MACHINE MALTS  
FOR HIM TO DO SO) AND PRESS CONTINUE. THIS PRESET VALUE  
CAN BE CHANGED AT ANY TIME BY SETTING DATA SWITCH \*0\*  
TO A "1". THE PROGRAM WILL THEN FOLLOW ACTUAL SETTINGS.

14.2 STARTING ADDRESS = 200  
14.3 MONITOR LOCATION 203 TO CHECK THE CURRENT PASS COUNT.

10006 ECL32

```
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
```

OPERATING PROCEDURE/OPERATOR INPUT  
-----  
LOAD THE PROGRAM VIA THE BINARY LOADER OR INSERT A  
PRELOADED MEMORY MODULE.  
SET SWITCHES TO 200.  
PRESS START.  
PROGRAM WILL HALT AFTER PRINTING THE MESSAGE  
'SET DATA SWITCHES AND PRESS CONTINUE'.  
SET REQUIRED SWITCH SETTINGS AND PRESS CONTINUE.  
THE PROGRAM WILL RUN UNTIL MANUALLY STOPPED. IN CASE  
OF MALFUNCTIONING, THE PROGRAM WILL PRINT ERROR  
MESSAGE AND TAKE APPROPRIATE ACTION AS PER THE SW  
SETTINGS.

PROGRAM OUTPUT/ERROR DESCRIPTION  
-----  
FOR ANY ERRORS DETECTED, THE PROGRAM WILL PRINT ERROR  
REPORT OR \* FAILURES DEPENDING UPON THE SW SETTINGS.  
ERROR REPORT CONSISTS OF ALL ACCUMULATORS, CARRY, FAILING  
RELOCATED PROGRAM COUNTER OF THE TEST THAT IS FAILING  
AND PC IN THE LISTING AT THE TIME OF FAILURE.  
THE PROGRAM WILL LOOP IN THE TEST THAT IS FAILING IF  
SW\*1\* IS 0.  
THE PRINTING OF ERROR REPORT CAN BE ABORTED BY SETTING  
SW\*2\* TO 1.  
IF LOOPING OCCURS IN THE PROGRAM, STOP THE COMPUTER  
AND CHECK LOCATION 201 TO FIND OUT THE TEST THAT WAS  
RUNNING BEFORE THE LOOPING OCCURRED.

10007 ECL32

10008 ECL32

01  
02  
03

063077 EMALT=HALT

PROGRAM DESCRIPTION/THEORY OF OPERATION  
-----

EACH TEST IS COMPLETE IN ITSELF, SO THE PROGRAM CAN BE STARTED FROM ANY TEST WITHOUT CAUSING ANY INITIALIZATION ERRORS.  
WHEN 'ECLIPSE32' IS STARTED AT LOCATION 200 OR BY DTOS, IT WILL SIZE UP THE MEMORY AND WILL PRINT THE TOP OF THE MEMORY.  
AFTER SETTING UP THE SWITCHES AND PRESSING CONTINUE, THE EXERCISER WILL RUN THE FIRST PASS VERY FAST. IN THE FIRST PASS EACH TEST IS RUN ONLY ONCE. ALL OTHER PASSES WILL TAKE MORE TIME AS EACH TEST IS RUN ACCORDING TO THE TEST COUNT SPECIFIED IN EACH TEST. AFTER THE 1ST PASS, ECLIPSE32 IS RELOCATED IN THE AVAILABLE MEMORY FOR ALL NEXT PASSES AND THE AREA BELOW AND ABOVE THE RELOCATED PROGRAM IS USED FOR SCRATCH BUFFER AREA. REFER TO THE LISTING TO FIND OUT THE INFORMATION ABOUT EACH TEST.

RESTRICTIONS/MISC  
-----

CERTAIN INSTRUCTIONS LIKE BLW, XCT, BAW, ETC., DO ALLOW INTERRUPTS TO OCCUR DURING THEIR EXECUTION. THIS FEATURE OF THOSE INSTRUCTIONS IS NOT CHECKED IN THIS TEST.

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





10011 ECL32

```

01 .MACRO EISZ3
02 TSTLOC
03 SETUP
04 RAND
05 RAND
06 STA
07 LOBFAD
08 STA
09 LDA
10 STA
11 HIBFAD
12 MOV
13 LDA
14 STA
15 .NOLOC
16 .IFE
17 .ELEM
18 SUB
19 .ENDC
20 .IFE
21 MOVZ
22 ADI
23 HLV
24 .NOLOC
25 ADDR
26 STA
27 .IF
28 MOV
29 LDA
30 LDA
31 SUB#
32 ERROR
33 LOOP
34
35
36
37

```

10012 FCL32

```

01 .MACRO EISZ4
02 TSTLOC
03 SETUP
04 RAND
05 RAND
06 STA
07 LOBFAD
08 STA
09 LDA
10 STA
11 HIBFAD
12 MOV
13 LDA
14 STA
15 .NOLOC
16 .IFE
17 .ELEM
18 SUB
19 .ENDC
20 .IFE
21 MOVZ
22 ADI
23 HLV
24 .NOLOC
25 ADDR
26 STA
27 .IF
28 MOVZ
29 ADI
30 HLV
31 .ENDC
32 .NOLOC
33 ADDR
34 STA
35 .IF
36 MOV
37 LDA
38 LDA
39 SUB#
40 ERROR
41 LOOP
42
43
44

```

```

01 .MACRO EISZ4
02 TSTLOC
03 SETUP
04 RAND
05 RAND
06 STA
07 LOBFAD
08 STA
09 LDA
10 STA
11 HIBFAD
12 MOV
13 LDA
14 STA
15 .NOLOC
16 .IFE
17 .ELEM
18 SUB
19 .ENDC
20 .IFE
21 MOVZ
22 ADI
23 HLV
24 .NOLOC
25 ADDR
26 STA
27 .IF
28 MOVZ
29 ADI
30 HLV
31 .ENDC
32 .NOLOC
33 ADDR
34 STA
35 .IF
36 MOV
37 LDA
38 LDA
39 SUB#
40 ERROR
41 LOOP
42
43
44

```

10013 ECL32

```

01 .MACRO EDSF1
02 TSTLOC
03 FILL
04 SETUP
05 LDA 0,1
06 LDA 1,2
07 LDA 0,1
08 STA 1,MXTBL
09 RNDADR
10 SBI
11 STA 0,TBLsiz
12 STA 0,TBLsiz
13 LDA 2,1
14 STA 0,0,2
15 LDA 1,TBLsiz
16 LDA 0,1
17 ADD 0,1
18 SBI
19 STA 1,1,2
20 MVLV 0
21 CLM 1,2
22 JMP *-8.
23 RAND
24 LDA 1,1
25 CLM 0,1
26 JMP *-3
27 ADI 2,2
28
29

```

10014 ECL32

```

01 .MACRO EDSF2
02 STA
03 STA 0,DSLCO
04 STA 2,DSLCO
05 LDA 1,2,2
06 ADD 0,2
07 SUB 1,2
08 STA 2,DSVV0
09 ELEM 1,1+3,-1,1
10 STA 1,0,2
11
12 STA 0,DSVV1
13 LDA 1,DSVV0
14 SUB# 0,1,SNR
15 JMP *-7
16 LDA 2,DSLCO
17 XCH 2,0
18 STA 0,0,2
19 .NOLC 1
20 .IFE (DSSW1-1)
21 *-4
22 STA 0,DSVV2
23 LDA 1,DSVV0
24 SUB# 0,1,SNR
25 JMP *-7
26 LDA 2,DSVV1
27 SUB# 0,2,SNR
28 JMP *-10.
29 XCH 0,2
30 ADDR 0,0
31 STA 0,0,2
32 .ENDC
33 MOV 2,0
34 .IFE (DSSW0-1)
35 ELEM 1,1,-,1
36 SUB 1,0
37 .ENDC
38 .IFE
39 MOVZR
40 ADI 1,3
41 HLV 0
42 .ENDC
43 .NOLC 0
44 ADDR
45 STA 0,1,1
46 LDA 1,DSLCO
47 DSPA 1,0,3
48 ERROR
49 LDA 0,DSPER
50 STA 0,0,DSVV0
51 STA 0,0,DSVV1
52 .NOLC 1
53 STA 0,0,DSVV2
54
55 .ENDC
56 .NOLC 0
57 LOOP
58
59

```

\*

10015 ECL32

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15

.MACRO EDSP3  
DSSWI=0  
EDSP1 21,22,23  
EDSP2 24,25,26

x

.MACRO EDSP4  
DSSWI=1  
EDSP1 21,22,23  
EDSP2 24,25,26,27

x

10016 ECL32

01  
02  
03  
04  
05  
06

.MACRO FILL  
JSR @JFILL

x

FILL UPPER AND LOWER  
SCRATCH BUFFER AREA  
WITH (\*)

10017 ECL32

01  
02  
03  
04  
05  
06

.MACRO    TSTLOC  
JSR        +1  
STA        5,HELP  
          X

;(ADDR OF TEST+1) IS  
;SAVED IN LOC. HELP

10018 ECL32

01  
02  
03  
04  
05  
06  
07  
08

.MACRO    LOBFAD        LOWBF,LOBFU  
          RNDADR  
          X  
.MACRO    HIBFAD        HIBGF,HIBFU  
          RNDADR  
          X

10019 ECL32

```

01 .MACRO LOTBAD
02 LDA 0,L0MBF
03 ADI 2,0
04 STA 0,LDSTB
05 RNDADR LDSTB,L0BFF
06
07 *
08 .MACRO HITBAD
09 LDA 0,HIGBF
10 ADI 2,0
11 STA 0,HDSTB
12 RNDADR HDSTB,HIGBF
13
14 *

```

```

MAKE SURE THAT RNDM
ADDR IS NOT EQ. TO
:(TABLE-1) OR (TABLE-2)

MAKE SURE THAT RNDM
ADDR IS NOT EQ. TO
:(TABLE-1) OR (TABLE-2)

```

10020 ECL32

```

01 .MACRO RNDADR
02 RAND
03 JSR *1
04 *2
05
06
07 *

```

GET ADDRESS IN THE RANGE  
PC(-1) AND C(-2)

10021 ECL32

```

01 : STANDARD MACROS
02
03
04
05 .MACRO LOOP
06 JSR @ENTLO
07 X
08
09 .MACRO SETUP
10 JSR @ENTIN
11 *1
12 X
13
14 .MACRO RAND
15 JSR @ENTRA
16 X
17
18 .MACRO CALL
19 JSR @ICAL
20 *1
21 X
22 .MACRO ERROR
23 JMP *2
24 **
25 ** STA 3,AC3
26 ** JSR @ENTER
27 X
28
29
30 .MACRO JMPER
31 LDA 3,ITRER
32 MOV 3,3,SZR
33 JMP *1
34 X

```

10022 ECL32

```

01 : ***** DIAGNOSTIC PROGRAM PREAMBLE *****
02
03
04 .LOC 0
05 00000 006643 : POINTER TO DIRT BLOCK
06
07 .LOC 40
08 : STACK CONTROL LOCATIONS
09 00040 000000
10 00041 000000
11 00042 000000
12 00043 000044
13 00044 063077
14
15 .LOC 45
16 00045 006607 : POINTER TO EGGS BLOCK
17
18 .ZREL
19 .BLK 8. : 8 LOCATIONS RESERVED
20 : FOR DEBUG BREAKPOINTS
21
22 00010-006607 .EGGS: EGGS
23 00011-000000 ME*TOP: 0
24 00012-000000 ICAT: 0
25
26 : SPACE RESERVED FOR CAT/NO CAT RESTART ENTRIES
27
28 .LOC 170
29 00170 102441 OFF: SUBO 0,0,SKP
30 00171 102000 ON: ADC 0,0
31 00172 142470 ESTA 0,CATSN
32
33 00174 002175 JMP @+1
34 00175 000700 SWMESS
35
36 : SPACE RESERVED FOR SPECIAL RESTART ENTRIES
37
38 .LOC 176
39 00176 000002 .BLK 2
40
41 : LOCATIONS 200 - 215 RESERVED FOR ECLIPSE TESTS
42
43
44 .LOC 200
45 00200 002202 DTOSB: JMP @BGNADR
46 00201 000000 ITRET: 0
47 00202 000644 BGNADR: NSTRT
48 00203 000000 PASS: 0
49 00204 000002 PASSIN: 2.
50 00205 000012 PASSVL: 10.
51
52 00206 000000 ITR: 0
53 00207 000000 ITRCI: 0
54 00210 000000 ITRER: 0
55 00211 000000 ITRER: 0
56 00212 000000 ERRET: 0
57 00213 000000 LOPRET: 0
58 00214 000000 RELOC: 0
59 00215 000000 LISTING: 0
60 00216 000000 HELP: 0

```

```

: PAGE ZERO STARTING LOC.
: LAST TEST ENTERED
: POINTER TO TEST SETUP ROUTINES
: PASS COUNT
: INTERNAL PASS COUNT
: INITIAL VALUE, INT. PASS COUNT
: ITERATION VALUE FOR THIS TEST
: ITERATION COUNTER
: ERROR SWITCH
: ERROR COUNTER
: ERROR RETURN
: LAST PLACE LOOP EXECUTED
: RELOCATION VALUE
: LISTING ADDR OF FAILING TEST
: START OF LAST TEST ENTERED

```

```

0023 ECL32
01
02 : *****LOCAL ZREL*****
03 : END OF PROGRAM
04 : RELOCATION RANGE OF PROGRAM
05
06 : POINTERS TO NON-RELOCATING SUBROUTINES
07
08 00221 001184 ICAL: CAL
09 00222 001350 IMESS: MESS
10 00223 001156 ISIZE: SIZE
11 00224 001313 IPOCT: POC
12 00225 001303 IPDEC: PDEC
13
14 : TEMPORARYS FOR TESTS AND SUBROUTINES
15
16 00226 000000 AC0: 0
17 00227 000000 AC1: 0
18 00230 000000 AC2: 0
19 00231 000000 AC3: 0
20 00232 000000 CRY: 0
21
22 00233 000001 ONE: 1
23 00234 000000 TSTOP: 0
24
25 00235 000000 CHARET: 0
26 00236 000000 CHORZ: 0
27 00237 000000 PDERET: 0
28 00240 000000 MESRET: 0
29 00241 123456 RAN: 123456
30 00242 000000 CAL0: 0
31 00243 000000 CAL1: 0
32 00244 000000 CAL2: 0
33
34 00245 023420 ITIHOU: 10000.
35 00246 000000 ISLC0: 0
36 00247 000000 ISLC1: 0
37 00250 000000 ISLC2: 0
38 00251 000000 DSLC0: 0
39 00252 000000 DSLC1: 0
40 00253 000000 DSSV0: 0
41 00254 000000 DSSV1: 0
42 00255 000000 DSSV2: 0
43 00256 000000 LDSTB: 0
44 00257 000000 HDSTB: 0
45 00260 000003 DS: 3
46 00261 000000 MX7BL: 0
47 00262 000000 TBL3IZ: 0
48 00263 000500 BUFP: L0BUFF
49

```

```

10024 ECL32
01
02 : ROUTINE TO RELOCATE TEST PROGRAM
03
04 00264 113710 BARMY: RAM ;BLOCK MOVE ON PROGRAM
05
06 00265 020214 LDA 0,RELOC
07 00266 024033- LDA 1,=8.
08 00267 030032- LDA 2,=RAMBLK
09 00270 034031- LDA 3,=RAMBLK+8.
10 00271 113710 RAM ;ADJUST POINTERS TO
11 CALL ;RELOCATED SUBROUTINES
12 JSR @ICAL ;EXIT TO TEST PROGRAM
13 ;CALL SUBROUTINE
14
15 BEGIN
16
17 00274 000743 ; INITIAL LOCATIONS OF RELOCATING SUBROUTINES
18 00275 000767 ;TEST INITIALIZER
19 00276 001043 ;TEST TERMINATOR
20 00277 001126 ;ERROR ROUTINE
21 00300 001235 ;RANDOM NUMBER GEN.
22 00301 000843 ;RANDOM ADDR. GEN.
23 00302 006662 ;LOBUFF+99.
24 00303 001256 ;JMFIL ;BOTTOM OF UPPER BUFFER
25 ;FILLS SCRATCH BUFFER
26
27 00304 000743 ; CORRECTED POINTERS TO RELOCATED ROUTINES
28 00305 000767 ;ENTINI: INIT ;CORRECTED VALUE=
29 00306 001043 ;ENTLO: LOP ;INITIAL VALUE +
30 00307 001126 ;ENTRA: RAND ;C(RELOC)
31 00310 001235 ;RNDOR: RNMAD
32 00311 000843 ;ROSFU: LOBUFF+99.
33 00312 006662 ;HIGBF: HIUBUF
34 00313 001256 ;JMFIL ;BOTTOM OF LOWER BUFFER
35 ;TOP OF UPPER BUFFER
36
37 00314 000000 L0WBF: 0
38 00315 000000 HIGFU: 0

```



10025 ECL32

```

01 000500 .LOC 500
02 ;
03 ; RELOCATABLE LOWER BUFFER USED IN THE FOLLOWING TESTS
04 ;
05 ;
06 ;
07 00500 000144 LOBUFF: .BLK 100.
08 ;
09 ;
10 ; *****SIZE SYSTEM & RESERVE MEMORY*****
11 00644 042677 NSTR: IORST
12 00645 046223 JSR
13 00646 150400 NEG
14 00647 150000 COM
15 00648 050011 STA
16 00650 050011-
17
18 00651 122470 ELDA 0,CATSW
19 00652 005737 MOV 0,0,SNR
20 00653 101005 NOCAT
21 00654 000414 JMP
22
23 00655 122470 ELDA 0,PRGENO
24 00656 006003 LDA 1,=1777
25 00657 024030- ADD 1,0
26 00660 123000 ADC# 0,2,SNC
27 00661 112033 JMP NOCAT
28 00662 000406
29
30 00663 132400 SUB 1,2
31 00664 024027- LDA 1,=400
32 00665 147000 ADD 2,1
33 00666 044012- STA 1,ICAT
34 00667 000406 JMP STLOC
35
36 00670 024026-NOCAT: LDA 1,=200
37 00671 132400 SUB 1,2
38 00672 126400 SUB 1,1
39 00673 146470 ESTA 1,CATSW
40 005715
41
42 00675 024217 STLOC: LDA 1,MINLOC
43 00676 132400 SUB 1,2
44 00677 050220 STA 2,MAXLOC

```

10026 ECL32

```

01
02 ; *****OUTPUT STRY MESSAGE & READ SWITCHES*****
03 ;PRINT SIZE OF MEMORY
04 00700 006222 SMES: JSR @IMESS
05 00701 001923 MESIZ
06 00702 024011- LDA 1,MEMTOP
07 00703 101040 MOVO 0,0
08 00704 006224 JSR @IPOCT
09 00705 006222 JSR @IMESS
10 00706 001537 KCRLF
11 00707 126400 SUB 1,1
12 00710 044203 STA 1,PASS
13
14 00711 126470 ELDA 1,AUTO
15 005675
16 00713 125004 MOV 1,1,SZR
17 00714 000413 JMP START
18
19 00715 006222 JSR @IMESS
20 00716 001541 SETSW
21 00717 006222 JSR @IMESS
22 00720 001537 KCRLF
23 00721 063077 HALT
24 00722 000401 JMP *+1
25 00723 060477 READS 0
26 00724 142470 ESTA 0,SMREG
27
28 00726 000401 JMP START
29
30
31 00727 176400 SUB 3,3
32 00730 054203 STA 3,RELOC
33 00731 054214 LDA 1,MINLOC
34 00732 024217 LDA 2,MAXLOC
35 00733 030220 ADD 1,2
36 00734 133000 LDA 1,2
37 00735 110110 SBI
38 00736 050234 STA 2,STTOP
39 00737 050315 STA 2,MIBFU
40 00740 024263 LDA 1,BUFF
41 00741 044314 STA 1,LOWBF
42 00742 000265 JMP BAMBY+1

```

START:

BEG:

```

SUB 3,3
STA 3,RELOC
LDA 1,MINLOC
LDA 2,MAXLOC
ADD 1,2
SBI
STA 2,STTOP
STA 2,MIBFU
LDA 1,BUFF
STA 1,LOWBF
JMP BAMBY+1

```

```

;CLEAR PASS COUNT
;SET RELOCATION TO 0
;STORE TOPMOST USABLE
;ADDRESS IN STTOP
;
;
;INITIALISE TOP OF UPPER BUFF
;INITIALISE BOTTOM OF LOWER
;BUFFER FOR INITIAL PASS
;AND START THE PROGRAM

```

```

1:0027 ECL32
01
02
03
04
05
06 00743 175400 INIT: INC 3,3
07 00744 054201 STA 3,ITRET
08 00745 040226 STA 0,ACO
09
10 00746 021777 LDA 0,-1,3
11 00747 040206 STA 0,ITR
12 00750 040207 STA 0,ITRCT
13
14 00751 020214 LDA 0,RELOC
15 00752 116400 SUB 0,3
16 00753 054215 STA 3,LISTING
17
18 00754 176400 SUB 3,3
19 00755 054210 STA 3,ITRER
20 00756 054211 STA 3,ITREC
21
22 00757 034203 LDA 3,PASS
23 00760 175004 MOV 3,3,SZR
24 00761 000404 JMP INIT1
25
26 00762 176520 SUBZL 3,3
27 00763 054206 STA 3,ITR
28 00764 054207 STA 3,ITRCT
29
30 00765 020226 INIT1: LDA 0,ACO
31 00766 002201 JMP @ITRET

: *****TEST UTILITY SUBROUTINES*****
: SUBROUTINE TO INITIALIZE A TEST LOOP
:
01
02
03
04
05
06 00767 054213 LOP: STA 3,LOPRET
07 00770 014207 DSZ ITRCT
08 00771 000440 JMP LOP3
09 00772 034210 LDA 3,ITRER
10 00773 034210 LDA 3,3,SNR
11 00774 002213 MOV @LOPRET
12 00775 034206 LDA 3,ITR
13 00776 054207 STA 3,ITRCT
14
15 00777 074477 LOP1: READS 3
16 01000 175112 MOVL# 3,3,SZC
17 01001 000403 JMP *3
18 01002 136470 ELDA 3,SWREG
19
20 01004 177100 ADDL 3,3
21 01005 177103 ADDL 3,3,SMC
22 01006 000421 JMP LOP2
23 01007 040526 STA 0,ACO
24 01010 044827 STA 1,AC1
25 01011 050230 STA 2,AC2
26 01012 006222 JSR @IMES
27 01013 001455 PERCENT 0,0
28 01014 102400 SUB 0,0
29 01015 024211 LDA 1,ITREC
30 01016 040211 STA 0,ITREC
31 01017 030025- LDA 2,-100-
32 01020 143710 MUL 2,ITR
33 01021 030506 LDA 2,ITR
34 01022 153710 DIV @IPREC
35 01023 006225 JSR 0,ACO
36 01024 020226 LDA 0,ACO
37 01025 024227 LDA 1,AC1
38 01026 030230 LDA 2,AC2
39 01027 176400 SUB 3,3
40 01030 054211 STA 3,ITREC
41
42 01031 034210 LOP3: LDA 3,ITRER
43 01032 175004 MOV 3,3,SZR
44 01033 074477 READS 3
45 01034 175112 MOVL# 3,3,SZC
46 01035 000403 JMP *3
47 01036 136470 ELDA 3,SWREG
48
49 01040 177113 ADDL# 3,3,SNR
50 01041 002201 JMP @ITRET
51 01042 002213 JMP @LOPRET

: SUBROUTINE TO TERMINATE A TEST LOOP
:
01
02
03
04
05
06 00767 054213 LOP: STA 3,LOPRET
07 00770 014207 DSZ ITRCT
08 00771 000440 JMP LOP3
09 00772 034210 LDA 3,ITRER
10 00773 034210 LDA 3,3,SNR
11 00774 002213 MOV @LOPRET
12 00775 034206 LDA 3,ITR
13 00776 054207 STA 3,ITRCT
14
15 00777 074477 LOP1: READS 3
16 01000 175112 MOVL# 3,3,SZC
17 01001 000403 JMP *3
18 01002 136470 ELDA 3,SWREG
19
20 01004 177100 ADDL 3,3
21 01005 177103 ADDL 3,3,SMC
22 01006 000421 JMP LOP2
23 01007 040526 STA 0,ACO
24 01010 044827 STA 1,AC1
25 01011 050230 STA 2,AC2
26 01012 006222 JSR @IMES
27 01013 001455 PERCENT 0,0
28 01014 102400 SUB 0,0
29 01015 024211 LDA 1,ITREC
30 01016 040211 STA 0,ITREC
31 01017 030025- LDA 2,-100-
32 01020 143710 MUL 2,ITR
33 01021 030506 LDA 2,ITR
34 01022 153710 DIV @IPREC
35 01023 006225 JSR 0,ACO
36 01024 020226 LDA 0,ACO
37 01025 024227 LDA 1,AC1
38 01026 030230 LDA 2,AC2
39 01027 176400 SUB 3,3
40 01030 054211 STA 3,ITREC
41
42 01031 034210 LOP3: LDA 3,ITRER
43 01032 175004 MOV 3,3,SZR
44 01033 074477 READS 3
45 01034 175112 MOVL# 3,3,SZC
46 01035 000403 JMP *3
47 01036 136470 ELDA 3,SWREG
48
49 01040 177113 ADDL# 3,3,SNR
50 01041 002201 JMP @ITRET
51 01042 002213 JMP @LOPRET

: IF NO ERROR, ITERATE
: OTHERWISE LOOK AT DATA
: *1 SWITCH FOR PROCEED,
: OR NOT.

```

10029 ECL32

```

01 ; ERROR SUBROUTINE: SETS ITRER FLAG, PRINTS MACHINE STATE.
02
03
04 STA 3,ERRRET
05 STA 0,AC0
06 SUBCL 0,0
07 STA 0,CRY
08 ISZ ITRER
09
10 LDA 0,ITRER
11 ADCL# 0,3,SNC
12 JMP ERR1
13
14 STA 3,ITRER
15 STA 1,AC1
16 STA 2,AC2
17 JSR @IMESS
18 ERMSG
19 LDA 1,PASS
20 INCZ 1,1
21 JSR @IPDEC
22 JSR @IMESS
23 JSR @IMESS
24 MOVZ 0,0
25 LDA 1,CRY
26 JSR @IPDEC
27 MOVZ 0,0
28 LDA 1,AC0
29 JSR @IPOCT
30 JSR @IPOCT
31 JSR @IPOCT
32 JSR @IPOCT
33 JSR @IPOCT
34 LDA 1,AC3
35 JSR @IPOCT
36
37 LDA 1,ERRRET
38 0,RELOC
39 SUBZ 0,1
40 JSR @IPOCT
41 LDA 1,ERRRET
42 JSR @IMESS
43 JSR @IMESS
44 KCRFLF
45
46 LDA 1,AC1
47 LDA 2,AC2
48 ELDA 3,AUTO
49
50 MOV 3,3,SNR
51 JMP ERR1
52
53 IORST
54 3,EGGS
55 LDA 3,4,3
56 JMP 0,3
57
58 LDA 0,AC0
59 LDA 3,AC3
60 JMP @ERRRET

```

0030 ECL32

```

01 ; RANDOM NUMBER GENERATOR SUBROUTINE.
02
03
04 LDA 0,RAN
05 LDA 1,ITRER
06 MOV 1,1,SZ
07 JMP 0,5
08
09 MOV 0,1
10 HXL 2,1
11 ADD 0,1
12 MOVZL 1,1
13 MOVZL 1,1
14 ADD 1,0
15 LDA 1,=33031
16 ADD 1,0
17 STA 0,RAN
18 JMP 0,5
19
20 ; CALL ROUTINE TO REACH RELOCATED SUBROUTINES.
21
22 STA 0,CALO
23 STA 1,CAL1
24 LDA 0,RELOC
25 LDA 1,0,3
26 ADD 1,0
27 STA 0,CAL2
28 INC 3,5
29 LDA 0,CALO
30 LDA 1,CAL1
31 JMP @CAL2
32
33 ; SIZING SUBROUTINE: RETURNS SIZE OF LOGICAL MEM IN AC2
34
35 LDA 2,MINLOC
36 INC 2,2,SZC
37 MOVZL # 0,6
38 JMP 0,0,2
39 LDA 2,0,2
40 STA 1,0,2
41 LDA 0,0,2
42 STA 1,2,SZR
43 SUB# 0,3
44 JMP 0,3
45
46 STA 2,MEMTOP
47 JMP SIZE+1

```

0030 ECL32

```

01 ; RANDOM NUMBER GENERATOR SUBROUTINE.
02
03
04 LDA 0,RAN
05 LDA 1,ITRER
06 MOV 1,1,SZ
07 JMP 0,5
08
09 MOV 0,1
10 HXL 2,1
11 ADD 0,1
12 MOVZL 1,1
13 MOVZL 1,1
14 ADD 1,0
15 LDA 1,=33031
16 ADD 1,0
17 STA 0,RAN
18 JMP 0,5
19
20 ; CALL ROUTINE TO REACH RELOCATED SUBROUTINES.
21
22 STA 0,CALO
23 STA 1,CAL1
24 LDA 0,RELOC
25 LDA 1,0,3
26 ADD 1,0
27 STA 0,CAL2
28 INC 3,5
29 LDA 0,CALO
30 LDA 1,CAL1
31 JMP @CAL2
32
33 ; SIZING SUBROUTINE: RETURNS SIZE OF LOGICAL MEM IN AC2
34
35 LDA 2,MINLOC
36 INC 2,2,SZC
37 MOVZL # 0,6
38 JMP 0,0,2
39 LDA 2,0,2
40 STA 1,0,2
41 LDA 0,0,2
42 STA 1,2,SZR
43 SUB# 0,3
44 JMP 0,3
45
46 STA 2,MEMTOP
47 JMP SIZE+1

```

0030 ECL32

```

01 ; RANDOM NUMBER GENERATOR SUBROUTINE.
02
03
04 LDA 0,RAN
05 LDA 1,ITRER
06 MOV 1,1,SZ
07 JMP 0,5
08
09 MOV 0,1
10 HXL 2,1
11 ADD 0,1
12 MOVZL 1,1
13 MOVZL 1,1
14 ADD 1,0
15 LDA 1,=33031
16 ADD 1,0
17 STA 0,RAN
18 JMP 0,5
19
20 ; CALL ROUTINE TO REACH RELOCATED SUBROUTINES.
21
22 STA 0,CALO
23 STA 1,CAL1
24 LDA 0,RELOC
25 LDA 1,0,3
26 ADD 1,0
27 STA 0,CAL2
28 INC 3,5
29 LDA 0,CALO
30 LDA 1,CAL1
31 JMP @CAL2
32
33 ; SIZING SUBROUTINE: RETURNS SIZE OF LOGICAL MEM IN AC2
34
35 LDA 2,MINLOC
36 INC 2,2,SZC
37 MOVZL # 0,6
38 JMP 0,0,2
39 LDA 2,0,2
40 STA 1,0,2
41 LDA 0,0,2
42 STA 1,2,SZR
43 SUB# 0,3
44 JMP 0,3
45
46 STA 2,MEMTOP
47 JMP SIZE+1

```

10031 ECL32

```

01 ;
02 ; RELOCATE SUBROUTINE:
03 ; ALLOCATES MEMORY FOR COPIES
04 ; OF TEST PROGRAM, WORKS IN CONJUNCTION
05 ; WITH RAM ROUTINE IN PAGE ZERO, WHICH
06 ; ACTUALLY COPIES THE TEST PROGRAM TO
07 ; THE NEW LOCATION.
08 01172 112470 REL: ETSZ CATSW
09 JMP 005416
10 01174 000402 JMR *+2
11 01175 005012- RAND @ICAT
12 JSR @ENTRA
13 01176 006307 LDA 1,MAXLOC
14 01177 024220 LDA 2,MINLOC
15 01200 030217 SUBZ# 2,1,SNC
16 01201 146433 JMP REL2
17 01202 000431 JMP REL2
18 01203 122422 JMP *+1
19 01204 000777 JMP *+1
20 01205 107000 ADD 0,1
21 01206 146433 SUBZ# 2,1,SNC
22 01207 000763 JMP REL
23 01210 020214 LDA 0,RELOC
24 01211 122432 SUBZ# 1,0,SZC
25 01212 000407 JMP REL1
26 01213 166400 SUB 0,1
27 01214 132433 SUBZ# 1,2,SNC
28 01215 000403 JMP *+3
29 01216 024220 LDA 1,MAXLOC
30 01217 102401 SUB 0,0,SKP
31 01220 107000 ADD 0,1
32 01221 042214 STA 1,RELOC
33 01222 136470 ELDA 3,PRGEN0
34 005436
35 01224 175400 INC 3,3
36 01225 054314 STA 3,LOWBF
37 01226 176400 SUB 3,3
38 01227 114710 XCH 0,3
39 01230 144710 XCH 2,1
40 01231 154710 XCH 2,3
41 01232 000264 JMP BAMB
42 CALL @ICAL
43 JSR @ICAL
44 01233 006221 REL2:
45 01234 001563 BEGIN

```

10032 ECL32

```

01 ;
02 ; RANDOM ADDRESS GENERATOR
03 ;
04 01235 044227 RNDMAD: STA 1,AC1
05 01236 050230 2,AC2
06 01237 125200 MOVR 1,1
07 01240 044232 STA 1,CRY
08 01241 033401 LDA 2,0,1,3
09 01242 027400 LDA 1,20,3
10 01243 132400 SUB 1,2
11 01244 124510 XCR 1,1
12 01245 164710 XCH 0,1
13 01246 153710 DIV
14 01247 027400 LDA 1,20,3
15 01250 123000 ADD 1,0
16 01251 054232 LDA 1,CRY
17 01252 125100 MOVL 1,1
18 01253 024227 LDA 1,AC1
19 01254 030230 LDA 2,AC2
20 01255 001402 JMP 2,3
21
22 ; ROUTINE TO FILL BUFFER WITH JSR @ENTER
23
24
25 01256 175400 JMFIL: INC 3,3
26 01257 054423 STA 3,JMEX
27 01260 025777 LDA 1,-1,3
28 01261 030314 LDA 2,LOWBF
29 01262 034311 LDA 3,LOWFU
30 01263 156400 SUB 2,3
31 01264 174000 COM 3,3
32 01265 045000 STA 1,0,2
33 01266 151400 INC 2,2
34 01267 175404 INC 3,3,SZR
35 01270 000775 JMP *+3
36 01271 030312 LDA 2,HIGBF
37 01272 034315 LDA 3,HIGFU
38 01273 156400 SUB 2,3
39 01274 174000 COM 3,3
40 01275 045000 STA 1,0,2
41 01276 151400 INC 2,2
42 01277 175404 INC 3,3,SZR
43 01300 000775 JMP *+3
44 01301 002401 JMP @JMEX
45 01302 000000 JMEX: 0

```

```

10033 ECL32
01 *****PRINT ROUTINES*****
02
03 STA 3, PDERET      !DECIMAL PRINT C(AC1).
04 JSR PDEC3          !RESET C(CARRY) FOR ZERO SUPPRESSION
05 INC 10000.         !SET C(CARRY) IF NOT
06
07 STA 3, PDEC3
08 JSR PDEC3
09 INC 1000.
10
11 STA 3, PDEC3
12 JSR PDEC3
13 INC 1000.
14
15 STA 3, PDEC3
16 JSR PDEC3
17 INC 1000.
18
19 STA 3, PDEC3
20 JSR PDEC3
21 INC 1000.
22
23 STA 3, PDERET
24 JSR PDEC3
25 INC 10000.
26
27 STA 3, PDEC3
28 JSR PDEC3
29 INC 10000.
30
31 STA 3, PDEC3
32 JSR PDEC3
33 INC 10000.
34
35 STA 3, PDEC3
36 JSR PDEC3
37 INC 10000.
38
39 STA 3, PDEC3
40 JSR PDEC3
41 INC 10000.
42
43 STA 3, PDEC3
44 JSR PDEC3
45 INC 10000.
46
47 STA 3, PDEC3
48 JSR PDEC3
49 INC 10000.
50
51 STA 3, PDEC3
52 JSR PDEC3
53 INC 10000.
54
55 STA 3, PDEC3
56 JSR PDEC3
57 INC 10000.
58
59 STA 3, PDEC3
60 JSR PDEC3
61 INC 10000.

```

```

10034 ECL32
01 *****PRINT ROUTINES*****
02
03 STA 3, PDERET      !DECIMAL PRINT C(AC1).
04 JSR PDEC3          !RESET C(CARRY) FOR ZERO SUPPRESSION
05 INC 1000.         !SET C(CARRY) IF NOT
06
07 STA 3, PDEC3
08 JSR PDEC3
09 INC 1000.
10
11 STA 3, PDEC3
12 JSR PDEC3
13 INC 1000.
14
15 STA 3, PDEC3
16 JSR PDEC3
17 INC 1000.
18
19 STA 3, PDEC3
20 JSR PDEC3
21 INC 1000.
22
23 STA 3, PDERET
24 JSR PDEC3
25 INC 10000.
26
27 STA 3, PDEC3
28 JSR PDEC3
29 INC 10000.
30
31 STA 3, PDEC3
32 JSR PDEC3
33 INC 10000.
34
35 STA 3, PDEC3
36 JSR PDEC3
37 INC 10000.
38
39 STA 3, PDEC3
40 JSR PDEC3
41 INC 10000.
42
43 STA 3, PDEC3
44 JSR PDEC3
45 INC 10000.
46
47 STA 3, PDEC3
48 JSR PDEC3
49 INC 10000.
50
51 STA 3, PDEC3
52 JSR PDEC3
53 INC 10000.
54
55 STA 3, PDEC3
56 JSR PDEC3
57 INC 10000.
58
59 STA 3, PDEC3
60 JSR PDEC3
61 INC 10000.

```

```

10034 ECL32
01 *****PRINT ROUTINES*****
02
03 STA 3, PDERET      !DECIMAL PRINT C(AC1).
04 JSR PDEC3          !RESET C(CARRY) FOR ZERO SUPPRESSION
05 INC 1000.         !SET C(CARRY) IF NOT
06
07 STA 3, PDEC3
08 JSR PDEC3
09 INC 1000.
10
11 STA 3, PDEC3
12 JSR PDEC3
13 INC 1000.
14
15 STA 3, PDEC3
16 JSR PDEC3
17 INC 1000.
18
19 STA 3, PDEC3
20 JSR PDEC3
21 INC 1000.
22
23 STA 3, PDERET
24 JSR PDEC3
25 INC 10000.
26
27 STA 3, PDEC3
28 JSR PDEC3
29 INC 10000.
30
31 STA 3, PDEC3
32 JSR PDEC3
33 INC 10000.
34
35 STA 3, PDEC3
36 JSR PDEC3
37 INC 10000.
38
39 STA 3, PDEC3
40 JSR PDEC3
41 INC 10000.
42
43 STA 3, PDEC3
44 JSR PDEC3
45 INC 10000.
46
47 STA 3, PDEC3
48 JSR PDEC3
49 INC 10000.
50
51 STA 3, PDEC3
52 JSR PDEC3
53 INC 10000.
54
55 STA 3, PDEC3
56 JSR PDEC3
57 INC 10000.
58
59 STA 3, PDEC3
60 JSR PDEC3
61 INC 10000.

```

0035 ECL32  
01 01446 000746  
02  
03 01447 000000 CHRVS: 0

JMP

CHAR1

:TEMP SAVE FOR AC2

```
10036 ECL32  
01  
02  
03 :*****MESSAGE DATA BLOCK*****  
04 01450 005215 PASHES: .TXTE !<15><12>PASS !  
05 040520  
06 051523  
07 120240  
08 000000  
09 01455 005215 PERCEN: .TXTE !<15><12>% FAIL=!  
10 120245  
11 040706  
12 146311  
13 000275  
14 01462 005215 ERMSG: .TXTE !<15><12><15><12>ERROR IN PASS: !  
15 005215  
16 151305  
17 147722  
18 129322  
19 047311  
20 000240  
21 051501  
22 035123  
23 000240  
24 01474 005215 HEADER: .TXTE !<15><12><15><12>  
25 005215  
26 01476 151303 CRY AC0 AC1 AC2 AC3 LISTING LOGICAL<15><12>!  
27 004531  
28 141501  
29 120060  
30 040411  
31 130703  
32 046640  
33 141501  
34 120262  
35 040411  
36 031703  
37 146011  
38 051711  
39 144724  
40 043516  
41 146011  
42 043717  
43 141711  
44 146101  
45 005215  
46 000000  
47 01523 005215 MESIZ: .TXTE !<15><12>LAST LOGICAL ADDRESS=!  
48 040714  
49 152123  
50 146240  
51 043717  
52 141711  
53 146101  
54 040640  
55 042104  
56 142722  
57 051523  
58 000275  
59 01537 005215 KCRUF: .TXTE !<15><12>!  
60 000000
```

```

0037 ECL32
01 01541 142523 SETSN: .TXTE ISET DATA SWITCHS AND PRESS CONTINUEI
02 120324
03 040504
04 040724
05 051640
06 144727
07 141724
08 051510
09 040640
10 042116
11 050240
12 143722
13 051523
14 141640
15 047317
16 144724
17 052516
18 000305

```

```

10038 ECL32
01
02 ?
03 ? *****FIRST TEST*****
04 ?
05 01563 100510 BEGIN: XOR 0,0
06 ?
07 ?
08 ?
09 ?
10 ?
11 ?
12 ?
13 ?
14 01564 00401 JSR *+1 IS
15 01565 054216 STA 3,HELP ?(ADDR OF TEST+1) IS
16 ?
17 01566 006304 JSR @ENTIN ?(SAVED IN LDC. HELP
18 01567 000144 100. ?INITIALIZE TEST.
19 LOBFAD
20 RNDADR LOBF,LOBFU
21 RAND
22 01570 006307 JSR @ENTRA ?C(ACO)=RANDOM #
23 01571 006310 JSR @RNADR ?GET ADDRESS IN THE RANGE
24 01572 000314 LOBF ?C(LOWBF) AND C(LOBFU)
25 01573 000311 LOBFU
26 01574 040246 STA 0,ISLCO ?EISZ INSTRUCTION SHOULD
27 01575 040410 STA 0,*8. ?NOT CHANGE ANY AC'S
28 01576 102400 SUB 0,0 ?STATE OF CARRY
29 01577 042246 STA 0,@ISLCO ?
30 RAND
31 01600 006307 JSR @ENTRA ?C(ACO)=RANDOM #
32 01601 115000 MOV 0,3 ?ACO=SAC=RANDOM DATA
33 01602 104400 NEG 0,1 ?AC1=SAC2=(RNDM DATA)
34 01603 131040 MOVO 1,2 ?CARRY IS INITIALIZED
35 01604 112070 EISZ 0,0 ?BY MOVO
36 01606 000000
37 01606 116414 SUB# 0,3,SZR ?
38 ERROR
39 01613 132414 SUB# 1,2,SZR ?
40 ERROR
41 01620 101003 MOV 0,0,SNC ?
42 ERROR
43 LOOP
44 01625 006305 JSR @ENTLO ?ITERATE TEST ROUTINE

```

```

10039 ECL32      ISZAI:      EISZ,L,SZC,HIBFAD
01      ISZAI:      ISZAI:      ISZSW=0
02      ISZAI:      ISZAI:      EISZ,L,LOBFAD,I280,0,ADI
03      ISZAI:      ISZAI:      TSTLOC
04      ISZAI:      ISZAI:      JSR @+1
05      ISZAI:      ISZAI:      STA 3,HELP
06      ISZAI:      ISZAI:      SETUP 100.
07      ISZAI:      ISZAI:      JSR @ENTIN
08      ISZAI:      ISZAI:      JSR @ENTIN
09      ISZAI:      ISZAI:      RAND
10      ISZAI:      ISZAI:      RANDR HIGBF,HIBFU
11      ISZAI:      ISZAI:      RAND
12      ISZAI:      ISZAI:      JSR @ENTRA
13      ISZAI:      ISZAI:      JSR @RNADR
14      ISZAI:      ISZAI:      HIGBF
15      ISZAI:      ISZAI:      HIBFU
16      ISZAI:      ISZAI:      STA 0,ISLCO
17      ISZAI:      ISZAI:      STA 0,+8.
18      ISZAI:      ISZAI:      SUB 0,0
19      ISZAI:      ISZAI:      STA 0,@ISLCO
20      ISZAI:      ISZAI:      RAND
21      ISZAI:      ISZAI:      JSR @ENTRA
22      ISZAI:      ISZAI:      MOV 0,3
23      ISZAI:      ISZAI:      NEG 0,1
24      ISZAI:      ISZAI:      MOVZ 1,2
25      ISZAI:      ISZAI:      EISZ 0,0
26      ISZAI:      ISZAI:      SUB# 0,3,SZR
27      ISZAI:      ISZAI:      ERROR
28      ISZAI:      ISZAI:      SUB# 1,2,SZR
29      ISZAI:      ISZAI:      ERROR
30      ISZAI:      ISZAI:      MOV 0,0,SZC
31      ISZAI:      ISZAI:      ERROR
32      ISZAI:      ISZAI:      LOOP
33      ISZAI:      ISZAI:      JSR @ENTLO

10040 ECL32      ISZHO:      ISZHO:      ISZSW=0
01      ISZHO:      ISZHO:      EISZ,L,LOBFAD,I280,0,ADI
02      ISZHO:      ISZHO:      TSTLOC
03      ISZHO:      ISZHO:      JSR @+1
04      ISZHO:      ISZHO:      STA 3,HELP
05      ISZHO:      ISZHO:      SETUP 100.
06      ISZHO:      ISZHO:      JSR @ENTIN
07      ISZHO:      ISZHO:      JSR @ENTIN
08      ISZHO:      ISZHO:      RAND
09      ISZHO:      ISZHO:      RAND
10      ISZHO:      ISZHO:      JSR @ENTRA
11      ISZHO:      ISZHO:      STA 0,ISLCO
12      ISZHO:      ISZHO:      LOBFAD
13      ISZHO:      ISZHO:      RANDR LOWBF,LOBFU
14      ISZHO:      ISZHO:      RAND
15      ISZHO:      ISZHO:      JSR @ENTRA
16      ISZHO:      ISZHO:      JSR @RNADR
17      ISZHO:      ISZHO:      LOBF
18      ISZHO:      ISZHO:      LOBFU
19      ISZHO:      ISZHO:      STA 0,ISLCO
20      ISZHO:      ISZHO:      LDA 1,ISLCO
21      ISZHO:      ISZHO:      STA 1,@ISLCO
22      ISZHO:      ISZHO:      .NOLOC 0
23      ISZHO:      ISZHO:      STA 0,+2
24      ISZHO:      ISZHO:      STA 0,0
25      ISZHO:      ISZHO:      EISZ 0,0
26      ISZHO:      ISZHO:      MOV 0,0
27      ISZHO:      ISZHO:      LDA 0,ISLCO
28      ISZHO:      ISZHO:      ADI 1,0
29      ISZHO:      ISZHO:      LDA 1,@ISLCO
30      ISZHO:      ISZHO:      SUB# 0,1,SZR
31      ISZHO:      ISZHO:      ERROR
32      ISZHO:      ISZHO:      LOOP
33      ISZHO:      ISZHO:      JSR @ENTLO

      ;(AC0)=RANDOM #
      ;GET ADDRESS IN THE RANGE
      ;C(HIGBF) AND C(HIBFU)
      ;EISZ INSTRUCTION SHOULD
      ;NDI CHANGE ANY AC'S
      ;STATE OF CARRY
      ;
      ;C(AC0)=RANDOM #
      ;AC0=AC3=RANDOM DATA
      ;AC1=AC2=-(RANDOM DATA)
      ;CARRY IS INITIALISED
      ;BY MOVZ
      ;
      ;SUB# 0,3,SZR
      ;
      ;SUB# 1,2,SZR
      ;
      ;MOV 0,0,SZC
      ;
      ;ITERATE TEST ROUTINE

```



```

10041 ECL32      10042 ECL32
01 000001 1SZ81: 01 000002 1SZ82:
02 1SZSM=1      E1SZ2 E1SZ, H1BFAD, IZ81, 1, ADI      E1SZ, L0BFAD, IZ82, 2, ADI
03 TSTLOC      TSTLOC
04 *+          *+
05 3, HELP     3, HELP
06 SETUP 100.  SETUP 100.
07 JSR @ENTIN JSR @ENTIN
08 *+          *+
09 INITIALIZE TEST. INITIALIZE TEST.
10 JRC(A0)=RANDOM # JRC(A0)=RANDOM #
11 JRC(1SLC0)=RANDOM DATA JRC(1SLC0)=RANDOM DATA
12 H1BFAD H1BFAD
13 RNDADR H1BGF, H1BFU RNDADR L0MBF, L0BFU
14 RAND
15 01730 006307 JSR @ENTRA JSR @ENTRA
16 01731 006310 JSR @RNADR JSR @RNADR
17 01732 000312 H1BGF H1BGF
18 01733 000315 H1BFU H1BFU
19 01730 040247 STA 0, 1SLC1 JRC(1SLC1)=RANDOM ADDRESS
20 01735 029246 LDA 1, 1SLC0 JSTORE RANDOM DATA IN
21 01736 040247 STA 1, @1SLC1 JRANDOM ADDRESS
22 01737 168470 E1EF 1, IZ81, --, 1 J(RNDM ADDR-PC OF E1SZ+1)
23 01741 122400 SUB 1, 0 JIS STORED IN E1SZ
24 *NOLC 0 J
25 01742 040402 STA 0, *+2 J
26 01743 112470 IZ81: E1SZ 0, 1 JEXECUTE E1SZ INSTRUCTION
27 000000
28 01745 101000 MOV 0, 0 J(RNDM ADDR) MUST BE
29 01746 020246 LDA 0, 1SLC0 J=(RNDM DATA+1) FOR E1SZ
30 01747 100010 ADI 1, 0 JAND =(RNDM DATA-1) FOR
31 01750 026247 LDA 1, @1SLC1 JEDSZ INSTRUCTION
32 01751 106414 SUB# 0, 1, SZR
33 ERROR
34 LOOP
35 01756 006305 JSR @ENTLO JITERATE TEST ROUTINE

```

```

10044 ECL32
01 ISZSW=3 ISZB3:
02 EISZ2 EISZ,HIBFAD,IZB3,3,ADI
03 YSTLOC
04 02014 004401 JSR *+1 ;(ADDR OF TEST+1) IS
05 02015 054216 STA 3,HELP ;SAVED IN LOC. HELP
06 02016 006304 SETUP 100. ;INITIALIZE TEST.
07 02017 000144 JSR @ENTIN
08 100.
09 RAND
10 02020 006307 JSR @ENTRA
11 02021 040246 HIBFAD 0,ISLCO
12 HIBFAD
13 RNDADR HIGBF,HIBFU
14 RAND
15 02022 006307 JSR @ENTRA
16 02023 006310 JSR @RNADR
17 05024 000312 HIGBF
18 02025 000315 HIBFU
19 02026 040247 STA 0,ISLCO
20 02027 024246 LDA 1,@ISLCO
21 02030 046247 MOVZR 0,3,SZC ;(RNDM ADDR/2)*BIT 15
22 02031 115222 STA 1,@ISLCO
23 02032 114010 ADI 1,3
24 02033 143370 HLV 0
25 02034 040402 NOLCO 0
26 02035 040402 STA 0,.*+2
27 02035 115470 IZB3:
28 000000
29 02037 101000 MOV 0,0
30 02040 020246 LDA 0,ISLCO
31 02041 100010 ADI 1,0
32 02042 026247 LDA 1,@ISLCO
33 02043 106414 SUB# 0,1,SZR
34 ERROR
35 LOOP
36 02050 006305 JSR @ENTLO

```

```

10044 ECL32
01 ISZSW=3 ISZB3:
02 EISZ2 EISZ,HIBFAD,IZB3,3,ADI
03 YSTLOC
04 02051 004401 JSR *+1 ;(ADDR OF TEST+1) IS
05 02052 054216 STA 3,HELP ;SAVED IN LOC. HELP
06 02053 006304 SETUP 5. ;INITIALIZE TEST.
07 02054 000005 JSR @ENTIN
08 5.
09 HIBFAD HIGBF,HIBFU
10 RNDADR
11 02055 006307 JSR @ENTRA
12 02056 006310 JSR @RNADR
13 02057 000312 HIGBF
14 02058 000315 HIBFU
15 02061 176400 SUB 3,3
16 02062 054246 STA 3,ISLCO
17 02063 111000 MOV 0,2
18 02064 055000 STA 3,0,2
19 02065 040402 STA 0,.*+2
20 02066 112070 IZB4:
21 000000
22 02070 101001 MOV 0,0,SKP
23 02071 000417 JMP IZB40
24 02072 010246 ISZ ISLCO
25 02073 101001 MOV 0,0,SKP
26 ERROR
27 02100 034246 LDA 3,ISLCO
28 02101 025000 LDA 1,0,2
29 02102 136414 SUB# 1,3,SZR
30 ERROR
31 02107 000757 JMP IZB4
32 02110 034246 LDA 3,ISLCO
33 02111 174014 COMM 3,5,SZR
34 ERROR
35 LOOP
36 02116 006305 JSR @ENTLO

```

```

10044 ECL32
01 ISZSW=3 ISZB3:
02 EISZ2 EISZ,HIBFAD,IZB3,3,ADI
03 YSTLOC
04 02051 004401 JSR *+1 ;(ADDR OF TEST+1) IS
05 02052 054216 STA 3,HELP ;SAVED IN LOC. HELP
06 02053 006304 SETUP 5. ;INITIALIZE TEST.
07 02054 000005 JSR @ENTIN
08 5.
09 HIBFAD HIGBF,HIBFU
10 RNDADR
11 02055 006307 JSR @ENTRA
12 02056 006310 JSR @RNADR
13 02057 000312 HIGBF
14 02058 000315 HIBFU
15 02061 176400 SUB 3,3
16 02062 054246 STA 3,ISLCO
17 02063 111000 MOV 0,2
18 02064 055000 STA 3,0,2
19 02065 040402 STA 0,.*+2
20 02066 112070 IZB4:
21 000000
22 02070 101001 MOV 0,0,SKP
23 02071 000417 JMP IZB40
24 02072 010246 ISZ ISLCO
25 02073 101001 MOV 0,0,SKP
26 ERROR
27 02100 034246 LDA 3,ISLCO
28 02101 025000 LDA 1,0,2
29 02102 136414 SUB# 1,3,SZR
30 ERROR
31 02107 000757 JMP IZB4
32 02110 034246 LDA 3,ISLCO
33 02111 174014 COMM 3,5,SZR
34 ERROR
35 LOOP
36 02116 006305 JSR @ENTLO

```

```

10045 ECL32
01 000000 ISZC0:
02 ISZSW=0
03 EISZ IZC0,0,ADI
04 TSLOC
05 JSR *+1
06 STA 3,HELP
07 SETUP 100.
08 JSR @ENTR
09 RAND
10 02123 006307
11 02124 000144
12 JSR @ENTR
13 STA 0,ISLCO
14 RNDADR LOWBF,LOBFU
15 RAND
16 02125 006307
17 02126 006310
18 02127 000314
19 02130 000311
20 02131 040247
21 02132 024246
22 02133 046247
23
24 RNDADR HIGBF,HIBFU
25 RAND
26 02134 006307
27 02135 006310
28 02136 000312
29 02137 000315
30 02140 111000
31 02141 024247
32 02142 045000
33 .NOLDC
34 ADDR 0,0
35 02144 040402
36 02145 112070 IZC0:
37 02147 101000
38 02150 020246
39 02151 100010
40 02152 026247
41 02153 106414
42 ERROR
43 LOOP
44 JSR @ENTLO
45 02160 006305

10046 ECL32
01 000001 ISZC1:
02 ISZSW=1
03 EISZ IZC1,1,ADI
04 TSLOC
05 JSR *+1
06 STA 3,HELP
07 SETUP 100.
08 JSR @ENTR
09 RAND
10 02165 006307
11 02166 040246
12 JSR @ENTR
13 STA 0,ISLCO
14 RNDADR LOWBF,LOBFU
15 RAND
16 02167 006307
17 02170 006310
18 02171 000314
19 02172 000311
20 02173 040247
21 02174 024246
22 02175 046247
23
24 RNDADR HIGBF,HIBFU
25 RAND
26 02176 006307
27 02200 000312
28 02201 000315
29 02202 111000
30 02203 024247
31 02204 045000
32 02205 166470
33 02207 122400
34 .NOLDC
35 ADDR 0,0
36 02211 040402
37 02212 112470 IZC1:
38 02214 100000
39 02215 101000
40 02216 020246
41 02217 100010
42 02217 026247
43 02220 106414
44 ERROR
45 LOOP
46 02225 006305

;(ACCO)=RANDOM #
;SET ADDRESS IN THE RANGE
;C(LOWBF) AND C(HIBFU)
;STORE RANDOM DATA INTO
;RANDOM ADDRESS 1
;
;(ACCO)=RANDOM #
;SET ADDRESS IN THE RANGE
;C(HIGBF) AND C(HIBFU)
;STORE RANDOM ADDRESS 1 INTO
;RANDOM ADDRESS 2
;
;(RNDM ADDR 2-PC OF EISZ+1)
;IS IN ACC
;ADD INDIRECT BIT TO ACC
;AND STORE IN EISZ
;EXECUTE EISZ INSTRUCTION
;IN CASE IF EISZ SKIPS
;(RNDM ADDR 1) MUST BE
;(RNDM DATA+1) FOR EISZ AND
;(RNDM DATA-1) FOR EDSZ
;
;ITERATE TEST ROUTINE

```

```

10047 ECL32          ISZSW=2  ISZC2:  ISZC2:  ISZC2:  ISZC2:  ISZC2:  ISZC2:  ISZC2:  ISZC2:  ISZC2:  ISZC2:
01 000002 000002 000002 000002 000002 000002 000002 000002 000002 000002
02 ISZ3     EISZ,I2C2,2,ADI
03 TSTLOC
04 02226 004001      ;(ADDR OF TEST+1) IS
05 02227 054216      ;SAVED IN LOC. HELP
06 SETUP 100.
07 02230 006304      JSR @ENTIN
08 02231 000144      100.
09 RAND
10 02232 006307      JSR @ENTRA
11 02233 040246      STA 0,ISLCO
12 LOBFAD
13 RNDADR LOBF,LOBFU
14 RAND
15 02234 006307      JSR @ENTRA
16 02235 006310      JSR @RNADR
17 02236 000314      LOBF
18 02237 000311      LOBFU
19 02240 040247      STA 0,ISLCO
20 02241 024246      LDA 1,ISLCO
21 02242 046247      STA 1,@ISLCO
22 HIBFAD
23 RNDADR HIBF,HIBFU
24 RAND
25 02243 006307      JSR @ENTRA
26 02244 006310      JSR @RNADR
27 02245 000312      HIBF
28 02246 000315      HIBFU
29 02247 110000      MOV 0,2
30 02250 024247      LDA 1,ISLCO
31 02251 045000      STA 1,0,2
32 02252 111222      MOVZR 0,2,SZC
33 02253 110010      ADI 1,2
34 02254 143370      HLV 0
35 02255 000000      -NOLCO
36 02256 102240      ADDR 0,0
37 02257 040402      STA 0,+2
38 02258 113070      EISZ @0,2
39 100000
40 02261 101000      MOV 0,0
41 02262 020246      LDA 0,ISLCO
42 02263 100010      ADI 1,0
43 02264 026247      LDA 1,@ISLCO
44 02265 106414      SUB# 0,1,SZR
45 ERROR
46 LOOP
47 02272 006305      JSR @ENTLO

10048 ECL32          ISZSW=3  ISZC3:  ISZC3:  ISZC3:  ISZC3:  ISZC3:  ISZC3:  ISZC3:  ISZC3:  ISZC3:  ISZC3:
01 000003 000003 000003 000003 000003 000003 000003 000003 000003 000003
02 ISZ3     EISZ,I2C3,3,ADI
03 TSTLOC
04 02273 004401      JSR @ENTRA
05 02274 054216      STA 3,HELP
06 SETUP 100.
07 02275 006304      JSR @ENTIN
08 02276 000144      100.
09 RAND
10 02277 006307      JSR @ENTRA
11 02278 040246      STA 0,ISLCO
12 LOBFAD
13 RNDADR LOBF,LOBFU
14 RAND
15 02281 006307      JSR @ENTRA
16 02282 006310      JSR @RNADR
17 02283 000314      LOBF
18 02284 000311      LOBFU
19 02285 040247      STA 0,ISLCO
20 02286 024246      LDA 1,ISLCO
21 02287 046247      STA 1,@ISLCO
22 HIBFAD
23 RNDADR HIBF,HIBFU
24 RAND
25 02291 006307      JSR @ENTRA
26 02292 006310      JSR @RNADR
27 02293 000312      HIBF
28 02294 000315      HIBFU
29 02295 111000      MOV 0,2
30 02296 024247      LDA 1,ISLCO
31 02297 045000      STA 1,0,2
32 02298 115222      MOVZR 0,2,SZC
33 02299 114010      ADI 1,5
34 02300 143370      HLV 0
35 02301 000000      -NOLCO
36 02302 103240      ADDR 0,0
37 02303 040402      STA 0,+2
38 02304 113470      EISZ @0,3
39 100000
40 02326 101000      MOV 0,0
41 02327 020246      LDA 0,ISLCO
42 02328 100010      ADI 1,0
43 02329 026247      LDA 1,@ISLCO
44 02330 106414      SUB# 0,1,SZR
45 ERROR
46 LOOP
47 02337 006305      JSR @ENTLO

;C(ACO)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(LOWBF) AND C(HIBFU)
;STORE RANDOM DATA INTO
;RANDOM ADDRESS 1
;C(ACO)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(HIGBF) AND C(HIBFU)
;STORE RANDOM ADDRESS 1 INTO
;RANDOM ADDRESS 2
;(RNDM ADDR 2/2)+BIT 15
;IS STORED IN AC3 AND
;ACO=(RNDM ADDR 2)/2
;ADD INDIRECT BIT TO ACO
;AND STORE IN EISZ
;EXECUTE EISZ INSTRUCTION
;IN CASE IF EISZ SKIPS
;(RNDM ADDR 1) MUST BE
;(RNDM DATA+1) FOR EISZ AND
;(RNDM DATA-1) FOR EDSZ
;ITERATE TEST ROUTINE

```

|       |                            |       |                            |
|-------|----------------------------|-------|----------------------------|
| 10049 | ECL32                      | 10050 | ECL32                      |
| 01    | 000000                     | 01    | 000001                     |
| 02    | ISZSW=0                    | 02    | ISZD1:                     |
| 03    | EISZ,IZD0,0,ADI,IZD00      | 03    | EISZ,IZD1,1,ADI,IZD10      |
| 04    | TSTLOC                     | 04    | TSTLOC                     |
| 05    | *(ADDR OF TEST+1) IS       | 05    | *(ADDR OF TEST+1) IS       |
| 06    | *(SAVED IN LOC. HELP       | 06    | *(SAVED IN LOC. HELP       |
| 07    | SETUP 100.                 | 07    | SETUP 100.                 |
| 08    | JSR @ENTIN                 | 08    | JSR @ENTIN                 |
| 09    | INITIALIZE TEST.           | 09    | INITIALIZE TEST.           |
| 10    | RAND                       | 10    | RAND                       |
| 11    | :(AC0)=RANDOM #            | 11    | :(AC0)=RANDOM #            |
| 12    | :(ISLCO)=RANDOM DATA       | 12    | :(ISLCO)=RANDOM DATA       |
| 13    | LOBFAD                     | 13    | LOBFAD                     |
| 14    | RNDADR LOWBF,LOBFU         | 14    | RNDADR LOWBF,LOBFU         |
| 15    | RAND                       | 15    | RAND                       |
| 16    | :(AC0)=RANDOM #            | 16    | :(AC0)=RANDOM #            |
| 17    | :(GET ADDRESS IN THE RANGE | 17    | :(GET ADDRESS IN THE RANGE |
| 18    | :(C(LOWBF) AND C(LOBFU)    | 18    | :(C(LOWBF) AND C(LOBFU)    |
| 19    | :(ISLCO)=RNDM ADDR 1       | 19    | :(ISLCO)=RNDM ADDR 1       |
| 20    | :(STORE RNDM DATA INTO     | 20    | :(STORE RNDM DATA INTO     |
| 21    | :(RNDM ADDR 1              | 21    | :(RNDM ADDR 1              |
| 22    | HIBFAD                     | 22    | HIBFAD                     |
| 23    | RNDADR HIGBF,HIBFU         | 23    | RNDADR HIGBF,HIBFU         |
| 24    | RAND                       | 24    | RAND                       |
| 25    | :(AC0)=RANDOM #            | 25    | :(AC0)=RANDOM #            |
| 26    | :(GET ADDRESS IN THE RANGE | 26    | :(GET ADDRESS IN THE RANGE |
| 27    | :(C(HIGBF) AND C(HIBFU)    | 27    | :(C(HIGBF) AND C(HIBFU)    |
| 28    | :(ISLCO)=RNDM ADDR 2       | 28    | :(ISLCO)=RNDM ADDR 2       |
| 29    | :(STORE RNDM ADDR 1 IN     | 29    | :(STORE RNDM ADDR 1 IN     |
| 30    | :(RNDM ADDR 2              | 30    | :(RNDM ADDR 2              |
| 31    | HIBFAD                     | 31    | HIBFAD                     |
| 32    | RNDADR HIGBF,HIBFU         | 32    | RNDADR HIGBF,HIBFU         |
| 33    | RAND                       | 33    | RAND                       |
| 34    | :(AC0)=RANDOM #            | 34    | :(AC0)=RANDOM #            |
| 35    | :(GET ADDRESS IN THE RANGE | 35    | :(GET ADDRESS IN THE RANGE |
| 36    | :(C(HIGBF) AND C(HIBFU)    | 36    | :(C(HIGBF) AND C(HIBFU)    |
| 37    | :(ISLCO)=RNDM ADDR 3       | 37    | :(ISLCO)=RNDM ADDR 3       |
| 38    | :(MAKE SURE THAT RNDM ADDR | 38    | :(MAKE SURE THAT RNDM ADDR |
| 39    | :(2 AND 3 ARE DIFFERENT    | 39    | :(2 AND 3 ARE DIFFERENT    |
| 40    | :(AND SAVE @RNDM ADDR 2    | 40    | :(AND SAVE @RNDM ADDR 2    |
| 41    | :(IN RNDM ADDR 3           | 41    | :(IN RNDM ADDR 3           |
| 42    | :(ADD INDIRECT BIT TO AC0  | 42    | :(ADD INDIRECT BIT TO AC0  |
| 43    | :(AND STORE IN EISZ        | 43    | :(AND STORE IN EISZ        |
| 44    | :(EXECUTE EISZ INSTRUCTION | 44    | :(EXECUTE EISZ INSTRUCTION |
| 45    | :(MOV 0                    | 45    | :(MOV 0                    |
| 46    | :(NOLOC                    | 46    | :(NOLOC                    |
| 47    | :(ADDOR 0,0                | 47    | :(ADDOR 0,0                |
| 48    | :(STA 0,0                  | 48    | :(STA 0,0                  |
| 49    | :(EISZ @0,0                | 49    | :(EISZ @0,0                |
| 50    | :(MOV 0,0                  | 50    | :(MOV 0,0                  |
| 51    | :(LOA 0,ISLCO              | 51    | :(LOA 0,ISLCO              |
| 52    | :(ADI 1,0                  | 52    | :(ADI 1,0                  |
| 53    | :(LOA 1,@ISLCO             | 53    | :(LOA 1,@ISLCO             |
| 54    | :(SUB# 0,1,SR              | 54    | :(SUB# 0,1,SR              |
| 55    | :(ERROR                    | 55    | :(ERROR                    |
| 56    | :(LOOP                     | 56    | :(LOOP                     |
| 57    | :(JSR @ENTLO               | 57    | :(JSR @ENTLO               |
| 58    | :(ITERATE TEST ROUTINE     | 58    | :(ITERATE TEST ROUTINE     |
| 59    | :(JMP 0,0                  | 59    | :(JMP 0,0                  |

```

10051 ECL32      01 000002 1S202:  ISZSW=2      EISZ,IZD2,2,ADI,IZD20
02 01574          EISZ,IZD3,3,ADI,IZD30
03 02473 004401  JSR @ENTRA  *+1      *(ADDR OF TEST+1) IS
04 02474 054216  STA 3-HELP    *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
05 02475 054216  SETUP 100.    *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
06 02476 063304  JSR @ENTIN   *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
07 02477 063304  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
08 02478 000144  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
09 02479 000144  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
10 02477 063307  JSR @ENTRA  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
11 02500 040246  STA 0,ISLCO  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
12 02501 040246  LOBFAD      *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
13 02502 040246  RNDADR LOWBF,LOBFU
14 02503 000314  RAND
15 02501 063307  JSR @ENTRA  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
16 02502 063310  JSR @RNADR  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
17 02503 000314  LOWBF      *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
18 02504 000311  LOBFU      *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
19 02505 040247  STA 0,ISLCO  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
20 02506 040246  LDA 1,ISLCO  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
21 02507 046247  STA 1,@ISLCO *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
22 02508 046247  HIRFAD      *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
23 02509 046247  RNDADR HIGHF,HIBFU
24 02510 063307  RAND
25 02510 063307  JSR @ENTRA  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
26 02511 063310  JSR @RNADR  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
27 02512 000312  HIGHF      *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
28 02513 000315  HIBFU      *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
29 02514 040250  STA 0,ISLCO  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
30 02515 042427  LDA 1,ISLCO  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
31 02516 046250  STA 1,@ISLCO *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
32 02517 046250  HIRFAD      *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
33 02518 046250  RNDADR HIGHF,HIBFU
34 02519 046250  RAND
35 02517 063307  JSR @ENTRA  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
36 02520 063310  JSR @RNADR  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
37 02521 000312  HIGHF      *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
38 02522 000315  HIBFU      *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
39 02523 024250  LDA 1,ISLCO  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
40 02524 106415  SUB# 0,1,SNR *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
41 02525 000772  JMP IZ020    *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
42 02526 111000  MOV 0,2     *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
43 02527 127240  ADDR 1,1   *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
44 02530 045000  STA 1,0,2  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
45 02531 111222  MOVZR 0,2,SZC *(RNDM ADDR 3/2)+BIT 15 IS
46 02532 110010  ADI 1,2     *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
47 02533 143370  HLW 0       *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
48 02534 000000  *NOLOC     *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
49 02534 103240  ADDR 0,0   *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
50 02535 040402  STA 0,+*2  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
51 02536 113070  EISZ @0,2  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
52 02537 100000  IZ02:
53 02540 101000  MOV 0,0    *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
54 02541 020246  LDA 0,ISLCO *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
55 02542 100010  ADI 1,0    *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
56 02543 026247  LDA 1,@ISLCO *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
57 02544 106414  SUB# 0,1,SZR *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1
58 02545 106414  ERROR
59 02546 106414  LOOP
60 02551 063305  JSR @ENTLO  *+1      *+1      *+1      *+1      *+1      *+1      *+1      *+1

```

```

0053 ECL32
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

; CHECKING OUT LONG "EDSZ" INSTRUCTION
;
DSZA0: EISZ1 EDZSZ,Z,SZC,LOBFAD
        TSILOC
        JSR @*1
        STA 3,HELP
        SETUP 100.
        JSR @ENTIN
        ; INITIALIZE TEST.
        LOBFAD
        RNDADR LOWBF,LOBFU
        RAND
        JSR @ENTRA
        JSR @RNADR
        LOWBF
        LOBFU
        STA 0,ISLCO
        STA 0,*8.
        SUB 0,0
        STA 0,@ISLCO
        RAND
        JSR @ENTRA
        MOV 0,3
        NEG 0,1
        MOVZ 1,2
        EDZSZ
        SUB# 0,3,SZR
        ERROR
        SUB# 1,2,SZR
        ERROR
        MOV 0,0,SZC
        LOOP
        JSR @ENTLO
        ; ITERATE TEST ROUTINE

10054 ECL32
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

DSZA1: EISZ1 EDZSZ,O,SNC,HIBFAD
        TSILOC
        JSR @*1
        STA 3,HELP
        SETUP 100.
        JSR @ENTIN
        ; INITIALIZE TEST.
        HIBFAD
        RNDADR HIGBF,HIRFU
        RAND
        JSR @ENTRA
        JSR @RNADR
        HIGBF
        HIRFU
        STA 0,ISLCO
        STA 0,*8.
        SUB 0,0
        STA 0,@ISLCO
        RAND
        JSR @ENTRA
        MOV 0,3
        NEG 0,1
        MOVZ 1,2
        EDZSZ
        SUB# 0,3,SZR
        ERROR
        SUB# 1,2,SZR
        ERROR
        MOV 0,0,SNC
        LOOP
        JSR @ENTLO
        ; ITERATE TEST ROUTINE

;C(CACO)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(HIGRF) AND C(HIRFU)
;EDSZ INSTRUCTION SHOULD
;NOT CHANGE ANY AC'S
;STATE OF CARRY
;
;C(CACO)=RANDOM #
;ACO=AC3=RANDOM DATA
;AC1=AC2=(RNDM DATA)
;CARRY IS INITIALIZED
;BY MOVZ
;
;C(CACO)=RANDOM #
;ACO=AC3=RANDOM DATA
;AC1=AC2=(RNDM DATA)
;CARRY IS INITIALIZED
;BY MOVZ
;
;C(CACO)=RANDOM #
;ACO=AC3=RANDOM DATA
;AC1=AC2=(RNDM DATA)
;CARRY IS INITIALIZED
;BY MOVZ
;
; ITERATE TEST ROUTINE

```

```

10055 ECL32
01 000000 03Z80: ISZSW=0
02 EISZ2 EDZS,LOBFAD,DZ80,0,SBI
03 TSTLOC *+J
04 02735 004401 *(ADDR OF TEST+1) IS
05 02736 054216 *SAVED IN LOC. HELP
06 SETUP 100.
07 02737 006304 *INITIALIZE TEST.
08 02740 000144 RAND
09
10 02741 006307 JSR @ENTRA
11 02742 040246 *(C(ACO)=RANDOM #
*(ISLCO)=RANDOM DATA
12 LOBFAD
13 RUDADR LOBF,LOBFU
14 RAND
15 02743 006307 JSR @ENTRA
16 02744 006310 JSR @RNADR
17 02745 000314 LOBF
18 02746 000311 LOBFU
19 02747 040247 STA 0,ISLC1
20 02750 024246 LDA 1,ISLC0
21 02751 046247 STA 1,@ISLC1
22 02752 000000 *NOLOC 0
23 02752 040402 STA 0,**+2
24 02755 116070 DZ80: *EXECUTE EDZS INSTRUCTION
25 02755 000000 MOV 0,0
26 02755 101000 LDA 0,ISLC0
27 02756 020246 SBI 1,0
28 02757 100110 *AND =(RNDM DATA-1) FOR
29 02760 026247 *EDZS INSTRUCTION
30 02761 106414 LDA 1,@ISLC1
31 SUB# 0,1,SZR
32 ERROR
33 LOOP
34 JSR @ENTLO
35 02766 006305 *ITERATE TEST ROUTINE
;

10056 ECL32
01 000001 03Z81: ISZSW=1
02 EISZ2 EDZS,HIBFAD,OZ81,1,SBI
03 TSTLOC *+J
04 02767 004401 *(ADDR OF TEST+1) IS
05 02770 054216 *SAVED IN LOC. HELP
06 SETUP 100.
07 02771 006304 *INITIALIZE TEST.
08 02772 000144 RAND
09
10 02773 006307 JSR @ENTRA
11 02774 040246 *(C(ACO)=RANDOM #
*(ISLCO)=RANDOM DATA
12 HIBFAD
13 RUDADR HIBGF,HIBFU
14 RAND
15 02775 006307 JSR @ENTRA
16 02776 006310 JSR @RNADR
17 02777 000312 HIBGF
18 03000 000315 HIBFU
19 03001 040247 STA 0,ISLC1
20 03002 024246 LDA 1,ISLC0
21 03003 046247 STA 1,@ISLC1
22 03004 166470 *RANDOM ADDR-PC OF EDZS+1)
23 03006 122400 *IS STORED IN EDZS
24 *NOLOC 0
25 03007 040402 STA 0,**+2
26 03010 116470 DZ81: *EXECUTE EDZS INSTRUCTION
27 03012 101000 MOV 0,0
28 03013 101000 LDA 0,ISLC0
29 03015 020246 *AND =(RNDM DATA+1) FOR EISZ
30 03014 100110 SBI 1,0
31 03015 026247 *EDZS INSTRUCTION
32 03016 106414 LDA 1,@ISLC1
33 SUB# 0,1,SZR
34 ERROR
35 03023 006305 JSR @ENTLO
;

```



```

10057 ECL32      000002 03282:  ISZSMF2  ISZSMF2
01 01 000002 03282:  EISZ2    EDSZ,HIBFAD,0282,2,SBI
02 02 03024 04401    T31LOC   *+1      ;(ADDR OF TEST+1) IS
03 03 03025 054216   STA      3,HELP  ;SAVED IN LOC. HELP
04 04 03026 06304    SETUP   100.    ;INITIALIZE TEST.
05 05 03027 000144   JSR @ENTIN
06 06 03027 000144   100.
07 07 03028 006304   RAND
08 08 03029 006307   JSR @ENTRA
09 09 03030 040246   STA      0,ISLCO ;C(AC0)=RANDOM #
10 10 03031 040246   HIBFAD   ;C(ISLCO)=RANDOM DATA
11 11 03032 006307   RAND
12 12 03033 006310   JSR @RNADR
13 13 03034 000312   HIBGF   ;C(AC0)=RANDOM #
14 14 03035 000315   HIBFU   ;C(HIBGF) AND C(HIBFU)
15 15 03036 040247   STA      0,ISLCO ;C(ISLCO)=RANDOM ADDRESS
16 16 03037 024246   LDA      1,ISLCO ;STORE RANDOM DATA IN
17 17 03038 046247   STA      1,@ISLCO ;RANDOM ADDRESS
18 18 03039 110222   MOVZR   0,2,SZC ;(RNDM ADDR/2)+BIT 15
19 19 03040 110010   ADI     1,2     ;IS STORED IN AC2 AND
20 20 03041 143370   HLV     0      ;(RNDM ADDR/2) IN EDSZ
21 21 03042 000000   -MOLCO 0
22 22 03043 040402   STA     0,+2   ;
23 23 03044 117070 0282:  EDSZ     ;EXECUTE EDSZ INSTRUCTION
24 24 03045 000000
25 25 03046 101000   MOV     0,0
26 26 03047 020246   LDA     0,ISLCO ;C(RNDM ADDR) MUST BE
27 27 03048 100110   SBI     1,0    ;=(RNDM DATA+1) FOR EISZ
28 28 03049 026247   LDA     1,@ISLCO ;AND =(RNDM DATA-1) FOR
29 29 03050 106414   SUB#    0,1,SZR ;EDSZ INSTRUCTION
30 30 03051 106414   ERROR
31 31 03052 106414   LOOP
32 32 03053 006305   JSR @ENTLO
33 33 03054 006305   ;ITERATE TEST ROUTINE
34 34
35 35
36 36 03060 006305

```

```

10058 ECL32      000003 03283:
01 01 000003 03283:  ISZSMF3  ISZSMF3
02 02 03061 04401    EISZ2    EDSZ,LORFAD,0283,3,SBI
03 03 03062 054216   T31LOC   *+1      ;(ADDR OF TEST+1) IS
04 04 03063 04401    STA      3,HELP  ;SAVED IN LOC. HELP
05 05 03064 054216   SETUP   100.    ;INITIALIZE TEST.
06 06 03065 046304   JSR @ENTIN
07 07 03066 000144   100.
08 08 03067 006304   RAND
09 09 03068 006307   JSR @ENTRA
10 10 03069 040246   STA      0,ISLCO ;C(AC0)=RANDOM #
11 11 03070 040246   HIBFAD   ;C(ISLCO)=RANDOM DATA
12 12 03071 000314   LORBF   ;C(AC0)=RANDOM #
13 13 03072 000311   LORFU   ;C(LORBF) AND C(LORFU)
14 14 03073 040247   STA      0,ISLCO ;C(ISLCO)=RANDOM ADDRESS
15 15 03074 024246   LDA      1,ISLCO ;STORE RANDOM DATA IN
16 16 03075 046247   STA      1,@ISLCO ;RANDOM ADDRESS
17 17 03076 115222   MOVZR   0,2,SZC ;(RNDM ADDR/2)+BIT 15
18 18 03077 142010   ADI     1,3     ;IS STORED IN AC2 AND
19 19 03078 143370   HLV     0      ;(RNDM ADDR/2) IN EDSZ
20 20 03079 000000   -MOLCO 0
21 21 03080 040402   STA     0,+2   ;
22 22 03081 117470 0283:  EDSZ     ;EXECUTE EDSZ INSTRUCTION
23 23 03082 000000
24 24 03083 101000   MOV     0,0
25 25 03084 020246   LDA     0,ISLCO ;C(RNDM ADDR) MUST BE
26 26 03085 100110   SBI     1,0    ;=(RNDM DATA+1) FOR EISZ
27 27 03086 026247   LDA     1,@ISLCO ;AND =(RNDM DATA-1) FOR
28 28 03087 106414   SUB#    0,1,SZR ;EDSZ INSTRUCTION
29 29 03088 106414   ERROR
30 30 03089 106414   LOOP
31 31 03090 006305   JSR @ENTLO
32 32 03091 006305   ;ITERATE TEST ROUTINE
33 33
34 34
35 35
36 36 03115 006305

```

```

10059 ECL32
01
02 DSZB4: TSTLOC
03 JSR @ENTRA
04 STA 3,HELP
05 SETUP 5.
06 JSR @ENTIN
07 JSR @ENTIN
08 L@BFAD
09 RNDADR L@BWF,L@BFU
10 RAND
11 03122 06307
12 03123 06310
13 03124 00314
14 03125 00311
15 03126 176400
16 03127 054246
17 03130 111000
18 03131 055000
19 03132 040402
20 03133 116070
21 000000
22 03135 101001
23 03136 00417
24 03137 014246
25 03140 101001
26 ERROR
27 03145 034246
28 03146 025000
29 03147 136414
30 ERROR
31 03154 000757
32 03155 034246
33 03156 020233
34 03157 162414
35 ERROR
36 LOOP
37 JSR @ENTLO
38 03164 006305

10060 ECL32
01
02 ISZM=0
03 ELSZ3 EDZS,DZCO,0,SBI
04 TSTLOC
05 03165 004401
06 03166 034216
07 STA 3,HELP
08 JSR @ENTIN
09 SETUP 100.
10 JSR @ENTIN
11 RAND
12 03171 06307
13 03172 040246
14 L@BFAD
15 RNDADR L@BWF,L@BFU
16 RAND
17 03173 06307
18 03174 06310
19 03175 00314
20 03176 00311
21 03200 024246
22 03201 046247
23 HTBFAD
24 RNDADR HIGBF,HIBFU
25 RAND
26 03202 06307
27 03203 06310
28 03204 00312
29 03205 00315
30 03206 111000
31 03207 024247
32 03210 045000
33 N@LOC
34 03211 103240
35 03212 040402
36 03213 116070
37 100000
38 03215 101000
39 03216 020246
40 03217 100110
41 03220 026247
42 03221 106414
43 ERROR
44 LOOP
45 JSR @ENTLO
46 03226 006305

;(ADDR OF TEST+1) IS
;SAVED IN LOC. HELP
;INITIALIZE TEST.
;RANDOM ADDR FROM L@BWF
;IS STORED IN EDZ INSTRUCTION
;(RNDM ADDR)=0
;EDZ MUST DECREMENT IT
;BY 1 EVERYTIME IT IS
;EXECUTED AND SKIP WHEN
;(RNDM ADDR) BECOMES 0
;(ISLCO) KEEPS TRACK OF
;(RNDM ADDR)
;(ISLCO) MUST BE 1
;AT THIS POINT
;ITERATE TEST ROUTINE
;ITERATE TEST ROUTINE

```

```

10061 ECL32      ISZSM=1      DSZC1:
01 000001 DSZC1:
02 EISZ5      EDSZ,DZC1,1,SBI
03 TSTLOC
04 JSR *+1      ;(ADDR OF TEST+1) IS
05 STA 3,HELP   ;SAVED IN LOC. HELP
06 SETUP 100.
07 JSR @ENTIN   ;INITIALIZE TEST.
08 RAND
09 JSR @ENTRA
10 STA 0,ISLCO  ;C(AC0)=RANDOM #
11 RNDADR LOWBF,LOBFU ;C(ISLCO)=RANDOM DATA
12 RND
13 JSR @ENTRA
14 JSR @RNADR    ;C(AC0)=RANDOM #
15 LOBF         ;GET ADDRESS IN THE RANGE
16 LOBF         ;C(LOWBF) AND C(HIBFU)
17 STA 0,ISLCO ;STORE RANDOM DATA INTO
18 LDA 1,ISLCO ;RANDOM ADDRESS 1
19 STA 1,@ISLCO
20 HIBFAD      HIGBF,HIBFU
21 RND
22 JSR @ENTRA
23 JSR @RNADR    ;C(AC0)=RANDOM #
24 LOBF         ;GET ADDRESS IN THE RANGE
25 LOBF         ;C(HIGBF) AND C(HIBFU)
26 MOV 0,2      ;STORE RANDOM ADDRESS 1 INTO
27 LDA 1,ISLCO ;RANDOM ADDRESS 2
28 STA 1,0,2
29 EIEF 1,DZC1--1
30 SUB 1,0
31 .NOLCO 0
32 ADDR 0,0
33 STA 0,0,2
34 EDSZ @0,1
35 MOV 0,0
36 LDA 0,ISLCO ;C(RNDM ADDR 1) MUST BE
37 SBI 1,0     ;(RNDM DATA+1) FOR EISZ AND
38 LDA 1,@ISLCO ;(RNDM DATA-1) FOR EDSZ
39 SUB# 0,1,SZR
40 ERROR
41 LOOP
42 JSR @ENTLO  ;ITERATE TEST ROUTINE
43
10062 ECL32      ISZSM=2      DSZC2:
01 000002 DSZC2:
02 EISZ5      EDSZ,DZC2,2,SBI
03 TSTLOC
04 JSR *+1      ;(ADDR OF TEST+1) IS
05 STA 3,HELP   ;SAVED IN LOC. HELP
06 SETUP 100.
07 JSR @ENTIN   ;INITIALIZE TEST.
08 RAND
09 JSR @ENTRA
10 STA 0,ISLCO  ;C(AC0)=RANDOM #
11 RNDADR LOWBF,LOBFU ;C(ISLCO)=RANDOM DATA
12 RND
13 JSR @ENTRA
14 JSR @RNADR    ;C(AC0)=RANDOM #
15 LOBF         ;GET ADDRESS IN THE RANGE
16 LOBF         ;C(LOWBF) AND C(HIBFU)
17 STA 0,ISLCO ;STORE RANDOM DATA INTO
18 LDA 1,ISLCO ;RANDOM ADDRESS 1
19 STA 1,@ISLCO
20 HIBFAD      HIGBF,HIBFU
21 RND
22 JSR @ENTRA
23 JSR @RNADR    ;C(AC0)=RANDOM #
24 LOBF         ;GET ADDRESS IN THE RANGE
25 LOBF         ;C(HIGBF) AND C(HIBFU)
26 MOV 0,2      ;STORE RANDOM ADDRESS 1 INTO
27 LDA 1,ISLCO ;RANDOM ADDRESS 2
28 STA 1,0,2
29 MOVZR 0,2,SZC
30 ADI 1,2
31 HLW 0
32 .NOLCO 0
33 ADDR 0,0
34 STA 0,0,2
35 EDSZ @0,2
36 MOV 0,0
37 LDA 0,ISLCO ;C(RNDM ADDR 1) MUST BE
38 SBI 1,0     ;(RNDM DATA+1) FOR EISZ AND
39 LDA 1,@ISLCO ;(RNDM DATA-1) FOR EDSZ
40 SUB# 0,1,SZR
41 ERROR
42 LOOP
43 JSR @ENTLO  ;ITERATE TEST ROUTINE
44

```

```

10063 ECL32      01 000003 08ZC3: ISZSN=3      EDSZ,DZC3,3,SBI
02 EISZ3          02          EDSZ,DZ00,0,SBI,DZD00
03 TSTLOC         03          TSTLOC
04 JSR *+1       04          *(ADDR OF TEST+1) IS
05 STA 3,HELP   05          *SAVED IN LOC. HELP
06 SETUP 100.   06          *INITIALIZE TEST.
07 JSR @ENTIN   07          *INITIALIZE TEST.
08 JSR @ENTIN   08          *INITIALIZE TEST.
09 RAND         09          *INITIALIZE TEST.
10 JSR @ENTRA   10          *INITIALIZE TEST.
11 STA 0,ISLCO 11          *INITIALIZE TEST.
12 LOBFAD      12          *INITIALIZE TEST.
13 RNDADR LOWBF,LOBFU 13          *INITIALIZE TEST.
14 RAND         14          *INITIALIZE TEST.
15 JSR @ENTRA   15          *INITIALIZE TEST.
16 JSR @RNADR   16          *INITIALIZE TEST.
17 LOBF        17          *INITIALIZE TEST.
18 LOBFU       18          *INITIALIZE TEST.
19 STA 0,ISLCO 19          *INITIALIZE TEST.
20 STA 1,ISLCO 20          *INITIALIZE TEST.
21 STA 1,@ISLCO 21          *INITIALIZE TEST.
22 HIBFAD      22          *INITIALIZE TEST.
23 RNDADR HIGBF,HIBFU 23          *INITIALIZE TEST.
24 RAND         24          *INITIALIZE TEST.
25 JSR @ENTRA   25          *INITIALIZE TEST.
26 JSR @RNADR   26          *INITIALIZE TEST.
27 HIGBF        27          *INITIALIZE TEST.
28 HIBFU        28          *INITIALIZE TEST.
29 MOV 0,2     29          *INITIALIZE TEST.
30 LDA 1,ISLCO 30          *INITIALIZE TEST.
31 STA 1,0,2   31          *INITIALIZE TEST.
32 MOVZR 0,3,SZC 32          *INITIALIZE TEST.
33 ADI 1,3     33          *INITIALIZE TEST.
34 HLV 0       34          *INITIALIZE TEST.
35 -NGLOC     35          *INITIALIZE TEST.
36 ADDR      36          *INITIALIZE TEST.
37 STA 0,+2   37          *INITIALIZE TEST.
38 EDSZ      38          *INITIALIZE TEST.
39 100000    39          *INITIALIZE TEST.
40 MOV 0,0    40          *INITIALIZE TEST.
41 LDA 0,ISLCO 41          *INITIALIZE TEST.
42 SBI 1,0    42          *INITIALIZE TEST.
43 LOA 1,@ISLCO 43          *INITIALIZE TEST.
44 SUB# 0,1,SZR 44          *INITIALIZE TEST.
45 ERROR      45          *INITIALIZE TEST.
46 LOOP      46          *INITIALIZE TEST.
47 JSR @ENTLO 47          *INITIALIZE TEST.
48 006305   48          *INITIALIZE TEST.
10064 ECL32      01 000000 08ZD00: ISZSN=0      EDSZ,DZ00,0,SBI,DZD00
02 EISZ4          02          EDSZ,DZ00,0,SBI,DZD00
03 TSTLOC         03          TSTLOC
04 JSR *+1       04          *(ADDR OF TEST+1) IS
05 STA 3,HELP   05          *SAVED IN LOC. HELP
06 SETUP 100.   06          *INITIALIZE TEST.
07 JSR @ENTIN   07          *INITIALIZE TEST.
08 JSR @ENTIN   08          *INITIALIZE TEST.
09 RAND         09          *INITIALIZE TEST.
10 JSR @ENTRA   10          *INITIALIZE TEST.
11 STA 0,ISLCO 11          *INITIALIZE TEST.
12 LOBFAD      12          *INITIALIZE TEST.
13 RNDADR LOWBF,LOBFU 13          *INITIALIZE TEST.
14 RAND         14          *INITIALIZE TEST.
15 JSR @ENTRA   15          *INITIALIZE TEST.
16 JSR @RNADR   16          *INITIALIZE TEST.
17 LOBF        17          *INITIALIZE TEST.
18 LOBFU       18          *INITIALIZE TEST.
19 STA 0,ISLCO 19          *INITIALIZE TEST.
20 STA 1,ISLCO 20          *INITIALIZE TEST.
21 STA 1,@ISLCO 21          *INITIALIZE TEST.
22 HIBFAD      22          *INITIALIZE TEST.
23 RNDADR HIGBF,HIBFU 23          *INITIALIZE TEST.
24 RAND         24          *INITIALIZE TEST.
25 JSR @ENTRA   25          *INITIALIZE TEST.
26 JSR @RNADR   26          *INITIALIZE TEST.
27 HIGBF        27          *INITIALIZE TEST.
28 HIBFU        28          *INITIALIZE TEST.
29 MOV 0,2     29          *INITIALIZE TEST.
30 LDA 1,ISLCO 30          *INITIALIZE TEST.
31 STA 1,0,2   31          *INITIALIZE TEST.
32 MOVZR 0,3,SZC 32          *INITIALIZE TEST.
33 ADI 1,3     33          *INITIALIZE TEST.
34 HLV 0       34          *INITIALIZE TEST.
35 -NGLOC     35          *INITIALIZE TEST.
36 ADDR      36          *INITIALIZE TEST.
37 STA 0,+2   37          *INITIALIZE TEST.
38 EDSZ      38          *INITIALIZE TEST.
39 100000    39          *INITIALIZE TEST.
40 MOV 0,0    40          *INITIALIZE TEST.
41 LDA 0,ISLCO 41          *INITIALIZE TEST.
42 SBI 1,0    42          *INITIALIZE TEST.
43 LOA 1,@ISLCO 43          *INITIALIZE TEST.
44 SUB# 0,1,SZR 44          *INITIALIZE TEST.
45 ERROR      45          *INITIALIZE TEST.
46 LOOP      46          *INITIALIZE TEST.
47 JSR @ENTLO 47          *INITIALIZE TEST.
48 006305   48          *INITIALIZE TEST.

```



```

10065 ECL32
01 000001 08ZD1:
02 EISZ4 EDZS,DZD1,1,98I,DZD10
03 TSILC
04 JSR @ENTRA ;(ADDR OF TEST+1) IS
05 STA 3,HELP ;(SAVED IN LOC. HELP
06 SETUP 100. ;INITIALIZE TEST.
07 JSR @ENTIN
08 RAND
09 JSR @ENTRA ;C(AC0)=RANDOM #
10 STA 0,ISLC0 ;C(ISLC0)=RANDOM DATA
11 LOBFAD LOWBF,LOBFU
12 RAND
13 JSR @ENTRA ;C(AC0)=RANDOM #
14 JSR @RNADR ;GET ADDRESS IN THE RANGE
15 LOBF ;C(LOWBF) AND C(HIBFU)
16 STA 0,ISLC1 ;C(ISLC1)=RNDM ADDR 1
17 LDA 1,ISLC0 ;STORE RNDM DATA INTO
18 STA 1,@ISLC1 ;RNDM ADDR 1
19 HIBFAD HIGBF,HIBFU
20 RNDADR
21 RAND
22 JSR @ENTRA ;C(AC0)=RANDOM #
23 JSR @RNADR ;GET ADDRESS IN THE RANGE
24 HIGBF ;C(HIGBF) AND C(HIBFU)
25 STA 0,ISLC2 ;C(ISLC2)=RNDM ADDR 2
26 LDA 1,ISLC1 ;STORE RNDM ADDR 1 IN
27 STA 1,@ISLC2 ;RNDM ADDR 2
28 HIBFAD HIGBF,HIBFU
29 RNDADR
30 RAND
31 JSR @ENTRA ;C(AC0)=RANDOM #
32 JSR @RNADR ;GET ADDRESS IN THE RANGE
33 HIGBF ;C(HIGBF) AND C(HIBFU)
34 LDA 1,ISLC2 ;MAKE SURE THAT RNDM ADDR
35 SUB# 0,1,SNR ;2 AND 3 ARE DIFFERENT
36 JMP DZD10 ;
37 MOV 0,2 ;RAND SAVE @RNDM ADDR 2
38 ADDR 1,1 ;IN RNDM ADDR 3
39 STA 1,0,2 ;
40 ELEM 1,0ZD1-..1 ;
41 SUB 0 ;AC0=(RNDM ADDR 3-PC OF
42 ADDR 0 ;EDZS+1)
43 STA 0,0 ;ADD INDIRECT BIT TO AC0
44 EDSZ @0,1 ;RAND STORE IN EDSZ
45 ;EXECUTE EDSZ INSTRUCTION
46 ;
47 MOV 0,0 ;
48 LDA 0,ISLC0 ;C(RNDM ADDR 1) MUST BE
49 SBI 1,0 ;=(RNDM DATA+1) FOR EISZ
50 LDA 1,@ISLC1 ;RAND =(RNDM DATA-1) FOR EDSZ
51 SUB# 0,1,SNR ;
52 ERROR ;
53 LOOP ;
54 JSR @ENTLO ;ITERATE TEST ROUTINE
55

```

```

:0066 ECL32
01 000002 08ZD2:
02 EISZ4 EDZS,DZD2,2,98I,DZD20
03 TSILC
04 JSR @ENTRA ;(ADDR OF TEST+1) IS
05 STA 3,HELP ;(SAVED IN LOC. HELP
06 SETUP 100. ;INITIALIZE TEST.
07 JSR @ENTIN
08 RAND
09 JSR @ENTRA ;C(AC0)=RANDOM #
10 STA 0,ISLC0 ;C(ISLC0)=RANDOM DATA
11 LOBFAD LOWBF,LOBFU
12 RAND
13 JSR @ENTRA ;C(AC0)=RANDOM #
14 JSR @RNADR ;GET ADDRESS IN THE RANGE
15 LOBF ;C(LOWBF) AND C(HIBFU)
16 STA 0,ISLC1 ;C(ISLC1)=RNDM ADDR 1
17 LDA 1,ISLC0 ;STORE RNDM DATA INTO
18 STA 1,@ISLC1 ;RNDM ADDR 1
19 HIBFAD HIGBF,HIBFU
20 RNDADR
21 RAND
22 JSR @ENTRA ;C(AC0)=RANDOM #
23 JSR @RNADR ;GET ADDRESS IN THE RANGE
24 HIGBF ;C(HIGBF) AND C(HIBFU)
25 STA 0,ISLC2 ;C(ISLC2)=RNDM ADDR 2
26 LDA 1,ISLC1 ;STORE RNDM ADDR 1 IN
27 STA 1,@ISLC2 ;RNDM ADDR 2
28 HIBFAD HIGBF,HIBFU
29 RNDADR
30 RAND
31 JSR @ENTRA ;C(AC0)=RANDOM #
32 JSR @RNADR ;GET ADDRESS IN THE RANGE
33 HIGBF ;C(HIGBF) AND C(HIBFU)
34 LDA 1,ISLC2 ;MAKE SURE THAT RNDM ADDR
35 SUB# 0,1,SNR ;2 AND 3 ARE DIFFERENT
36 JMP DZD20 ;
37 MOV 0,2 ;RAND SAVE @RNDM ADDR 2
38 ADDR 1,1 ;IN RNDM ADDR 3
39 STA 1,0,2 ;
40 ELEM 0,2,SZC ;(RNDM ADDR 3/2)+BIT 15 IS
41 ADI 1,2 ;STORED IN AC2 AND
42 HLV 0 ;AC0=(RNDM ADDR 3/2)
43 ADDR 0 ;ADD INDIRECT BIT TO AC0
44 ADDR 0,0 ;RAND STORE IN EDSZ
45 STA @0,2 ;EXECUTE EDSZ INSTRUCTION
46 ;
47 MOV 0,0 ;
48 LDA 0,ISLC0 ;C(RNDM ADDR 1) MUST BE
49 SBI 1,0 ;=(RNDM DATA+1) FOR EISZ
50 LDA 1,@ISLC1 ;RAND =(RNDM DATA-1) FOR EDSZ
51 SUB# 0,1,SNR ;
52 ERROR ;
53 LOOP ;
54 JSR @ENTLO ;ITERATE TEST ROUTINE
55

```

```

01 000003 08ZD3:
02 EISZ4 EDZS,DZD3,3,98I,DZD30
03 TSILC
04 JSR @ENTRA ;(ADDR OF TEST+1) IS
05 STA 3,HELP ;(SAVED IN LOC. HELP
06 SETUP 100. ;INITIALIZE TEST.
07 JSR @ENTIN
08 RAND
09 JSR @ENTRA ;C(AC0)=RANDOM #
10 STA 0,ISLC0 ;C(ISLC0)=RANDOM DATA
11 LOBFAD LOWBF,LOBFU
12 RAND
13 JSR @ENTRA ;C(AC0)=RANDOM #
14 JSR @RNADR ;GET ADDRESS IN THE RANGE
15 LOBF ;C(LOWBF) AND C(HIBFU)
16 STA 0,ISLC1 ;C(ISLC1)=RNDM ADDR 1
17 LDA 1,ISLC0 ;STORE RNDM DATA INTO
18 STA 1,@ISLC1 ;RNDM ADDR 1
19 HIBFAD HIGBF,HIBFU
20 RNDADR
21 RAND
22 JSR @ENTRA ;C(AC0)=RANDOM #
23 JSR @RNADR ;GET ADDRESS IN THE RANGE
24 HIGBF ;C(HIGBF) AND C(HIBFU)
25 STA 0,ISLC2 ;C(ISLC2)=RNDM ADDR 2
26 LDA 1,ISLC1 ;STORE RNDM ADDR 1 IN
27 STA 1,@ISLC2 ;RNDM ADDR 2
28 HIBFAD HIGBF,HIBFU
29 RNDADR
30 RAND
31 JSR @ENTRA ;C(AC0)=RANDOM #
32 JSR @RNADR ;GET ADDRESS IN THE RANGE
33 HIGBF ;C(HIGBF) AND C(HIBFU)
34 LDA 1,ISLC2 ;MAKE SURE THAT RNDM ADDR
35 SUB# 0,1,SNR ;2 AND 3 ARE DIFFERENT
36 JMP DZD30 ;
37 MOV 0,2 ;RAND SAVE @RNDM ADDR 2
38 ADDR 1,1 ;IN RNDM ADDR 3
39 STA 1,0,2 ;
40 ELEM 0,2,SZC ;(RNDM ADDR 3/2)+BIT 15 IS
41 ADI 1,2 ;STORED IN AC2 AND
42 HLV 0 ;AC0=(RNDM ADDR 3/2)
43 ADDR 0 ;ADD INDIRECT BIT TO AC0
44 ADDR 0,0 ;RAND STORE IN EDSZ
45 STA @0,2 ;EXECUTE EDSZ INSTRUCTION
46 ;
47 MOV 0,0 ;
48 LDA 0,ISLC0 ;C(RNDM ADDR 1) MUST BE
49 SBI 1,0 ;=(RNDM DATA+1) FOR EISZ
50 LDA 1,@ISLC1 ;RAND =(RNDM DATA-1) FOR EDSZ
51 SUB# 0,1,SNR ;
52 ERROR ;
53 LOOP ;
54 JSR @ENTLO ;ITERATE TEST ROUTINE
55

```

0067 ECL32

```

01 000003 0SZD3: IZSW#3 EDZ, DZ03, 3, SBI, DZ03
02 01 TSTLOC
03 JSR *+1 ;(ADDR OF TEST+1) IS
04 03620 00401 STA 3,HELP ;SAVED IN LOC. HELP
05 03621 054216 SETUP 100. ;INITIALIZE TEST.
06 07 03622 06304 JSR @ENTIN
07 03623 000144 RAND
08 03624 006307 JSR @ENTRA
09 03625 040246 STA 0, ISLCO
10 LOBFAD LOMBF, LOBFU
11 RNDADR LOMBF, LOBFU
12 RAND
13
14
15 03626 006307 ;C(AC0)=RANDOM #
16 03627 006310 JSR @ENTRA
17 03630 000314 JSR @RNADR ;GET ADDRESS IN THE RANGE
18 03631 000311 LOMBF ;C(LOWBF) AND C(HIBFU)
19 03632 040247 STA 0, ISLCO
20 03633 024246 LDA 1, ISLCO ;STORE RNDM DATA INTO
21 03634 046247 STA 1, @ISLCO ;RNDM ADDR 1
22 HIBFAD HIBGF, HIBFU
23 RNDADR HIBGF, HIBFU
24 RAND
25 03635 006307 JSR @ENTRA
26 03636 006310 JSR @RNADR ;GET ADDRESS IN THE RANGE
27 03637 000312 HIBGF ;C(HIGBF) AND C(HIBFU)
28 03640 000315 HIBFU
29 03641 040250 STA 0, ISLCO
30 03642 024247 LDA 1, ISLCO ;STORE RNDM ADDR 1 IN
31 03643 046250 STA 1, @ISLCO ;RNDM ADDR 2
32
33
34
35 03644 006307 RAND
36 03645 006310 JSR @ENTRA
37 03646 000312 HIBGF ;C(AC0)=RANDOM #
38 03647 000315 HIBFU ;GET ADDRESS IN THE RANGE
39 03650 024250 LDA 1, ISLCO ;C(HIGBF) AND C(HIBFU)
40 03651 108415 SUB# 0, 1, SBR ;MAKE SURE THAT RNDM ADDR
41 03652 000772 JMP DZD30 ;2 RND 3 ARE DIFFERENT
42 03653 111000 MOV 0, 2 ;AND SAVE @RNDM ADDR 2
43 03654 127240 ADDR 1, 1 ;IN RNDM ADDR 3
44 03655 045000 STA 1, 0, 2 ;
45 03656 115222 MOVR 0, 3, SZC ;(RNDM ADDR 3/2)+BIT 15 IS
46 03657 114010 ADI 1, 3 ;STORED IN AC3 AND
47 03660 143370 MIV 0 ;AC0=(RNDM ADDR 3/2)
48 ;NOLCO
49 03661 103240 ADDR 0, 0 ;ADD INDIRECT BIT TO AC0
50 03662 040402 STA 0, *2 ;AND STORE IN EDZ
51 03663 117470 DZ03: EDZ @0, 3 ;EXECUTE EDZ INSTRUCTION
52
53 03665 101000 MOV 0, 0 ;
54 03666 020246 LDA 0, ISLCO ;C(RNDM ADDR 1) MUST BE
55 03667 100110 SBI 1, 0 ;=(RNDM DATA+1) FOR EISZ
56 03670 026247 LDA 1, @ISLCO ;AND =(RNDM DATA-1) FOR EDZ
57 03671 106414 SUB# 0, 1, SZR ;
58 ERROR
59 LOOP
60 03676 006305 JSR @ENTLO ;ITERATE TEST ROUTINE

```

0068 ECL32

```

01 000003 0SZD3: IZSW#3 EDZ, DZ03, 3, SBI, DZ03
02 01 TSTLOC
03 JSR *+1 ;(ADDR OF TEST+1) IS
04 03620 00401 STA 3,HELP ;SAVED IN LOC. HELP
05 03621 054216 SETUP 100. ;INITIALIZE TEST.
06 07 03622 06304 JSR @ENTIN
07 03623 000144 RAND
08 03624 006307 JSR @ENTRA
09 03625 040246 STA 0, ISLCO
10 LOBFAD LOMBF, LOBFU
11 RNDADR LOMBF, LOBFU
12 RAND
13
14
15 03705 006307 ;C(AC0)=RANDOM #
16 03706 040421 JSR @ENTRA
17 03707 112470 DZE00: EISZ 0, DZE0 ;GET ADDRESS IN THE RANGE
18 ;C(LOWBF) AND C(HIBFU)
19 03711 101000 STA 0, 0 ;C(AC0)=RANDOM #
20 03712 116470 EDZ DZE0-*, 1 ;AFTER EXECUTING
21 000015 ;EISZ AND EDZ
22 03714 101000 MOV 0, 0 ;C(DZE0) SHOULD BE SAME
23 03715 014246 DSZ ISLCO ;
24 03716 000771 JMP DZE00 ;
25 03717 024410 LOA 1, DZE0 ;
26 03720 106414 SUB# 0, 1, SZR ;
27 ERROR ;
28 LOOP ;
29 03725 006305 JSR @ENTLO ;ITERATE TEST ROUTINE
30 03726 101001 MOV 0, 0, SRP ;
31 03727 000000 DZE0: 0 ;
32

```

```

10069 ECL32
01
02
03
04
05
06
07
08 03730 00401
09 03731 054216
10 03732 006313
11 03733 17777
12
13
14
15 03734 006304
16 03735 000144
17 03736 020314
18 03737 024311
19 03740 106400
20 03741 046261
21
22
23 03742 006307
24 03743 006310
25 03744 000260
26 03745 000261
27 03746 120110
28 03747 040262
29
30 03750 006307
31 03751 030314
32 03752 041000
33 03753 024262
34 03754 107000
35 03755 104110
36 03756 045001
37 03757 143370
38 03760 132370
39 03761 000770
40
41 03762 006307
42 03763 024314
43 03764 106570
44 03765 000775
45 03766 130010
46
47 03767 040251
48
49 03770 006307
50 03771 114400
51 03772 105040
52 03773 020251
53 03774 143170
54
55 03776 101003
56
57 04003 137014
58
59
60 04010 006305

; CHECKING OUT "OSPA" INSTRUCTION
;
OSPA0:
EDSP1 LOMB.L0BFU,17777
TSTLOC
JSR *+1
STA 3,HELP
FILL 17777
JSR @JFILL
17777
SETUP 100.
JSR @ENTIN
100.
LDA 0,LOWBF
LDA 1,LOBFU
SUB 0,1
STA 1,MXTBL
RANDADR D3,MXTBL
RAND
JSR @ENTRA
JSR @RNADR
D3
MXTBL
SBI 2,0
STA 0,TBLSIZ
RAND
JSR @ENTRA
LDA 2,LOWBF
STA 0,0,2
LDA 1,TBLSIZ
ADD 0,1
SBI 1,1
STA 1,1,2
HLV 0
CLM 1,2
JMP *-8.
RAND
JSR @ENTRA
LDA 1,LOWBF
CLM 0,1
JMP *-5
ADI 2,2
STA 0,DSLCO
RAND
JSR @ENTRA
NEG 0,3
MOV0 0,1
LDA 0,DSLCO
DSPA 0,0,2
MOV 0,0,SNC
ERROR
ADD# 1,3,SZR
ERROR
LOOP
JSR @ENTLO
ITERATE TEST ROUTINE

```

10071 ECL32

```

01
02
03
04 04011 004401
05 04012 054216
06 04013 006313
07 04014 177777
08
09
10
11 04015 006304
12 04016 000144
13 04017 020312
14 04020 024315
15 04021 106400
16 04022 044261
17
18
19 04023 006307
20 04024 006310
21 04025 000260
22 04026 000261
23 04027 120110
24 04030 040262
25
26 04031 006307
27 04032 030312
28 04033 041000
29 04034 024262
30 04035 107000
31 04036 104110
32 04037 045001
33 04040 143370
34 04041 132370
35 04042 000770
36
37 04043 006307
38 04044 024312
39 04045 106370
40 04046 000775
41 04047 130010
42
43 04050 040251
44 04051 050252
45
46 04052 006307
47 04053 111020
48 04054 024251
49 04055 034252
50 04056 147570
51
52 04060 101002
53
54 04065 112414
55
56
57 04072 006305
58

```

10072 ECL32

```

01
02
03
04
05
06 04073 004401
07 04074 054216
08
09 04075 006313
10 04076 006306
11
12
13 04077 006304
14 04100 000144
15 04101 020314
16 04102 024311
17 04103 106400
18 04104 044261
19
20
21 04105 006307
22 04106 006310
23 04107 000260
24 04110 000261
25 04111 120110
26 04112 040262
27
28 04113 006307
29 04114 030314
30 04115 041000
31 04116 024262
32 04117 107000
33 04120 104110
34 04121 045001
35 04122 143370
36 04123 132370
37 04124 000770
38
39 04125 006307
40 04126 024314
41 04127 106370
42 04130 000775
43 04131 130010
44
45 04132 050410
46 04133 025376
47 04134 113000
48 04135 132400
49 04136 186470
50 000005
51 04140 045000
52 04141 142170
53
54
55 04147 024013
56 04150 045000
57
58 04151 006305
59

```

```

;USED IN OSPA TESTS
;ENTER
;LOWBF,LOBFU,DSPER
;+1
;+SAVED IN LOC. HELP
;+ADDR OF TEST+1) IS
;+FILL UPPER AND LOWER
;+SCRATCH BUFFER AREA
;+WITH (177777)
;INITIALIZE TEST.
;+MAX. AVAILABLE AREA
;+FOR TABLE IS =
;+C(HIGBF)-C(HIGBF)
;+RANDOM SIZE < C(MXTBL)-2
;+C(AC0)=RANDOM #
;+GET ADDRESS IN THE RANGE
;+C(D3) AND C(MXTBL)
;+TABLE SIZE =
;+1<RANDOM SIZE< C(MXTBL)-2
;+C(AC0)=RANDOM #
;+C(TABLE-2)= RANDOM DATA 1
;+C(TABLE-1)=(RANDOM DATA 1+
;+TABLE SIZE-1)
;+ACTUAL TABLE BEGIN ADDR
;+IS AT C(LOWBF)
;+MAKE SURE C(TABLE-2) <
;+C(TABLE-1)
;+C(AC0)=RANDOM #
;+GET RANDOM DATA 2 IN
;+THE RANGE OF C(TABLE-2)
;+AND C(TABLE-1)
;+AC2=TABLE BEGIN ADDR
;+STORE TABLE ADDR IN
;+OSPA INSTRUCTION
;+SUB
;+ELEM
;+IN DISPATCH LOCATION
;+EXECUTE OSPA, INDEX=00
;+DSPA
;+ERROR
;+STA
;+LOOP
;+ITERATE TEST ROUTINE

```



```

0074 ECL32
01
02 04236 006305
03
LOOP JSR @ENTLO
:ITERATE TEST ROUTINE

10073 ECL32
01
02 04236 006305
03
LOOP JSR @ENTLO
:ITERATE TEST ROUTINE

01
02
03
04 04152 004401 JSR *1 ;(ADDR OF TEST+1) IS
05 04153 054216 STA 3,HELP ;SAVED IN LOC. HELP
06 04154 006313 FILL DSPER ;FILL UPPER AND LOWER
07 04155 006306 @JFILL JSR ;SCRATCH BUFFER AREA
08 04156 006304 ;WITH (DSPER)
09
10
11 04157 000144 SETUP 100.
12 04158 000144 JSR @ENTIN
13 04159 000144 LDA 100.
14 04160 020312 LDA 0,HIGBF
15 04161 024315 LDA 1,HIRFU
16 04162 106400 SUB 0,1
17 04163 044261 STA 1,MXTBL
18 04164 006307 RNDADR D3,MXTBL
19 04165 006310 RAND
20 04166 006310 JSR @ENTRA
21 04167 000260 JSR @RNDADR
22 04168 000261 D3
23 04169 120110 MXTBL
24 04170 040262 SBI 2,0
25 04171 040262 STA 0,TBLSIZ
26 04172 006307 RAND
27 04173 030312 JSR @ENTRA
28 04174 041000 LDA 2,HIGBF
29 04175 024262 STA 0,0,2
30 04176 107000 LDA 1,TBLSIZ
31 04177 104110 ADD 0,1
32 04200 045001 SBI 1,1
33 04201 143370 STA 1,1,2
34 04202 132370 HLV 0
35 04203 000770 CLM 1,2
36 04204 006307 JMP --8.
37 04205 024312 RAND
38 04206 106370 JSR @ENTRA
39 04207 000775 LDA 1,HIGBF
40 04210 130010 CLM 0,1
41 04211 050251 JMP *3
42 04212 166470 ADI 2,2
43 04213 050251 STA 2,DSLCO
44 04214 166470 ELEM 1,DPB1*,.1
45 04215 050412 SUB 1,2
46 04216 030251 STA 2,DPB1+1
47 04217 025376 LDA 2,DSLCO
48 04218 132400 LDA 1,-2,2
49 04219 130000 ADD 0,2
50 04220 132400 SUB 1,2
51 04221 166470 ELEM 1,DPB1*3*-.1,1
52 04222 000006 STA 1,0,2
53 04223 045000 ;IN DISPATCH LOCATION
54 04224 115000 MOV 0,3
55 04225 156570 DSPA 3,0,1
56 04226 000000 ;EXECUTE DSPA,INDEX=01
57
58
59 04234 024013- ERROR
60 04235 045000 LDA 1,=DSPER
:TO JSR @EMTER'

```

```

10075 ECL32
01
02
03
04 04237 004401
05 04240 054216
06
07 04241 006313
08 04242 006306
09
10
11 04243 006304
12 04244 000144
13 04245 020314
14 04246 024311
15 04247 106400
16 04250 044261
17
18
19 04251 006307
20 04252 006310
21 04253 000260
22 04254 000261
23 04255 120110
24 04256 040262
25
26 04257 006307
27 04260 030314
28 04261 041000
29 04262 024262
30 04263 107000
31 04264 104110
32 04265 045001
33 04266 143370
34 04267 132370
35 04270 000770
36
37 04271 006307
38 04272 024314
39 04273 106370
40 04274 000775
41 04275 130010
42
43 04276 050251
44 04277 023376
45 04300 113000
46 04301 132000
47 04302 050252
48 04303 166470
49 000013
50 04305 045000
51 04306 030251
52 04307 145000
53 04310 151222
54 04311 110010
55 04312 147370
56 04313 044402
57 04314 143170
58 000000
59
60 04322 034013-

```

```

OSPR2:
EOSP1
TSTLDC
STA 3,HELP
FILL DSPER
JSR @FILL
DSPER
SETUP 100.
JSR @ENTIN
100.
LOA 0,LOWBF
FOR TABLE IS =
SUB 0,1
STA 1,MXTBL
R@NADR 03,MXTBL
RAND
JSR @ENTRA
@R@NADR
D3
MXTBL
S@I 2,0
STA 0,T@LSIZ
RAND
JSR @ENTRA
LOA 2,LOWRF
STA 0,0,2
LOA 1,T@LSIZ
ADD 0,1
S@I 1,1
STA 1,1,2
HLV 0
CLM 1,2
JMP -8.
RAND
JSR @ENTRA
LOA 1,LOWRF
CLM 0,1
JMP -3
ADI 2,2
STA 2,0SLCO
LOA 1,-2,2
ADD 0,2
STA 1,2
STA 2,0SLC1
ELEF 1,0PR2*3,-1,1
STA 1,0,2
LOA 2,0SLCO
MOV 2,1
MOVZR 2,2,SZC
ADI 1,2
HLV 1
STA 1,0PR2*1
OSPA 0,0,2
ERROR
LOA 3,EDSPER

```

```

L0076 ECL32
01 04323 030232
02 04324 055000
03
04 04325 006305

```

```

LDA 2,0SLC1
STA 3,0,2
JSR @ENTLO

```

```

;TO 'JSR @ENTER'
;
;ITERATE TEST ROUTINE

```

```

;(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(03) AND C(MXTBL)
;TABLE SIZE =
;1<RANDOM SIZE< C(MXTBL)-2
;C(AC0)=RANDOM #
;C(TABLE-2)= RANDOM DATA 1
;C(TABLE-1)=(RANDOM DATA 1+
;TABLE SIZE-1)
;ACTUAL TABLE BEGIN ADDR
;IS AT C(LOWRF)
;MAKE SURE C(TABLE-2) <
;C(TABLE-1)
;
;C(AC0)=RANDOM #
;GET RANDOM DATA 2 IN
;THE RANGE OF C(TABLE-2)
;AND C(TABLE-1)
;AC0=DISPATCH ADDR
;STORE ADDR (0PR2+3) IN
;DISPATCH LOCATION
;
;AC2=(TABLE ADDR/2)+
;BIT 15
;ALSO STORE (TABLE ADDR/2)
;IN OSPA INSTRUCTION
;EXECUTE DSPA,INDEX=02
;RESTORE DISPATCH LOCATION

```

```

10077 ECL32
01
02
03
04 04326 004601
05 04327 054216
06
07 04330 006313
08 04331 006306
09
10
11 04332 006304
12 04333 001144
13 04334 020312
14 04335 024315
15 04336 106400
16 04337 042261
17
18
19 04340 006307
20 04341 006310
21 04342 000260
22 04343 000261
23 04344 120110
24 04345 040262
25
26 04346 006307
27 04347 030312
28 04350 041000
29 04351 024262
30 04352 107000
31 04353 104110
32 04354 045001
33 04355 143370
34 04356 132370
35 04357 000770
36
37 04360 006307
38 04361 024312
39 04362 106370
40 04363 000775
41 04364 130010
42
43 04365 050251
44 04366 025376
45 04367 113000
46 04370 132400
47 04371 050252
48 04372 166470
49 04373 000012
50 04374 045000
51 04375 030251
52 04376 155222
53 04377 114010
54 04400 153370
55 04401 050402
56 04402 143570
57
58
59 04410 024013-
60 04411 030252

```

```

0078 ECL32
01 04412 045000
02
03 04413 006305
04

```

```

STA 1,0,2
LOOP
JSR @ENTLO

```

```

;JSR @ENTER
;ITERATE TEST ROUTINE

```

```

DSP83:
EDSP1
HIGBF,HIBFU,DSPER

```

```

TSTLOC
JSR
+1
STA 3,HELP
OSPER
FILL
JSR 5JFILL
OSPER

```

```

SETUP 100.
JSR @ENTIN
100.
LDA 0,HIGBF
LDA 1,HIBFU
LDA 0,1
STA 1,MXTBL
RNDADR D3,MXTBL
RAND

```

```

JSR @ENTRA
@RNADR
D3
MXTBL
SBI 2,0
STA 0,TBLSIZ
RAND

```

```

JSR @ENTRA
LDA 2,HIGBF
STA 0,0,2
LDA 1,TBLSIZ
ADD 0,1
SBI 1,1
STA 1,1,2
HLV 0
CLM 1,2
JMP *-8.

```

```

RAND
JSR @ENTRA
LDA 1,HIGBF
CLM 0,1
JMP *-3
ADI 2,2
STA 2,DSLCO
LDA 1,-2,2
ADD 0,2
SUB 1,2
STA 2,DSLCL1
ELEF 1,DPB3+3*,-1,1

```

```

STA 1,0,2
LDA 2,DSLCO
MOVZR 2,3,8ZC
ADI 1,3
HLV 2
STA 2,DPB3+1
DSPA 0,0,3
ERROR
LDA 1,-DSPER
LDA 2,DSLCL1

```

```

;C(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(D3) AND C(MXTBL)
;TABLE SIZE =
;1-RANDOM SIZE< C(MXTBL)-2
;C(AC0)=RANDOM #
;C(TABLE-2)=RANDOM DATA 1
;C(TABLE-1)=(RANDOM DATA 1+
;TABLE SIZE-1)
;ACTUAL TABLE BEGIN ADDR
;IS AT C(HIGBF)
;MAKE SURE C(TABLE-2) <
;C(TABLE-1)

```

```

;C(AC0)=RANDOM #
;GET RANDOM DATA 2 IN
;THE RANGE OF C(TABLE-2)
;AND C(TABLE-1)
;AC0=DISPATCH ADDEND
;AC2=TABLE BEGIN ADDR
;STORE ADDR (DPB3+3) IN
;DISPATCH LOCATION
;BYT 15
;AC3=(TABLE ADDR/2)+
;ALSO STORE (TABLE ADDR/2)
;IN DSPA INSTRUCTION
;EXECUTE DSPA,INDEX=03
;RESTORE DISPATCH
;LOCATION TO

```

10079 FCL32

10080 ECL32

|    |                      |                   |                      |                   |
|----|----------------------|-------------------|----------------------|-------------------|
| 01 | EDSPI                | HIGBF,MHBFU,OSPER | EDSPI                | HIGBF,MHBFU,OSPER |
| 02 | TSLOC                |                   | TSLOC                |                   |
| 03 | JSR                  | *+1               | JSR                  | *+1               |
| 04 | STA                  | 3,HELP            | STA                  | 3,HELP            |
| 05 | FILL                 | OSPER             | FILL                 | OSPER             |
| 06 | JSR                  | @JFILL            | JSR                  | @JFILL            |
| 07 | OSPER                |                   | OSPER                |                   |
| 08 | SETUP                | 100.              | SETUP                | 100.              |
| 09 | JSR                  | @ENTN             | JSR                  | @ENTN             |
| 10 | 100.                 |                   | 100.                 |                   |
| 11 | LDA                  | 0,HIGBF           | LDA                  | 0,HIGBF           |
| 12 | LDA                  | 1,MHBFU           | LDA                  | 1,MHBFU           |
| 13 | LDA                  | 0,1               | LDA                  | 0,1               |
| 14 | SUB                  | 1,MXTBL           | SUB                  | 1,MXTBL           |
| 15 | STA                  | 1,MXTBL           | STA                  | 1,MXTBL           |
| 16 | RNDADR               | D3,MXTBL          | RNDADR               | D3,MXTBL          |
| 17 | RAND                 |                   | RAND                 |                   |
| 18 | JSR                  | @ENTN             | JSR                  | @ENTN             |
| 19 | D3                   | @RNDADR           | D3                   | @RNDADR           |
| 20 | MXTBL                |                   | MXTBL                |                   |
| 21 | SBI                  | 2,0               | SBI                  | 2,0               |
| 22 | STA                  | 0,TBLSIZ          | STA                  | 0,TBLSIZ          |
| 23 | RAND                 |                   | RAND                 |                   |
| 24 | JSR                  | @ENTN             | JSR                  | @ENTN             |
| 25 | LDA                  | 2,HIGBF           | LDA                  | 2,HIGBF           |
| 26 | STA                  | 0,0,2             | STA                  | 0,0,2             |
| 27 | LDA                  | 1,TBLSIZ          | LDA                  | 1,TBLSIZ          |
| 28 | ADD                  | 0,1               | ADD                  | 0,1               |
| 29 | SBI                  | 1,1               | SBI                  | 1,1               |
| 30 | STA                  | 1,1,2             | STA                  | 1,1,2             |
| 31 | HLV                  | 0                 | HLV                  | 0                 |
| 32 | CLM                  | 1,2               | CLM                  | 1,2               |
| 33 | JMP                  | *-8.              | JMP                  | *-8.              |
| 34 | JSR                  | @ENTN             | JSR                  | @ENTN             |
| 35 | LDA                  | 1,HIGBF           | LDA                  | 1,HIGBF           |
| 36 | CLM                  | 0,1               | CLM                  | 0,1               |
| 37 | JMP                  | *-3               | JMP                  | *-3               |
| 38 | ADI                  | 2,2               | ADI                  | 2,2               |
| 39 | STA                  | 2,DPB4+1          | STA                  | 2,DPB4+1          |
| 40 | LDA                  | 1,-2,2            | LDA                  | 1,-2,2            |
| 41 | ADD                  | 1,2               | ADD                  | 1,2               |
| 42 | SUB                  | 1,2               | SUB                  | 1,2               |
| 43 | ELEF                 | 1,DPB4+3,-1,1     | ELEF                 | 1,DPB4+3,-1,1     |
| 44 | STA                  | 1,0,2             | STA                  | 1,0,2             |
| 45 | MOV                  | 0,3               | MOV                  | 0,3               |
| 46 | OSPA                 | 3,0,0             | OSPA                 | 3,0,0             |
| 47 | ERROR                |                   | ERROR                |                   |
| 48 | LDA                  | 1,-DSPER          | LDA                  | 1,-DSPER          |
| 49 | STA                  | 1,0,2             | STA                  | 1,0,2             |
| 50 | LOOP                 |                   | LOOP                 |                   |
| 51 | JSR                  | @ENTN             | JSR                  | @ENTN             |
| 52 | ITERATE TEST ROUTINE |                   | ITERATE TEST ROUTINE |                   |
| 53 | RESTORE DISPATCH LOC |                   | RESTORE DISPATCH LOC |                   |
| 54 | TO 'JSR @ENTER'      |                   | TO 'JSR @ENTER'      |                   |
| 55 | OSPA                 | 3,0,0             | OSPA                 | 3,0,0             |
| 56 | STA                  | 1,0,2             | STA                  | 1,0,2             |
| 57 | STA                  | 1,0,2             | STA                  | 1,0,2             |
| 58 | STA                  | 1,0,2             | STA                  | 1,0,2             |
| 59 | STA                  | 1,0,2             | STA                  | 1,0,2             |
| 60 | LOOP                 |                   | LOOP                 |                   |

0081 ECL32  
01 04557 006305  
02

JSR @ENTLO

ITERATE TEST ROUTINE

10082 ECL32

```
01 02
03
04 04560 00401 JSR @ENTLO
05 04561 054216 STA 3,HELP
06 04562 006313 FILL @JFILL
07 04563 006306 DPER
08
09
10
11 04564 006304 SETUP 100.
12 04565 000144 JSR @ENTIN
13 04566 020312 LDA 0,HIGBF
14 04567 024315 LDA 1,HIRFU
15 04570 106400 SUB 0,1
16 04571 044261 STA 1,MXTBL
17
18
19 04572 006307 RNDADR D3,MXTBL
20 04573 006310 JSR @ENTRA
21 04574 002660 JSR @RNADR
22 04575 002661 MXTBL
23 04576 120110 STA 2,0
24 04577 040262 STA 0,TBL8IZ
25
26 04600 006307 RAND
27 04601 030312 JSR @ENTRA
28 04602 041000 LDA 2,HIGBF
29 04603 024262 STA 0,0.2
30 04604 107000 LDA 1,TBL8IZ
31 04605 104110 ADO 0,1
32 04606 045001 SBI 1,1
33 04607 143370 STA 1,1.2
34 04610 132370 HLV 0
35 04611 000770 CLM 1,2
36
37 04612 006307 JMP @B.
38
39 04614 106370 RAND
40 04615 000775 JSR @ENTRA
41 04616 130010 LDA 1,HIGBF
42
43 04617 050251 CLM 0,1
44 04620 025376 JMP @3
45 04621 113000 ADI 2,2
46 04622 132400 STA 2,0SLC0
47 04623 050252 STA 1,DP86+3--1,1
48 04624 166470 ELEV
49
50 04626 045000 STA 1,0.2
51 04627 030251 LDA 2,0SLC0
52 04630 145000 MOV 2,1
53 04631 151222 MOVZ 2,2,SZC
54 04632 110010 ADI 1,2
55 04633 147370 HLV 1
56 04634 044403 STA 1,DP86+1
57 04635 115000 MOV 0,3
58 04636 157170 DPER
59
60
```

ERROR

0083 ECL32  
01 04644 020013-  
02 04645 030232  
03 04646 041000  
04  
05 04647 006305  
06

LDA 0.=DSPER  
LOA 2,0SLC1  
STA 0,0,2  
LOOP  
JSR @ENTLO

:RESTORE DISPATCH LOC.  
:TO 'JSR @ENTER'  
;

:ITERATE TEST ROUTINE

1:0084 ECL32

01  
02  
03  
04 04650 004401  
05 04651 054216  
06  
07 04652 006313  
08 04653 006306  
09  
10  
11 04654 006304  
12 04655 000144  
13 04656 020314  
14 04657 024311  
15 04660 106400  
16 04661 044261  
17  
18  
19 04662 006307  
20 04663 006310  
21 04664 000260  
22 04665 000261  
23 04666 120110  
24 04667 040262  
25  
26 04670 006307  
27 04671 030314  
28 04672 041000  
29 04673 024262  
30 04674 107000  
31 04675 104110  
32 04676 045001  
33 04677 153370  
34 04700 152370  
35 04701 000770  
36  
37 04702 006307  
38 04703 024314  
39 04704 106370  
40 04705 000775  
41 04706 130010  
42  
43 04707 050251  
44 04710 025376  
45 04711 113000  
46 04712 132400  
47 04713 050252  
48 04714 166470  
49  
50 04716 045000  
51 04717 030251  
52 04720 155222  
53 04721 114010  
54 04722 153370  
55 04723 050403  
56 04724 111000  
57 04725 153570  
58  
59  
60 04733 024013-

EDSP1 LOWBF,LOBFU,DSPER  
TSTLOC  
JSR \*+1  
STA 3,HELP  
FILL DSPER  
JSR @JFILL  
DSPER  
SETUP 100.  
JSR @ENTIN  
100.  
LOA 0,LOWBF  
LDA 1,LOBFU  
SUB 0,1  
STA 1,MXTBL  
RNDADR 0,3,MXTBL  
RAND  
JSR @ENTRA  
JSR @RNADR  
03  
MXTBL  
SBI 2,0  
STA 0,TBLSIZ  
RAND  
JSR @ENTRA  
LOA 2,LOWBF  
STA 0,0,2  
LOA 1,TBLSIZ  
ADD 0,1  
SBI 1,1  
STA 1,1,2  
HLV 0  
CLM 1,2  
JMP \*-8.  
RAND  
JSR @ENTRA  
LOA 1,LOWBF  
CLM 0,1  
JMP \*-3  
ADI 2,2  
STA 2,0SLC0  
LOA 1,-2,2  
ADD 0,2  
SUB 1,2  
STA 2,0SLC1  
ELEV 1,DPB7+3,-1,1  
STA 1,0,2  
LOA 2,0SLC0  
MOVZR 2,3,87C  
ADI 1,3  
HLV 2,DPB7+1  
STA 2,DPB7+1  
MOV 0,2  
DSPA 2,0,3  
ERROR  
LOA 1,-DSPER

:(ADDR OF TEST+1) IS  
:SAVED IN LOC. HELP  
:FILL UPPER AND LOWER  
:SCRATCH BUFFER AREA  
:WITH (DSPER)  
:INITIALIZE TEST.  
:MAX. AVAILABLE AREA  
:FOR TABLE IS  
:(C(LOBFU)-C(LOWBF))  
:(AC0)=RANDOM #  
:GET ADDRESS IN THE RANGE  
:(C(03) AND C(MXTBL))  
:TABLE SIZE =  
:1<RANDOM SIZE< C(MXTBL)-2  
:(AC0)=RANDOM #  
:(C(TABLE-2))= RANDOM DATA 1  
:(C(TABLE-1))=RANDOM DATA 1+  
:TABLE SIZE-1  
:ACTUAL TABLE BEGIN ADDR  
:IS AT C(LOWBF)  
:MAKE SURE C(TABLE-2) <  
:(C(TABLE-1)  
:(AC0)=RANDOM #  
:GET RANDOM DATA 2 IN  
:THE RANGE OF C(TABLE-2)  
:AND C(TABLE-1)  
:AC0=DISPATCH ADDRESS  
:AC2=TABLE BEGIN ADDR  
:SAVE AC2  
:STORE ADDR (DPB7+3) IN  
:DISPATCH LOCATION  
:AC3=(TBL ADDR/2)+  
:811 15  
:AC2=(TBL ADDR/2)  
:STORE AC2 IN DSPA  
:AC2=AC0  
:EXECUTE DSPA  
:RESTORE DISPATCH LOC

```

0085 ECL32
01 04734 030252
02 04735 045000
03
04 04736 006305
05

!0086 ECL32
01
02
03
04
05
06
07 04737 004401
08 04740 054216
09
10 04741 006313
11 04742 006306
12
13
14 04743 006304
15 04744 000144
16 04745 020314
17 04746 024311
18 04747 106400
19 04750 044261
20
21
22 04751 004307
23 04752 004310
24 04753 000260
25 04754 000261
26 04755 120110
27 04756 040262
28
29 04757 004307
30 04760 030314
31 04761 041000
32 04762 024262
33 04763 107000
34 04764 104110
35 04765 045001
36 04766 143370
37 04767 132370
38 04770 000770
39
40 04771 006307
41 04772 024314
42 04773 106370
43 04774 000775
44 04775 130010
45
46
47 04776 040251
48 04777 050252
49 05000 025376
50 05001 113000
51 05002 132400
52 05003 050253
53 05004 166470
54 05005 000827
55 05006 045000
56
57 05007 020312
58 05010 120010
59 05011 040257
60
!TO 'JSR @ENTER'
!
!ITERATE TEST ROUTINE
LDA 2,08LCL1
STA 1,012
LOOP @ENT10
JSR @ENT10
D9SW0=0
D9SP3
D9SW1=0
EDSP1
TSTLOC
JSR *+1
STA 3,HELP
FILL D9SPR
JSR @JFILL
D9SPR
SETUP 100-
JSR @ENTIN
100-
LDA 0,LOWBF
LDA 1,LOWFU
SUB 0,1
STA 1,MXTBL
RNDADR 03,MXTBL
RAND
JSR @ENTRA
JSR @RNDADR
D3
MXTBL
SRI 2,0
STA 0,TBLSIZ
RAND
JSR @ENTRA
LDA 2,LOWBF
STA 0,0,2
LDA 1,TBLSIZ
ADD 0,1
SBI 1,1
STA 1,1,2
HLV 0
CLM 1,2
JMP *-8-
RAND
JSR @ENTRA
LDA 1,LOWBF
CLM 0,1
JMP *-3
ADI 2,2
E0SP2
DPC0,HITRAD,0
STA 0,08LCL0
STA 2,08LCL1
LDA 1,-2,2
ADD 0,2
SUB 1,2
STA 2,08SSV0
ELEF 1,DPC0+3,-,-1,1
STA 1,0,2
HITRAD
LDA 0,HIGBF
ADI 2,0
STA 0,H0ST8
RNDADR HOST8,HIRFU
?C(A0)=RANDOM #
?GET ADDRESS IN THE RANGE
?C(I03) AND C(MXTBL)
?TABLE SIZE =
?:RANDOM SIZE< C(MXTBL)-2
?C(A0)=RANDOM #
?C(TABLE-2)=RANDOM DATA 1
?C(TABLE-1)=(RANDOM DATA 1+
?TABLE SIZE-1)
?ACTUAL TABLE BEGIN ADDR
?IS AT C(LOWBF)
?MAKE SURE C(TABLE-2) <
?C(TABLE-1)
?C(A0)=RANDOM #
?GET RANDOM DATA 2 IN
?THE RANGE OF C(TABLE-2)
?AND C(TABLE-1)
?A0=DISPATCH ADDEND
?A2=TABLE BEGIN ADDR
?SAVE A0
?SAVE AC1
?
?
?
?SAVE DISPATCH ADDR
?AND STORE (DPC0+3) IN
?DISPATCH ADDR LOCATION
?MAKE SURE THAT RNDM
?ADDR IS NOT EQ. TO
?(TABLE-1) OR (TABLE-2)

```

0087 ECL32

```

01 05012 006307
02 05013 006310
03 05014 002257
04 05015 000315
05 05016 040254
06 05017 024253
07 05020 106415
08 05021 000771
09 05022 030252
10 05023 140710
11 05024 041000
12 05025 141000
13 05026 000000
14 05027 040403
15 05030 024251
16 05031 146170
17 05037 020013
18 05040 042253
19 05041 042254
20 05042 006305
21 05043 020013
22 05044 042253
23 05045 042254
24 05046 000000
25 05047 006305
26 05048 006305
27 05049 006305
28 05050 006305

```

10088 ECL32

```

01 000001 DSPCI:
02 000000
03 000000
04 000000
05 05043 004401
06 05044 054216
07 05045 004313
08 05046 006306
09 05047 006304
10 05050 000144
11 05051 020312
12 05052 024315
13 05053 106400
14 05054 042261
15 05055 006307
16 05056 006310
17 05057 000260
18 05058 000261
19 05059 120110
20 05062 040262
21 05063 006307
22 05064 030312
23 05065 041000
24 05066 024262
25 05067 107000
26 05070 104110
27 05071 045001
28 05072 143370
29 05073 132370
30 05074 000770
31 05075 006307
32 05076 024312
33 05077 106370
34 05100 000775
35 05101 130010
36 05102 040251
37 05103 050252
38 05104 025376
39 05105 113000
40 05106 132400
41 05107 050253
42 05110 166470
43 05112 000032
44 05112 045000
45 05113 020312
46 05114 120010
47 05115 040257
48 05116 040257
49 05117 040257
50 05118 040257
51 05119 040257
52 05120 040257
53 05121 040257
54 05122 040257
55 05123 040257
56 05124 040257
57 05125 040257
58 05126 040257
59 05127 040257
60 05128 040257

```

0087 ECL32

```

01 05012 006307
02 05013 006310
03 05014 002257
04 05015 000315
05 05016 040254
06 05017 024253
07 05020 106415
08 05021 000771
09 05022 030252
10 05023 140710
11 05024 041000
12 05025 141000
13 05026 000000
14 05027 040403
15 05030 024251
16 05031 146170
17 05037 020013
18 05040 042253
19 05041 042254
20 05042 006305
21 05043 020013
22 05044 042253
23 05045 042254
24 05046 000000
25 05047 006305
26 05048 006305
27 05049 006305
28 05050 006305

```



```

00A9 ECL32
01 05116 006307 JSR @ENTRA
02 05117 006310 JSR @RNADR
03 05120 002257 HDSTB
04 05121 000315 HIBFU
05 05122 040254 STA 0,0SSV1
06 05123 024253 LDA 1,0SSV0
07 05124 106415 SUB# 0,1,SMR
08 05125 000771 JMP -7
09 05126 030252 LDA 2,0SLC1
10 05127 140710 XCH 2,0
11 05130 041000 STA 0,0,2
12 05131 141000 MOV 2,0
13 05132 166870 ELEM 1,0PC1-,,1
14 05134 122400 SUB 1,0
15 05135 000000 .NOLOC 0
16 05136 103240 .ADDR 0,0
17 05137 040403 STA 0,0PC1*1
18 05137 024251 LDA 1,0SLC0
19 05140 146570 DPC1: DSPA 1,0,1
20 100000
21 ERROR
22 05146 020013- LDA 0,0SPER
23 05147 042253 STA 0,0SSV0
24 05150 042254 STA 0,0088V1
25 .NOLOC 0
26 LOOP
27 JSR @ENTLO
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

```

10090 ECL32

```

01 000002 DSPC2: DSSWD=2
02 EOSP3
03 DSSMI=0
04 LOMBF,LOBFU,DSFER,DPC2,HITBAD,2
05 LOMBF,LOBFU,DSFER
06 **1
07 3,HELP
08 *1
09 05154 066313 JSR @JFILL
10 05155 006306 DSFER
11
12 SETUP 100.
13 JSR @ENTIN
14 100.
15 0,LOWBF
16 05160 020314 LDA 1,LOBFU
17 05161 024311 SUB 0,1
18 05162 106400 STA 1,MXTBL
19 RNDADR D3,MXTBL
20 RAND
21 05164 066307 JSR @ENTRA
22 05165 066310 JSR @RNADR
23 D3
24 MXTBL
25 SBI 2,0
26 STA 0,TBLSIZ
27 RAND
28 05172 006307 JSR @ENTRA
29 05173 030314 LDA 2,LOWBF
30 05174 041000 STA 0,0,2
31 05175 024262 LDA 1,TBLSIZ
32 05176 107000 ADD 0,1
33 05177 104110 STA 1,1
34 05200 049001 STA 1,1,2
35 05201 143370 HLV 0
36 05202 132370 CLM 1,2
37 05203 000770 JMP *-8.
38 RAND
39 05204 006307 JSR @ENTRA
40 05205 024314 LDA 1,LOWBF
41 05206 106370 CLM 0,1
42 05207 000775 JMP *-3
43 05210 130010 ADI 2,2
44
45 EOSP2
46 05211 040251 STA 0,0SLC0
47 05212 050252 STA 2,0SLC1
48 05213 025376 LDA 1,-2,2
49 05214 113000 ADD 0,2
50 05215 132400 SUB 1,2
51 05216 050253 STA 2,0SSV0
52 05217 166870 ELEM 1,DPC2*3-,,1,1
53 000032
54 05221 045000 STA 1,0,2
55 HITBAD
56 05222 020312 LDA 0,HIGBF
57 05223 120010 ADI 2,0
58 05224 040257 STA 0,HOSTB
59 RNDADR
60 HDSTB,HIBFU

```

```

;C(ACO)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(OSTR) AND C(HIBFU)
;SAVE RNDM ADDR 1
;MAKE SURE IT'S NOT
;EQUAL TO DISPATCH
;ADDR
;STORE TABLE ADDR IN
;RNDM ADDR 1
;
;AC0=AC2=RANDOM ADDR
;AC0=(RNDM ADDR-PC OF
;DSPA+1)
;ADD INDIRECT BIT TO AC0
;STORE THIS IN DSPA
;AC1=DISPATCH ADDR
;EXECUTE DSPA,INDEX=01
;RESTORE THE RNDM ADDRESSES
;AND DISPATCH LOC TO
;JSR @ENTRA.
;ITERATE TEST ROUTINE

```

```

;C(ACO)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(OSTR) AND C(HIBFU)
;SAVE RNDM ADDR 1
;MAKE SURE IT'S NOT
;EQUAL TO DISPATCH
;ADDR
;STORE TABLE ADDR IN
;RNDM ADDR 1
;
;AC0=AC2=RANDOM ADDR
;AC0=(RNDM ADDR-PC OF
;DSPA+1)
;ADD INDIRECT BIT TO AC0
;STORE THIS IN DSPA
;AC1=DISPATCH ADDR
;EXECUTE DSPA,INDEX=01
;RESTORE THE RNDM ADDRESSES
;AND DISPATCH LOC TO
;JSR @ENTRA.
;ITERATE TEST ROUTINE
;C(ACO)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(OSTR) AND C(HIBFU)
;TABLE SIZE =
;1-RANDOM SIZE< C(MXTBL)-2
;C(ACO)=RANDOM #
;C(TABLE-2)= RANDOM DATA 1
;C(TABLE-1)= (RANDOM DATA 1+
;TABLE SIZE-1)
;ACTUAL TABLE BEGIN ADDR
;IS AT C(LOWBF)
;MAKE SURE C(TABLE-2) <
;C(TABLE-1)
;C(ACO)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(OSTR) AND C(HIBFU)
;AC0=DISPATCH ADDR
;AC2=TABLE BEGIN ADDR
;SAVE AC0
;SAVE AC1
;
;SAVE DISPATCH ADDR
;AND STORE (DPC2*3) IN
;DISPATCH ADDR LOCATION
;MAKE SURE THAT RNDM
;ADDR IS NOT EQ. TO
;(TABLE-1) OR (TABLE-2)

```

```

0091 ECL32
01 05225 006307      JSR @ENTRA
02 05226 006310      JSR @RNADR
03 05227 000257      HOSTB
04 05230 000315      HIBFU
05 05231 040254      STA 0,DSSV1
06 05232 042253      LDA 1,DSSV0
07 05233 106415      SUB# 0,1,SNR
08 05234 000771      JMP -7
09 05235 030252      LDA 2,OSL1
10 05236 140710      ACH 2,0
11 05237 041000      STA 0,0,2
12 05240 141000      MOV 2,0
13 05241 111222      MOVZR 0,2,SZC ;AC2=(RANDOM ADDR/2)+
14 05242 110010      ADI 1,2
15 05243 143370      HLV 0
      -NDLOC 0
16 05244 103240      ADDR 0,0
17 05245 040403      STA 0,0,2
18 05246 042251      LDA 1,DSLC0
19 05247 147170      DPC2:
20 05248 100000      OSPA 1,0072
21
22
23 05255 020013-    ERROR
24 05256 042253      LDA 0,0,DSPER
25 05257 042254      STA 0,0,DSSV0
26 05258 000000      -NDLOC 0
27
28 05260 006305      JSR @ENTLO
29
30
0002 ECL32
01 0000003 0SPC3:  DSSW0=3
02
03
04
05
06 05261 004401      HIGBF,HIBFU,OSPER,DPC3,LOTBAD,3
07 05262 054216      EDSP1
08
09 05263 006313      HIGBF,HIRFU,DSPER
10 05264 006306      TSTLOC
11
12
13
14
15
16
17
18
19
20
21 05273 006307      JSR @ENTRA
22 05274 006310      JSR @RNADR
23 05275 000260      D3
24 05276 000261      MKTBL
25 05277 120110      SBI 2,0
26 05300 00262      STA 0,TBLSIZ
27
28 05301 006307      RAND
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
05312 000770      JSR @ENTRA
05313 006307      LDA 2,HIGBF
05314 024312      STA 0,0,2
05315 106370      LDA 1,7BLSIZ
05316 000775      ADD 0,1
05317 130010      SBI 1,1
05318 045001      STA 1,1,2
05319 143370      HLV 0
05320 132370      CLM 1,2
05321 000770      JMP *-8.
05322 006307      RAND
05323 006307      JSR @ENTRA
05324 024312      LDA 1,HIGBF
05325 106370      CLM 0,1
05326 000775      JMP *-3
05327 130010      ADI 2,2
05328 040251      EDSP2
05329 040251      STA 0,DSLCO
05330 050352      STA 2,OSLC1
05331 050352      LDA 1,-2,2
05332 025376      ADD 0,2
05333 113000      SUB 1,2
05334 132400      STA 2,DSSV0
05335 050253      ELEF 1,DPC3+3,-1,1
05336 166470      STA 1,0,2
05337 000032      LOTBAD
05338 045000      LDA 0,L0M8F
05339 020314      ADI 2,0
05340 120010      STA 0,L0STB
05341 040256      RAND
05342 040256

```

```

;C(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(MDSTB) AND C(HIBFU)
;SAVE RNDM ADDR 1
;MAKE SURE IT'S NOT
;EQUAL TO DISPATCH
;ADDR
;STORE TABLE ADDR IN
;RNDM ADDR 1
;
;AC0=AC2=RANDOM ADDR
;BIT 15
;AC0=(RNDM ADDR/2)
;ADD INDIRECT BIT TO AC0
;STORE THIS IN OSPA
;AC1=DISPATCH ADDR
;EXECUTE OSPA,INDEX=02
;
;RESTORE THE RNDM ADDRESSES
;AND DISPATCH LOC TO
;'JSR @ENTER'
;
;ITERATE TEST ROUTINE

```

0093 ECL32  
01 05334 006307 JSR @ENTRA  
02 05335 006310 JSR @ENTRA  
03 05336 000256 LORFB  
04 05337 000311 STA  
05 05340 040254 LDA  
06 05341 024253 LDA  
07 05342 106415 SUB#  
08 05343 000711 JMP  
09 05344 030252 LDA  
10 05345 140710 XCH  
11 05346 041000 STA  
12 05347 141000 MOV  
13 05350 115222 MOVR  
14 05351 114010 ADI  
15 05352 143370 HLV  
16 05353 103240 \*NOLDC 0  
17 05354 040403 ADDOR  
18 05355 040403 STA  
19 05356 147370 LDA  
20 05357 042254 \*NOLDC 0  
21 100000 DSPA  
22 05364 020013- OPCS:  
23 05365 042253  
24 05366 042254  
25 05367 000000  
26 05368 147370  
27 05369 042254  
28 05370 006305  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000

0095 ECL32

```

01 02 05443 006307
02 05444 006310
03 05445 000256
04 05446 000311
05 05447 040254
06 05450 024253
07 05451 106415
08 05452 000771
09 05453 030252
10 05454 140710
11 05455 041000
12 05456 020312
13 05457 120010
14 05458 040257
15 05461 006307
16 05462 006310
17 05463 000257
18 05464 000315
19 05465 040255
20 05466 024253
21 05467 106415
22 05470 000771
23 05471 030254
24 05472 112415
25 05473 000766
26 05474 110710
27 05475 103240
28 05476 041000
29 05477 141000
30 05478 000000
31 05500 103240
32 05501 040403
33 05502 024251
34 05503 146170
35 05503 146170
36 100000
37
38 05511 020013
39 05512 042253
40 05513 042254
41 05514 042255
42
43
44 05515 006305
45
46

```

0096 ECL32

```

01 000001 0SPD1:
02
03
04
05
06 05516 004401
07 05517 054216
08
09 05520 006313
10 05521 006306
11
12
13 05522 006304
14 05523 000104
15 05524 020312
16 05525 024315
17 05526 106400
18 05527 044261
19
20
21 05530 006307
22 05531 006310
23 05532 000260
24 05533 000261
25 05534 120110
26 05535 040262
27
28 05536 006307
29 05537 030312
30 05540 041000
31 05541 024262
32 05542 107400
33 05543 104110
34 05544 045001
35 05545 133370
36 05546 132370
37 05547 000770
38
39 05550 006307
40 05551 024312
41 05552 106370
42 05553 000775
43 05554 130010
44
45
46 05555 040251
47 05556 050252
48 05557 025376
49 05560 113000
50 05561 132400
51 05562 050253
52 05563 166470
53 000053
54 05565 045000
55
56 05566 020312
57 05567 120010
58 05570 040257
59
60

```

0099 ECL32

```

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

```

```

0097 ECL32
01 05371 006307 JSR @ENTRA
02 05372 006310 JSR @RNADR
03 05373 002257 HDSTB
04 05374 000315 HIBFU
05 05375 040254 STA
06 05376 024253 LDA
07 05377 104415 SUB#
08 05400 000771 JMP
09 05601 030252 LDA
10 05602 140710 XCH
11 05603 041000 STA
12 05604 020312 LDA
13 05605 120010 ADI
14 05606 040257 STA
15 05607 006307 JSR @ENTRA
16 05610 006310 JSR @RNADR
17 05611 000257 HDSTB
18 05612 000315 HIBFU
19 05613 040255 STA
20 05614 024253 LDA
21 05615 104415 SUB#
22 05616 000771 JMP
23 05617 030254 LDA
24 05620 112415 SUB#
25 05621 000766 JMP
26 05622 110710 XCH
27 05623 103240 ADDR
28 05624 041000 STA
29 05625 141000 MOV
30 05626 166870 LDA
31 05630 122400 SUB
32 05631 000000 *NOLOC
33 05632 103240 ADDR
34 05633 040403 STA
35 05633 024251 LDA
36 05634 146570 DPD1:
37 100000
38
39 05642 020013-
40 05643 042253 STA
41 05644 042254 STA
42 05645 042255 STA
43 000000 *NOLOC
44 LOOP
45 JSR @ENTLO
46
47
0098 ECL32
01 000002 DSPO2:
02 000001
03 000001
04
05
06 05647 004401 JSR
07 05650 054216 *+1
08 05651 006313 JSR @JFILL
09 05652 006306 DSPPER
10
11
12 SETUP 100.
13 JSR @ENTIN
14 100.
15 05654 000144 LDA
16 05655 020314 LDA
17 05656 024311 SUB
18 05657 106400 STA
19 05660 044261 RNDADR D3,MXTBL
20 RND
21 05661 006307 JSR @ENTRA
22 05662 006310 JSR @RNADR
23 05663 000260 D3
24 05664 000261 MXTBL
25 05665 120110 SBI
26 05666 040262 *STBLSTZ
27
28 05667 006307 JSR @ENTRA
29 05670 030314 LDA
30 05671 041000 STA
31 05672 024262 LDA
32 05673 107000 ADD
33 05674 104110 SBI
34 05675 045001 STA
35 05676 143370 HLV
36 05677 132370 CLM
37 05700 000770 JMP *-8.
38
39 05701 006307 JSR @ENTRA
40 05702 024314 LDA
41 05703 106370 CLM
42 05704 000775 JMP *-3
43 05705 130010 ADI
44
45
46 05706 040251 STA
47 05707 050252 STA
48 05710 025376 LDA
49 05711 113000 ADD
50 05712 132400 SUB
51 05713 050253 STA
52 05714 166470 ELEM
53 05716 040053 STA
54 05716 045000 STA
55 HITB40
56 05717 020312 LDA
57 05720 120010 ADI
58 05721 040257 STA
59 RNDADR HDSTB,HIBFU
60 RND

```

```

;C(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(D3) AND C(MXTBL)
;TABLE SIZE =
;I-RANDOM SIZE< C(MXTBL)-2
;C(AC0)=RANDOM #
;C(TABLE-2)=RANDOM DATA 1
;C(TABLE-1)=(RANDOM DATA 1+
;TABLE SIZE-1)
;ACTUAL TABLE BEGIN ADDR
;IS AT C(L0WBF)
;MAKE SURE C(TABLE-2) <
;C(TABLE-1)
;C(AC0)=RANDOM #
;GET RANDOM DATA 2 IN
;THE RANGE OF C(TABLE-2)
;AND C(TABLE-1)
;AC0=DISPATCH ADDEND
;AC2=TABLE BEGIN ADDR
OPD2,HITB40,2,LOTB40
;SAVE AC0
;SAVE AC1
;
;
;
;SAVE DISPATCH ADDR
;AND STORE (DPD2+3) IN
;DISPATCH ADDR LOCATION
;MAKE SURE THAT RNDM
;ADDR IS NOT EQ. TO
;(TABLE-1) OR (TABLE-2)
RNDADR HDSTB,HIBFU
RND

```

```

;C(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(CHOSTB) AND C(HIBFU)
;SAVE RNDM ADDR 1
;MAKE SURE IT'S NOT
;EQUAL TO DISPATCH
;ADDR
;STORE TABLE ADDR IN
;RNDM ADDR 1
;
;MAKE SURE THAT RNDM
;ADDR IS NOT EQ. TO
;(TABLE-1) OR (TABLE-2)
;C(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(CHOSTB) AND C(HIBFU)
;SAVE RANDOM ADDR 2
;MAKE SURE THAT
;RANDOM ADDR 2 IS NOT
;EQ. TO RNDM ADDR 1 OR
;DISPATCH ADDR
;
;STORE (RANDOM ADDRESS 1)
;IN RANDOM ADDR 2
;AC0=AC2=RANDOM ADDR
;AC0=(RNDM ADDR-PC OF
;DSPFA*1)
;ADD INDIRECT BIT TO AC0
;STORE THIS IN DSPA
;AC1=DISPATCH ADDEND
;EXECUTE DSPA,INDEX=01
;
;RESTORE THE RNDM ADDRESSES
;AND DISPATCH LOC TO
;JSR @ENTER,
;
;ITERATE TEST ROUTINE

```

0099 ECL32

```

01 05722 006307 JSR @ENTRA
02 05723 006310 JSR @RNADR
03 05724 00257 HOSTB
04 05725 000315 HIRFU
05 05726 040254 STA 0,0SSV1
06 05727 024253 LDA 1,0SSV0
07 05730 106415 SUB# 0,1,SNR
08 05731 000771 JWP -7
09 05732 030252 LDA 2,0SLCI
10 05733 140710 XCH 2,0
11 05734 041000 STA 0,0,2
12 05735 020314 LDA 0,LOWBF
13 05736 120010 ADI 2,0
14 05737 040256 STA 0,LOSTB
15 05740 006307 JSR @ENTRA
16 05741 006310 JSR @RNADR
17 05742 000256 LOSTB
18 05743 000311 LOGFU
19 05744 040355 STA 0,0SSV2
20 05745 024253 LDA 1,0SSV0
21 05746 106415 SUB# 0,1,SNR
22 05747 000771 JWP -7
23 05750 030254 LDA 2,0SSV1
24 05751 112415 SUB# 0,2,SNR
25 05752 000766 JMP -10
26 05753 110710 XCH 0,2
27 05754 103240 AODOR 0,0
28 05755 041000 STA 0,0,2
29 05756 141000 MOVZ 2,0
30 05757 112222 ADI 1,2
31 05760 110010 HLV 0
32 05761 143370 -NOLOC
33 05762 103240 AODOR 0,0
34 05763 040403 STA 0,DPO2+1
35 05764 024251 LDA 1,0SLC0
36 05765 147170 DPO2:
37 05765 147170 DPO2:
38 05765 100000
39 ERROR
40 05773 020013- LDA 0,-DSPER
41 05774 022253 STA 0,0SSV0
42 05775 042254 STA 0,0DSSV1
43 05776 042255 STA 0,0DSSV2
44 -NOLOC 0
45 JSR @ENTLO
46 05777 006305
47
48

```

10100 ECL32

```

01 000003 DSP03: DSSW0=3
02 EDSP4 HIRBF,HIRFU,DSPER,DPO3,LOSTBA0,3,HIRTRA0
03 DSSW1=1
04 EDSP1 HIRBF,HIRFU,DSPER
05 TSTLOC
06 JSR *+1
07 STA ;(ADDR OF TEST+1) IS
08 DSPER ;SAVED IN LOC. HELP
09 JSR @JFILL
10 DSPER ;FILL UPPER AND LOWER
11 ;SCRATCH BUFFER AREA
12 ;WITH (DSPER)
13 SETUP 100.
14 JSR @ENTIN
15 100.
16 LOA 0,HIRBF
17 LOA 1,HIRFU
18 SUB 0,1
19 STA 1,MXTBL
20 RNDADR D3,MXTBL
21 RAND
22 JSR @ENTRA
23 JSR @RNADR
24 D3
25 MXTBL
26 SBI 2,0
27 STA 0,TBLSIZ
28 RAND
29 JSR @ENTRA
30 LDA 2,HIRBF
31 STA 0,0,2
32 LDA 1,TBLSIZ
33 ADD 0,1
34 SBI 1,1
35 STA 1,1,2
36 HLV 0
37 CLM 1,2
38 JMP *-6.
39 RAND
40 JSR @ENTRA
41 LDA 1,HIRGF
42 CLM 0,1
43 JMP *-3
44 ADI 2,2
45 FOSP2 DPO3,LOSTBA0,3,HIRTRA0
46 STA 0,0SLC0
47 STA 2,0SLC1
48 LDA 1,-2,2
49 ADD 0,2
50 SUB 1,2
51 STA 2,0SSV0
52 ELEF 1,DPO3+3*-1,1
53
54 STA 1,0,2
55 LOSTBA0
56 LDA 0,LOWBF
57 ADI 2,0
58 STA 0,LOSTB
59 RNDADR LOSTB,LORFU
60 RAND

```

10100 ECL32

```

01 000003 DSP03: DSSW0=3
02 EDSP4 HIRBF,HIRFU,DSPER,DPO3,LOSTBA0,3,HIRTRA0
03 DSSW1=1
04 EDSP1 HIRBF,HIRFU,DSPER
05 TSTLOC
06 JSR *+1
07 STA ;(ADDR OF TEST+1) IS
08 DSPER ;SAVED IN LOC. HELP
09 JSR @JFILL
10 DSPER ;FILL UPPER AND LOWER
11 ;SCRATCH BUFFER AREA
12 ;WITH (DSPER)
13 SETUP 100.
14 JSR @ENTIN
15 100.
16 LOA 0,HIRBF
17 LOA 1,HIRFU
18 SUB 0,1
19 STA 1,MXTBL
20 RNDADR D3,MXTBL
21 RAND
22 JSR @ENTRA
23 JSR @RNADR
24 D3
25 MXTBL
26 SBI 2,0
27 STA 0,TBLSIZ
28 RAND
29 JSR @ENTRA
30 LDA 2,HIRBF
31 STA 0,0,2
32 LDA 1,TBLSIZ
33 ADD 0,1
34 SBI 1,1
35 STA 1,1,2
36 HLV 0
37 CLM 1,2
38 JMP *-6.
39 RAND
40 JSR @ENTRA
41 LDA 1,HIRGF
42 CLM 0,1
43 JMP *-3
44 ADI 2,2
45 FOSP2 DPO3,LOSTBA0,3,HIRTRA0
46 STA 0,0SLC0
47 STA 2,0SLC1
48 LDA 1,-2,2
49 ADD 0,2
50 SUB 1,2
51 STA 2,0SSV0
52 ELEF 1,DPO3+3*-1,1
53
54 STA 1,0,2
55 LOSTBA0
56 LDA 0,LOWBF
57 ADI 2,0
58 STA 0,LOSTB
59 RNDADR LOSTB,LORFU
60 RAND

```

```

0101 ECL32
01 06053 006307
02 06054 006310
03 06055 000256
04 06056 000311
05 06057 040254
06 06060 024253
07 06061 106415
08 06062 000771
09 06063 030252
10 06064 140710
11 06065 041000
12 06066 020312
13 06067 120010
14 06070 040257
15 06071 006307
16 06072 006310
17 06073 000257
18 06074 000315
19 06075 040255
20 06076 024253
21 06077 106415
22 06100 000771
23 06101 030254
24 06102 112415
25 06103 000766
26 06104 110710
27 06105 103240
28 06106 041000
29 06107 141000
30 06110 115222
31 06111 114010
32 06112 143370
33 000000
34 06113 103240
35 06114 040403
36 06115 024251
37 06116 147570 DP03:
38 100000
39
40 06124 020013=
41 06125 042253
42 06126 042254
43 06127 042255
44 000000
45
46 06130 006305
47
48

JSR @ENTRA
JSR @RNADR
LDBT8
LDBFU
STA
LDA
SUB#
JMP
LDA
XCH
STA
LDA
SUB#
JMP
LDA
SUB#
JMP
XCH
ADDR
STA
MOV
ADJ
HLV
.NOLOC
ADDR
STA
LDA
DSPA
ERROR
LDA
STA
STA
STA
.NOLOC
LOOP
JSR @ENTLO

;(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;(LDBT8) AND C(LDBFU)
;SAVE RNDM ADDR 1
;MAKE SURE IT'S NOT
;EQUAL TO DISPATCH
;ADDR
;STORE TABLE ADDR IN
;RNDM ADDR 1
;
;MAKE SURE THAT RNDM
;ADDR IS NOT EG. TO
;(TABLE-1) OR (TABLE-2)
;(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;(HDBT8) AND C(HDBFU)
;
;SAVE RANDOM ADDR 2
;MAKE SURE THAT
;RANDOM ADDR 2 IS NOT
;EG. TO RNDM ADDR 1 OR
;DISPATCH ADDR
;
;STORE (RANDOM ADDRESS 1)
;IN RANDOM ADDR 2
;
;AC0=AC2=RANDOM ADDR
;RNDM ADDR/2)+
;BIT IS
;AC0=(RNDM ADDR/2)
;
;ADD INDIRECT BIT TO AC0
;STORE THIS IN DSPA
;AC1=DISPATCH ADDEND
;EXECUTE DSPA,INDEX=03
;
;RESTORE THE RNDM ADDRESSES
;AND DISPATCH LOC TO
;'JSR @ENTER'
;
;ITERATE TEST ROUTINE

JSR @ENTRA
JSR @RNADR
LDBT8
LDBFU
STA
LDA
SUB#
JMP
LDA
XCH
STA
LDA
SUB#
JMP
XCH
ADDR
STA
MOV
ADJ
HLV
.NOLOC
ADDR
STA
LDA
DSPA
ERROR
LDA
STA
STA
STA
.NOLOC
LOOP
JSR @ENTLO

;(AC0)=RANDOM #
;SET ADDRESS IN THE RANGE
;(CDB3) AND C(MXTBL)
;TABLE SIZE =
;TABLE SIZE < C(MXTBL)-2
;(AC0)=RANDOM #
;(TABLE-2)=RANDOM DATA 1
;(TABLE-1)=(RANDOM DATA 1+
;TABLE SIZE-1)
;ACTUAL TABLE BEGIN ADDR
;IS AT C(HIGBF)
;MAKE SURE C(TABLE-2) <
;(TABLE-1)
;(AC0)=RANDOM #
;SET RANGE OF C(TABLE-2) IN
;AND C(TABLE-1)
;AC0=DISPATCH ADDEND
;AC2=TABLE BEGIN ADDR
;AC1=17777
;STORE TBL ADDR IN
;
;DSPA INSTRUCTION AND
;
;17777 IN DISPATCH LOC.
;DSPA SHOULD RETURN TO
;DPE0+2
;RESTORE DISPATCH LOC.
;TO 'JSR @ENTER'
;ITERATE TEST ROUTINE

```

10102 ECL32

```

01
02
03
04 06131 004401
05 06132 054216
06
07 06133 006313
08 06134 006306
09
10
11 06135 006304
12 06136 000144
13 06137 020312
14 06140 024315
15 06141 106400
16 06142 044261
17
18
19 06143 006307
20 06144 006310
21 06145 000260
22 06146 000261
23 06147 120110
24 06150 040262
25
26 06151 006307
27 06152 030312
28 06153 041000
29 06154 024262
30 06155 107000
31 06156 104110
32 06157 045001
33 06160 143370
34 06161 132370
35 06162 000770
36
37 06163 006307
38 06164 024312
39 06165 106370
40 06166 000775
41 06167 150010
42
43 06170 126000
44 06171 050406
45 06172 035376
46 06173 113000
47 06174 172400
48 06175 045000
49 06176 042170 OPE0:
50 000000
51 06200 101001
52
53 06205 024013=
54 06206 045000
55
56 06207 006305
57

```

10103 ECL32

0104 ECL32  
01 06273 006305  
02

JSR @ENTLO ;ITERATE TEST ROUTINE

```

01 01 06222 006307 JSR @ENTRA
02 02 06223 006310 JSR @RNADR
03 03 06224 000260 D3
04 04 06225 000261 MXTBL
05 05 06226 120110 SBI 2,0
06 06 06227 040262 STA 0,TBLSIZ
07 07 06230 006307 JSR @ENTRA
08 08 06231 050314 LDA 2,LOWRF
09 09 06232 041000 STA 0,0,2
10 10 06233 024262 LDA 1,TBLSIZ
11 11 06234 107000 ADD 0,1
12 12 06235 104110 SBI 1,1
13 13 06236 045001 STA 1,1,2
14 14 06237 143370 HLW 0
15 15 06240 132370 CLM 1,2
16 16 06241 000770 JMP *-8
17
18
19 06242 006307 JSR @ENTRA
20 06243 024314 LDA 1,LOWRF
21 06244 106370 CLM 0,1
22 06245 000775 JMP *-3
23 06246 130010 ADI 2,2
24
25 06247 050251 STA 2,DSLCO
26 06250 152222 MOVZR 2,3,SZC
27 06251 114010 ADI 1,5
28 06252 025376 LDA 1,-2,2
29 06253 113000 ADD 0,2
30 06254 132400 SUB 1,2
31 06255 126000 ADC 1,1
32 06256 045000 STA 1,0,2
33 06257 024251 LDA 1,DSLCO
34 06260 147370 HLW 1
35 06261 044002 STA 1,0PEI+1
36 06262 143570 OPEI: DSPA 0,0,5
37
38 06264 101001 MOV 0,0,8KP
39
40 06271 024013- ERROR
41 06272 045000 LDA 1,-DSPER
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

```

OSPE1: L0WRF,L0BFFU,DSPER

;(ADDR OF TEST+1) IS  
;SAVED IN LOC. HELP  
;FILL UPPER AND LOWER  
;SCRATCH BUFFER AREA  
;WITH (DSPER)

;INITIALIZE TEST.  
;MAX. AVAILABLE AREA  
;FOR TABLE IS =  
;C(L0BFFU)-C(L0WRF)

;C(AC0)=RANDOM #  
;GET ADDRESS IN THE RANGE  
;C(D3) AND C(MXTBL)  
;TABLE SIZE =  
;1<RANDOM SIZE< C(MXTBL)-2

;C(AC0)=RANDOM #  
;C(TABLE-2)= RANDOM DATA 1  
;C(TABLE-1)=(RANDOM DATA 1+  
;TABLE SIZE-1)  
;ACTUAL TABLE BEGIN ADDR  
;IS AT C(L0WRF)  
;MAKE SURE C(TABLE-2) <  
;C(TABLE-1)

;C(AC0)=RANDOM #  
;GET RANDOM DATA 2 IN  
;THE RANGE OF C(TABLE-2)  
;AND C(TABLE-1)  
;AC0=DISPATCH ADDR  
;AC2=TABLE BEGIN ADDR  
;SAVE AC2  
;AC3=(TABLE ADDR/2)+  
;BIT 15

;STORE 17777 IN DISPATCH  
;FAC1=17777  
;LOCATION  
;STORE (TABLE ADDR/2)  
;IN DSPA INSTRUCTION  
;DSPA SHOULD RETURN TO  
;NEXT INSTRUCTION  
;RESTORE DISPATCH LOC TO  
;JSR \$ENTER'

;



```

010105 ECL32
01 06306 006307 JSR @ENTRA
02 06307 006310 JSR @RNADR
03 06310 000260 D3
04 06311 000261 MXTBL
05 06312 120110 SBI 2,0
06 06313 000262 STA 0,TBLSIZ
07 06314 006307 JSR @ENTRA
08 06315 030314 LDA 2,LONBF
09 06316 041000 STA 0,0,2
10 06317 024262 LDA 1,TBLSIZ
11 06318 107000 ADD 0,1
12 06319 104110 SBI 1,1
13 06320 045001 STA 1,1,2
14 06321 143370 HLW 0
15 06322 132370 CLM 1,2
16 06323 000770 JMP *-6.
17
18 06326 006307 JSR @ENTRA
19 06327 024314 LDA 1,LONBF
20 06328 106370 CLM 0,1
21 06329 000775 JMP *-3
22 06330 130010 ADI 2,2
23 06331 050427 STA 2,DPF0+1
24 06332 025376 LDA 1,-2,2
25 06333 113000 ADD 0,2
26 06334 132400 SUB 1,2
27 06335 040251 STA 0,OSLCO
28 06336 050252 STA 2,OSLCO
29 06337 020312 LDA 0,HIGBF
30 06338 120010 STA 2,0
31 06339 040257 ADI 0,HOSTR
32 06340 050257 RNDADR HOSTB,HIBFU
33
34 06344 006307 JSR @ENTRA
35 06345 006310 JSR @RNADR
36 06346 000257 HDSTB
37 06347 000315 HIBFU
38 06348 105000 MOV 0,1
39 06349 127240 AODDR 1,1
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

```

```

0106 ECL32
01 06352 034252 :RNDM ADDR IS STORED
02 06353 045400 :IN DISPATCH ADDR LOC
03 06354 166470 :ALSO STORE ADDR (DPF0+3)
04 06355 000007 :
05 06356 115000 :IN RNDM ADDR
06 06357 045400 :
07 06360 030251 :AC2=DISPATCH ADDEND
08 06361 152170 DPF0 :EXECUTE DSPA INSTRUCTION
09
10
11 06367 024013= :RESTORE DISPATCH ADDR
12 06370 045400 :LOC. AND RNDM ADDR TO
13 06371 046252 :JSR @ENTER,
14
15 06372 006305 :ITERATE TEST ROUTINE
16

```

```

0106 ECL32
01 06352 034252 :RNDM ADDR IS STORED
02 06353 045400 :IN DISPATCH ADDR LOC
03 06354 166470 :ALSO STORE ADDR (DPF0+3)
04 06355 000007 :
05 06356 115000 :IN RNDM ADDR
06 06357 045400 :
07 06360 030251 :AC2=DISPATCH ADDEND
08 06361 152170 DPF0 :EXECUTE DSPA INSTRUCTION
09
10
11 06367 024013= :RESTORE DISPATCH ADDR
12 06370 045400 :LOC. AND RNDM ADDR TO
13 06371 046252 :JSR @ENTER,
14
15 06372 006305 :ITERATE TEST ROUTINE
16

```

```

0106 ECL32
01 06352 034252 :RNDM ADDR IS STORED
02 06353 045400 :IN DISPATCH ADDR LOC
03 06354 166470 :ALSO STORE ADDR (DPF0+3)
04 06355 000007 :
05 06356 115000 :IN RNDM ADDR
06 06357 045400 :
07 06360 030251 :AC2=DISPATCH ADDEND
08 06361 152170 DPF0 :EXECUTE DSPA INSTRUCTION
09
10
11 06367 024013= :RESTORE DISPATCH ADDR
12 06370 045400 :LOC. AND RNDM ADDR TO
13 06371 046252 :JSR @ENTER,
14
15 06372 006305 :ITERATE TEST ROUTINE
16

```

10107 ECL32

```

01
02
03 06373 00401      DSPF1:  TSTLOC
04 06374 054216    JSR      +1
05      3,HELP
06      100.
07 06375 006304    JSR @ENTIN
08 06377 162470    ELEM 0,DPF1+3,-1,1 :STORE ADDR (DPF1+3) IN
09      100.
10 06401 040416    STA 0,DPF10
11 06402 162470    ELEM 0,DPF10,-1,1 :STORE (@DPF10) IN
12      100.
13 06404 103240    ADDROR 0,0
14 06405 040416    STA 0,DPTB0+3
15 06406 152520    SUBZL 2,2
16 06407 152570    DSPA 2,DPTB0+2,-1,1 :EXECUTE DSPA
17      000012
18      ERROR
19      LOOP
20 06415 006305    JSR @ENTLO
21 06416 000406    JMP  DPTB0+4
22 06417 000000    DPF10: 0
23 06420 000000    DPTB0: 0
24 06421 000001    1
25 06422 177777    177777
26 06423 000000    0
27

```

10108 ECL32

```

01
02
03 06424 004401      DSPF2:  TSTLOC
04 06425 054216    JSR      +1
05      3,HELP
06      100.
07 06426 006304    JSR @ENTIN
08 06430 162470    ELEM 0,DPF2+3,-1,1 :STORE ADDR (DPF2+3) IN
09      100.
10 06432 040416    STA 0,DPF20
11 06433 162470    ELEM 0,DPF20,-1,1 :STORE (@DPF20) IN
12      100.
13 06435 103240    ADDROR 0,0
14 06436 040415    STA 0,DPTB1+2
15 06437 150510    XOR 2,2
16 06440 152570    DSPA 2,DPTB1+2,-1,1 :EXECUTE DSPA
17      000012
18      ERROR
19      LOOP
20 06446 006305    JSR @ENTLO
21 06447 000406    JMP  DPTB1+4
22 06450 000000    DPF20: 0
23 06451 000000    DPTB1: 0
24 06452 000001    1
25 06453 000000    0
26 06454 177777    177777
27

```

```

10109 ECL32
01
02 DSPF3: TSTLOC
03 JSR *+1 ;(ADDR OF TEST+1) IS
04 STA 3,HELP ;(SAVED IN LOC. HELP
05 SETUP 100.
06 JSR @ENTIN ;INITIALIZE TEST.
07 ELEF 0,DPF3*-1,1 ;STORE ADDR (DPF3+3) IN
08 STA 0,DPF3
09 ELEF 0,DPF3*-1,1 ;STORE (@DPF3) IN
10 ADDR 0,0 ;DISPATCH LOCATION
11 STA 0,DPTB2+3
12 SUBZL 2,2 ;DISPATCH ADDEND IN AC2=1
13 ELEF 0,DPTB2+1--1,1 ;STORE (TBL ADDR+1) IN
14 STA 0,DPF3+1 ;(DPF3+1)
15 DSPA 2,0,2 ;EXECUTE DSPA
16 ERROR
17 LOOP
18 JSR @ENTLO ;ITERATE TEST ROUTINE
19 JMP DPTB2+4 ;GO TO NEXT TEST
20 DPF3: 0
21 DPTB2: 0
22 DPTB2: 1
23 DPTB2: 17777
24 DPTB2: 0
25 DPTB2: 0
26 DPTB2: 0
27 DPTB2: 0
28 DPTB2: 0
29 DPTB2: 0
30

10110 ECL32
01
02 DSPF4: TSTLOC
03 JSR *+1 ;(ADDR OF TEST+1) IS
04 STA 3,HELP ;(SAVED IN LOC. HELP
05 SETUP 100.
06 JSR @ENTIN ;INITIALIZE TEST.
07 ELEF 0,DPF4+3*-1,1 ;STORE ADDR (DPF4+3) IN
08 STA 0,DPF4
09 ELEF 0,DPF4*-1,1 ;STORE (@DPF4) IN
10 ADDR 0,0 ;DISPATCH LOCATION
11 STA 0,DPTB3+3
12 SUBZL 3,3 ;DISPATCH ADDEND IN AC3=1
13 ELEF 0,DPTB3+1--1,1 ;STORE (TBL ADDR-1) IN
14 STA 0,DPF4+1 ;(DPF4+1)
15 DSPA 3,0,3 ;EXECUTE DSPA
16 ERROR
17 LOOP
18 JSR @ENTLO ;ITERATE TEST ROUTINE
19 JMP DPTB3+4 ;GO TO NEXT TEST
20 DPF4: 0
21 DPTB3: 0
22 DPTB3: 1
23 DPTB3: 17777
24 DPTB3: 0
25 DPTB3: 0
26 DPTB3: 0
27 DPTB3: 0
28 DPTB3: 0
29 DPTB3: 0
30

```

10111 ECL32

```

01 02 05545 014204 EXTR:      PASSIN      0,0,SZC
03 06546 000437 JMP          DOMOR
04 06547 010203 ISZ          PASS
05 06550 010001 MOV          0,0,SKP
06 06551 065077 HALT
07 06552 020205 STA          0,PASSVL
08 06553 040204 STA          0,PASSIN
09
10 06554 060477 READS      0
11 06555 101112 MOVL#
12 06556 000403 JMP          +3
13 06557 122470 ELDA          0,SWREG
14
15 06561 143770 ANDI          184,0
16
17 06563 101004 MOV          0,0,SZR
18 06564 000406 JMP          PSCK1
19
20 06565 062222 JSR          @IMSS
21 06566 001450 PSMES
22 06567 125020 MOV
23 06570 020203 LDA          1,1
24 06571 006225 JSR          @PDEC
25
26 06572 034010-PSCK1: LDA          3,-EGGS
27 06573 021400 LDA          0,0,3
28 06574 101005 MOV          0,0,SNR
29 06575 000410 JMP          +8,
30 06576 015403 DSZ          3,3
31 06577 000406 JMP          +6
32 06600 062677 IDPST
33 06601 021403 LDA          0,3,3
34 06602 035404 LDA          3,4,3
35 06603 041776 STA          0,-2,3
36 06604 001400 JMP          0,3
37
38 06605 006221 DOMOR: CALL
39
40 06606 001172 JSR          @ICAL
41
42
43
44

```

10112 ECL32

```

01
02
03
04
05 06607 000000 AUTO:      0
06 06610 000000 DEV:      0
07 06611 000000 CATSW:   0
08 06612 000000 PCNT:    0
09 06613 000000 RTRN:    0
10 06614 000000 SWRES:   0
11
12 06615 047503          .TXI /COPYRIGHT(C)DGC,1974,76
13 06616 054520
14 06617 044522
15 06618 024124
16 06619 024503
17 06620 043504
18 06621 026103
19 06622 034461
20 06623 032067
21 06624 033454
22 06625 040466 ALL RIGHTS RESERVED/
23
24 06630 051040
25 06631 043511
26 06632 022110
27 06633 020123
28 06634 042522
29 06635 042523
30 06636 053122
31 06637 042105
32 06638 000000
33
34 06643 141705 DIRT:      .TXTE !ECLIPSE32!
35
36 06644 144714
37 06645 031705
38 06646 130662
39 06647 000000
40 06648 000000
41 06649 000000
42 06650 002000 DTOSB
43 06651 175772
44 06652 000000
45 06653 000000
46 06654 000000
47 06655 000000
48 06656 000000
49
50 06661 006661 PRGEND: .
51
52
53
54
55
56
57 06662 000144 HIRUFF: .BLK 100.
58
59
60

```

RELOCATABLE UPPER BUFFER USED IN ABOVE TEST

END DTOSB

0113 ECL32  
 01 00013-006306  
 02 000240  
 03 177770  
 04 000212  
 05 104400  
 06 022000  
 07 000377  
 08 000060  
 09 000011  
 10 033031  
 11 000144  
 12 000200  
 13 000400  
 14 001777  
 15 000304  
 16 000274  
 17 000010

0114 ECL32  
 AC0 000226  
 AC1 000227  
 AC2 000230  
 AC3 000231

AUTO 006607  
 BAMBL 000274  
 BAMBL 000264  
 BEG 000727  
 BEGIN 001563  
 BGNAD 000202  
 BUFP 000263  
 CAL 001144  
 CALO 000242  
 CALL 000243  
 CALL 000244  
 CALL 001220  
 CATSW 006611  
 CHAR 001365  
 CHAR1 001414  
 CHAR2 001423  
 CHAR3 001431  
 CHAR4 001442  
 CHASE 00235  
 CHORZ 000236  
 CHRVS 001447  
 CRY 000232  
 D3 000260

MC

DEV 006610  
 DIRT 006643  
 DMMOR 006605  
 DP80 004141  
 DP81 004526  
 DP82 004314  
 DP83 004402  
 DP84 004463  
 DP85 004547  
 DP86 004636  
 DP87 004725  
 OPC0 005031  
 OPC1 005140  
 OPC2 005247  
 OPC3 005356  
 OPD0 005503

|        |        |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|--------|
| 23/16  | 27/08  | 27/30  | 28/21  | 28/34  | 29/05  | 29/28  |
| 29/58  | 28/22  | 28/35  | 29/15  | 29/30  | 29/46  | 32/04  |
| 32/18  | 28/23  | 28/36  | 29/16  | 29/32  | 29/47  | 32/05  |
| 32/19  | 29/34  | 29/59  | 38/39  | 38/41  | 38/43  | 39/28  |
| 39/30  | 39/32  | 40/32  | 41/34  | 42/35  | 43/35  | 44/27  |
| 44/31  | 44/35  | 45/44  | 46/45  | 47/46  | 48/46  | 49/37  |
| 50/58  | 51/59  | 52/59  | 53/33  | 53/35  | 53/37  | 54/28  |
| 54/30  | 54/32  | 55/32  | 56/34  | 57/35  | 58/35  | 59/27  |
| 59/31  | 59/36  | 60/44  | 61/45  | 62/46  | 63/46  | 64/57  |
| 65/58  | 66/59  | 67/59  | 68/28  | 69/57  | 69/59  | 71/54  |
| 71/56  | 72/55  | 73/59  | 75/60  | 77/59  | 79/54  | 80/58  |
| 83/01  | 84/60  | 87/21  | 89/22  | 91/23  | 93/23  | 95/38  |
| 97/39  | 99/40  | 101/40 | 102/53 | 103/58 | 106/11 | 107/19 |
| 108/19 | 109/22 | 110/22 |        |        |        |        |
| 26/14  | 29/48  | 112/05 |        |        |        |        |
| 24/08  | 24/09  | 24/17  |        |        |        |        |
| 24/04  | 26/42  | 31/41  |        |        |        |        |
| 26/31  |        | 38/05  |        |        |        |        |
| 24/14  | 31/44  |        |        |        |        |        |
| 22/45  | 22/47  |        |        |        |        |        |
| 23/48  | 26/40  |        |        |        |        |        |
| 23/08  | 30/22  |        |        |        |        |        |
| 23/30  | 30/22  | 30/29  |        |        |        |        |
| 23/31  | 30/23  | 30/30  |        |        |        |        |
| 23/32  | 30/27  | 30/31  |        |        |        |        |
| 21/18  | 24/11  | 31/42  | 111/37 | 112/07 |        |        |
| 22/31  | 25/18  | 25/39  | 111/37 |        |        |        |
| 33/38  | 33/51  | 33/54  | 34/03  |        |        |        |
| 34/30  | 34/47  | 35/01  |        |        |        |        |
| 34/33  | 34/39  |        |        |        |        |        |
| 34/40  | 34/46  |        |        |        |        |        |
| 34/27  | 34/57  |        |        |        |        |        |
| 23/25  | 34/04  |        |        |        |        |        |
| 34/26  | 34/30  | 34/53  |        |        |        |        |
| 34/05  | 34/52  | 34/50  | 34/57  |        |        |        |
| 23/20  | 29/07  | 35/03  |        |        |        |        |
| 23/45  | 69/25  | 29/25  | 32/07  | 32/16  | 75/21  | 77/21  |
| 79/21  | 80/21  | 71/21  | 72/23  | 73/21  | 88/23  | 90/23  |
| 92/23  | 94/24  | 82/21  | 84/21  | 86/24  | 88/23  | 102/21 |
| 105/21 |        | 96/23  | 98/23  | 100/23 | 102/21 | 103/21 |
| 112/06 |        |        |        |        |        |        |
| 22/05  | 112/35 |        |        |        |        |        |
| 111/03 | 111/37 |        |        |        |        |        |
| 72/45  | 73/49  |        |        |        |        |        |
| 73/44  | 73/47  | 72/52  |        |        |        |        |
| 75/48  | 75/56  | 73/52  | 73/56  |        |        |        |
| 77/48  | 77/56  | 75/57  |        |        |        |        |
| 79/43  | 79/47  | 77/56  |        |        |        |        |
| 80/44  | 80/47  | 79/51  |        |        |        |        |
| 82/48  | 82/56  | 80/52  |        |        |        |        |
| 84/48  | 84/55  | 82/58  |        |        |        |        |
| 86/53  | 87/14  | 84/57  |        |        |        |        |
| 89/52  | 89/13  | 87/16  |        |        |        |        |
| 90/52  | 91/13  | 89/17  |        |        |        |        |
| 92/52  | 93/13  | 91/18  |        |        |        |        |
| 94/53  | 95/31  | 93/18  |        |        |        |        |
|        |        | 95/33  |        |        |        |        |

0115 ECL32

|       |        |        |        |        |        |        |        |  |
|-------|--------|--------|--------|--------|--------|--------|--------|--|
| DPD1  | 065634 | 96/52  | 97/30  | 97/34  | 97/36  |        |        |  |
| DPD2  | 065765 | 98/52  | 99/30  | 99/35  | 99/37  |        |        |  |
| DPD3  | 065116 | 100/52 | 101/30 | 101/35 | 101/37 |        |        |  |
| DPD4  | 065176 | 102/44 | 102/49 |        |        |        |        |  |
| DPD5  | 065262 | 103/53 | 103/54 |        |        |        |        |  |
| DPD6  | 065361 | 105/43 | 106/03 |        |        |        |        |  |
| DPD7  | 065407 | 107/08 | 107/16 | 106/08 |        |        |        |  |
| DPD8  | 065417 | 107/10 | 107/11 | 107/22 |        |        |        |  |
| DPD9  | 065440 | 108/08 | 108/16 | 108/22 |        |        |        |  |
| DPD10 | 065450 | 108/10 | 108/11 | 109/19 |        |        |        |  |
| DPD11 | 065474 | 109/08 | 109/18 | 109/25 |        |        |        |  |
| DPD12 | 065504 | 109/10 | 109/11 | 109/25 |        |        |        |  |
| DPD13 | 065530 | 109/08 | 110/11 | 110/19 |        |        |        |  |
| DPD14 | 065540 | 110/08 | 110/11 | 110/25 |        |        |        |  |
| DPD15 | 065540 | 110/10 | 110/11 | 110/25 |        |        |        |  |
| DPD16 | 065620 | 107/14 | 107/16 | 107/21 | 107/23 |        |        |  |
| DPD17 | 065451 | 108/14 | 108/16 | 108/21 | 108/23 |        |        |  |
| DPD18 | 065505 | 109/14 | 109/16 | 109/24 | 109/26 |        |        |  |
| DPD19 | 065541 | 110/14 | 110/16 | 110/24 | 110/26 |        |        |  |
| DSL00 | 000251 | 23/34  | 69/47  | 69/52  | 71/83  | 71/48  | 73/48  |  |
|       |        | 75/43  | 75/51  | 77/43  | 77/51  | 80/43  | 82/43  |  |
|       |        | 82/51  | 84/43  | 84/51  | 86/47  | 87/17  | 88/46  |  |
|       |        | 90/46  | 91/19  | 92/46  | 93/19  | 94/47  | 95/34  |  |
|       |        | 97/35  | 98/46  | 99/36  | 100/46 | 101/36 | 103/43 |  |
|       |        |        | 105/47 | 106/07 |        |        |        |  |
|       |        | 23/39  | 71/44  | 71/49  | 75/47  | 76/01  | 77/47  |  |
|       |        | 82/47  | 83/02  | 84/47  | 85/01  | 86/48  | 87/10  |  |
|       |        | 89/09  | 90/47  | 91/09  | 92/47  | 93/09  | 94/48  |  |
|       |        | 96/47  | 97/09  | 98/47  | 99/09  | 100/47 | 101/09 |  |
|       |        |        |        |        |        |        |        |  |
|       |        | 106/01 | 106/13 |        |        |        |        |  |
|       |        | 69/06  |        |        |        |        |        |  |
| DSL01 | 000252 | 71/02  |        |        |        |        |        |  |
| DSB00 | 003730 | 72/04  |        |        |        |        |        |  |
| DSB01 | 004011 | 73/02  |        |        |        |        |        |  |
| DSB02 | 004152 | 75/02  |        |        |        |        |        |  |
| DSB03 | 004237 | 77/02  |        |        |        |        |        |  |
| DSB04 | 004326 | 79/02  |        |        |        |        |        |  |
| DSB05 | 004414 | 80/02  |        |        |        |        |        |  |
| DSB06 | 004474 | 82/02  |        |        |        |        |        |  |
| DSB07 | 004560 | 84/02  |        |        |        |        |        |  |
| DSB08 | 004650 | 86/02  |        |        |        |        |        |  |
| DSB09 | 004737 | 88/01  |        |        |        |        |        |  |
| DSB10 | 005043 | 90/01  |        |        |        |        |        |  |
| DSB11 | 005152 | 92/01  |        |        |        |        |        |  |
| DSB12 | 005261 | 94/02  |        |        |        |        |        |  |
| DSB13 | 005370 | 96/01  |        |        |        |        |        |  |
| DSB14 | 005516 | 98/01  |        |        |        |        |        |  |
| DSB15 | 005647 | 100/01 |        |        |        |        |        |  |
| DSB16 | 006000 | 102/02 |        |        |        |        |        |  |
| DSB17 | 006131 | 103/02 |        |        |        |        |        |  |
| DSB18 | 006210 | 105/02 |        |        |        |        |        |  |
| DSB19 | 006306 | 72/10  | 72/55  | 73/08  | 73/59  | 75/08  | 75/60  |  |
|       |        | 77/06  | 77/59  | 79/08  | 79/54  | 80/08  | 82/08  |  |
|       |        | 83/01  | 84/08  | 84/60  | 86/11  | 87/21  | 88/10  |  |
|       |        | 90/10  | 91/23  | 92/10  | 93/23  | 94/11  | 95/38  |  |
|       |        | 97/39  | 98/10  | 99/40  | 100/10 | 101/40 | 102/08 |  |
|       |        | 103/08 | 103/58 | 105/08 | 106/11 |        |        |  |
|       |        |        |        |        |        |        |        |  |
|       |        | 107/02 |        |        |        |        |        |  |
| DSPF0 | 006274 | 107/02 |        |        |        |        |        |  |
| DSPF1 | 006373 | 109/02 |        |        |        |        |        |  |
| DSPF2 | 006424 | 109/02 |        |        |        |        |        |  |

0116 ECL32

|       |        |        |        |        |        |        |       |       |
|-------|--------|--------|--------|--------|--------|--------|-------|-------|
| DSPF3 | 006455 | 109/02 |        |        |        |        |       |       |
| DSPF4 | 006511 | 110/02 |        |        |        |        |       |       |
| DS8V0 | 000253 | 23/40  | 86/52  | 87/07  | 87/13  | 87/22  | 88/51 | 89/06 |
|       |        | 89/12  | 89/23  | 90/51  | 91/06  | 91/12  | 91/24 | 92/51 |
|       |        | 93/06  | 93/12  | 93/24  | 94/52  | 95/07  | 95/21 | 95/39 |
|       |        | 96/51  | 97/06  | 97/20  | 97/40  | 98/51  | 99/06 | 99/20 |
|       |        | 99/41  | 100/51 | 101/06 | 101/20 | 101/41 |       |       |
| DS8V1 | 000254 | 23/41  | 87/04  | 87/13  | 87/23  | 89/05  | 89/12 | 89/24 |
|       |        | 91/05  | 91/12  | 91/25  | 93/05  | 93/12  | 93/25 | 95/06 |
|       |        | 95/24  | 95/40  | 97/05  | 97/23  | 97/41  | 99/05 | 99/23 |
|       |        | 99/42  | 101/05 | 101/23 | 101/42 |        |       |       |
| DS8V2 | 000255 | 23/42  | 87/13  | 87/24  | 89/12  | 89/25  | 91/12 | 91/26 |
|       |        | 93/12  | 93/26  | 95/20  | 95/41  | 97/19  | 97/42 | 99/19 |
|       |        | 99/43  | 101/19 | 101/43 |        |        |       |       |
| DS8W0 | 000003 | 86/02  | 87/14  | 88/01  | 89/13  | 89/15  | 90/01 | 91/13 |
|       |        | 92/01  | 93/13  | 94/02  | 95/31  | 96/01  | 97/30 | 97/32 |
|       |        | 98/01  | 99/30  | 100/01 | 101/30 |        |       |       |
| DS8W1 | 000001 | 86/04  | 87/13  | 87/24  | 88/03  | 89/12  | 89/25 | 90/03 |
|       |        | 91/12  | 91/24  | 92/03  | 93/12  | 93/26  | 94/04 | 95/13 |
|       |        | 95/41  | 96/03  | 97/12  | 97/42  | 98/03  | 99/12 | 99/43 |
|       |        | 100/03 | 101/12 | 101/43 |        |        |       |       |
| DS240 | 002631 | 53/06  |        |        |        |        |       |       |
| DS2A1 | 002673 | 54/01  |        |        |        |        |       |       |
| DS2B0 | 002735 | 55/01  |        |        |        |        |       |       |
| DS2B1 | 002767 | 56/01  |        |        |        |        |       |       |
| DS2B2 | 003024 | 57/01  |        |        |        |        |       |       |
| DS2B3 | 003061 | 58/01  |        |        |        |        |       |       |
| DS2B4 | 003116 | 59/02  |        |        |        |        |       |       |
| DS2C0 | 003185 | 60/02  |        |        |        |        |       |       |
| DS2C1 | 003227 | 61/01  |        |        |        |        |       |       |
| DS2C2 | 003274 | 62/01  |        |        |        |        |       |       |
| DS2C3 | 003341 | 63/01  |        |        |        |        |       |       |
| DS2D0 | 003406 | 64/02  |        |        |        |        |       |       |
| DS2D1 | 003462 | 65/01  |        |        |        |        |       |       |
| DS2D2 | 003541 | 66/01  |        |        |        |        |       |       |
| DS2D3 | 003620 | 67/01  |        |        |        |        |       |       |
| DS2E0 | 003677 | 68/06  |        |        |        |        |       |       |
| DS2E0 | 000200 | 22/45  | 112/42 | 112/60 |        |        |       |       |
| DS2E0 | 002753 | 55/22  | 55/24  |        |        |        |       |       |
| DS2E1 | 003010 | 56/24  | 56/24  |        |        |        |       |       |
| DS2E2 | 003045 | 57/22  | 57/22  |        |        |        |       |       |
| DS2E3 | 003102 | 58/22  | 58/27  |        |        |        |       |       |
| DS2E4 | 003133 | 59/20  | 59/31  |        |        |        |       |       |
| DS2E5 | 003155 | 59/23  | 59/32  |        |        |        |       |       |
| DS2E6 | 003213 | 60/33  | 60/36  |        |        |        |       |       |
| DS2E7 | 003260 | 61/32  | 61/37  |        |        |        |       |       |
| DS2E8 | 003325 | 62/32  | 62/38  |        |        |        |       |       |
| DS2E9 | 003372 | 63/32  | 63/38  |        |        |        |       |       |
| DS2F0 | 003446 | 64/36  | 64/49  |        |        |        |       |       |
| DS2F1 | 003432 | 64/33  | 64/42  |        |        |        |       |       |
| DS2F2 | 003525 | 65/45  | 65/50  |        |        |        |       |       |
| DS2F3 | 003506 | 65/32  | 65/41  |        |        |        |       |       |
| DS2F4 | 003604 | 66/45  | 66/51  |        |        |        |       |       |
| DS2F5 | 003565 | 66/32  | 66/41  |        |        |        |       |       |
| DS2F6 | 003663 | 67/45  | 67/51  |        |        |        |       |       |
| DS2F7 | 003644 | 67/32  | 67/41  |        |        |        |       |       |
| DS2F8 | 003727 | 68/16  | 68/17  |        |        |        |       |       |
| DS2F9 | 003707 | 68/17  | 68/20  |        |        |        |       |       |
| DS2F0 | 003707 | 68/17  | 68/25  |        |        |        |       |       |
| DS2F1 | 003707 | 68/17  | 68/31  |        |        |        |       |       |







0121 ECL32

0122 ECL32

|              |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
|--------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| LOBFU 000311 | 24/32  | 32/29  | 38/25  | 40/18  | 42/18  | 45/19  | 46/18  | 33/05  | 33/13  | 33/25  | 33/37  | 33/05  | 33/13  | 33/25  | 33/37  | 33/05  | 33/13  | 33/25  | 33/37  | 33/05  | 33/13  | 33/25  | 33/37  |
|              | 47/18  | 48/18  | 49/19  | 50/18  | 51/18  | 52/18  | 53/19  | 23/27  | 23/27  | 23/27  | 23/27  | 23/27  | 23/27  | 23/27  | 23/27  | 23/27  | 23/27  | 23/27  | 23/27  | 23/27  | 23/27  | 23/27  | 23/27  |
|              | 55/18  | 58/18  | 59/18  | 60/19  | 61/18  | 62/18  | 63/18  | 28/25  | 28/25  | 28/25  | 28/25  | 28/25  | 28/25  | 28/25  | 28/25  | 28/25  | 28/25  | 28/25  | 28/25  | 28/25  | 28/25  | 28/25  | 28/25  |
|              | 64/19  | 65/18  | 66/18  | 67/18  | 68/18  | 69/18  | 70/18  | 33/11  | 33/11  | 33/11  | 33/11  | 33/11  | 33/11  | 33/11  | 33/11  | 33/11  | 33/11  | 33/11  | 33/11  | 33/11  | 33/11  | 33/11  | 33/11  |
|              | 84/14  | 86/17  | 90/16  | 93/04  | 95/05  | 98/16  | 99/18  | 23/03  | 25/23  | 31/33  | 112/50 | 23/03  | 25/23  | 31/33  | 112/50 | 23/03  | 25/23  | 31/33  | 112/50 | 23/03  | 25/23  | 31/33  | 112/50 |
|              | 101/08 | 103/14 | 105/14 |        |        |        |        | 111/18 | 111/26 |        |        | 111/18 | 111/26 |        |        | 111/18 | 111/26 |        |        | 111/18 | 111/26 |        |        |
| LOBUF 000500 | 23/48  | 24/22  | 24/32  | 25/07  | 41/34  | 42/35  | 43/35  | 33/05  | 33/13  | 30/16  | 36/30  | 33/05  | 33/13  | 30/16  | 36/30  | 33/05  | 33/13  | 30/16  | 36/30  | 33/05  | 33/13  | 30/16  | 36/30  |
|              | 23/48  | 24/22  | 24/32  | 25/07  | 41/34  | 42/35  | 43/35  | 40/14  | 41/09  | 41/14  | 42/09  | 40/14  | 41/09  | 41/14  | 42/09  | 40/14  | 41/09  | 41/14  | 42/09  | 40/14  | 41/09  | 41/14  | 42/09  |
|              | 21/05  | 28/43  | 39/32  | 47/46  | 48/46  | 49/57  | 50/58  | 44/10  | 45/10  | 45/15  | 45/25  | 44/10  | 45/10  | 45/15  | 45/25  | 44/10  | 45/10  | 45/15  | 45/25  | 44/10  | 45/10  | 45/15  | 45/25  |
|              | 44/35  | 45/44  | 46/45  | 54/32  | 55/32  | 56/34  | 57/35  | 48/14  | 48/24  | 48/24  | 48/24  | 48/14  | 48/24  | 48/24  | 48/24  | 48/14  | 48/24  | 48/24  | 48/24  | 48/14  | 48/24  | 48/24  | 48/24  |
|              | 51/59  | 52/59  | 53/37  | 61/45  | 62/46  | 63/46  | 64/45  | 49/15  | 49/25  | 49/25  | 49/25  | 49/15  | 49/25  | 49/25  | 49/25  | 49/15  | 49/25  | 49/25  | 49/25  | 49/15  | 49/25  | 49/25  | 49/25  |
|              | 65/58  | 66/59  | 67/59  | 68/28  | 69/59  | 71/56  | 72/57  | 49/09  | 49/15  | 51/24  | 51/24  | 49/09  | 49/15  | 51/24  | 51/24  | 49/09  | 49/15  | 51/24  | 51/24  | 49/09  | 49/15  | 51/24  | 51/24  |
|              | 68/28  | 69/59  | 71/56  | 75/56  | 80/60  | 83/04  | 85/03  | 51/09  | 51/14  | 51/24  | 51/24  | 51/09  | 51/14  | 51/24  | 51/24  | 51/09  | 51/14  | 51/24  | 51/24  | 51/09  | 51/14  | 51/24  | 51/24  |
|              | 74/01  | 76/03  | 78/02  | 79/56  | 80/60  | 83/04  | 85/03  | 52/34  | 53/14  | 53/24  | 54/10  | 52/34  | 53/14  | 53/24  | 54/10  | 52/34  | 53/14  | 53/24  | 54/10  | 52/34  | 53/14  | 53/24  | 54/10  |
|              | 81/25  | 89/26  | 91/27  | 93/27  | 95/43  | 97/44  | 99/45  | 56/09  | 56/14  | 57/09  | 57/14  | 56/09  | 56/14  | 57/09  | 57/14  | 56/09  | 56/14  | 57/09  | 57/14  | 56/09  | 56/14  | 57/09  | 57/14  |
|              | 101/45 | 102/55 | 103/60 | 106/14 | 107/19 | 108/19 | 109/22 | 60/10  | 60/15  | 60/25  | 61/09  | 60/10  | 60/15  | 60/25  | 61/09  | 60/10  | 60/15  | 60/25  | 61/09  | 60/10  | 60/15  | 60/25  | 61/09  |
|              | 110/22 | 24/28  | 28/04  |        |        |        |        | 62/14  | 62/24  | 63/09  | 63/14  | 62/14  | 62/24  | 63/09  | 63/14  | 62/14  | 62/24  | 63/09  | 63/14  | 62/14  | 62/24  | 63/09  | 63/14  |
| L0P 000767   | 28/13  | 28/37  | 28/40  | 28/09  | 28/49  | 100/55 | 42/17  | 66/14  | 66/24  | 66/34  | 67/09  | 66/14  | 66/24  | 66/34  | 67/09  | 66/14  | 66/24  | 66/34  | 67/09  | 66/14  | 66/24  | 66/34  | 67/09  |
| L0P1 000777  | 28/20  | 28/37  | 28/40  | 28/09  | 28/49  | 100/55 | 42/17  | 68/14  | 69/22  | 69/29  | 69/40  | 68/14  | 69/22  | 69/29  | 69/40  | 68/14  | 69/22  | 69/29  | 69/40  | 68/14  | 69/22  | 69/29  | 69/40  |
| L0P2 001027  | 28/06  | 28/40  | 28/40  | 28/09  | 28/49  | 100/55 | 42/17  | 71/36  | 71/45  | 72/20  | 72/27  | 71/36  | 71/45  | 72/20  | 72/27  | 71/36  | 71/45  | 72/20  | 72/27  | 71/36  | 71/45  | 72/20  | 72/27  |
| L0P3 001031  | 28/57  | 28/04  | 28/04  | 28/09  | 28/49  | 100/55 | 42/17  | 73/36  | 73/45  | 75/25  | 75/36  | 73/36  | 73/45  | 75/25  | 75/36  | 73/36  | 73/45  | 75/25  | 75/36  | 73/36  | 73/45  | 75/25  | 75/36  |
| L0P4 000213  | 19/01  | 92/55  | 94/56  | 99/12  | 38/24  | 40/17  | 51/17  | 79/18  | 79/25  | 79/36  | 80/18  | 79/18  | 79/25  | 79/36  | 80/18  | 79/18  | 79/25  | 79/36  | 80/18  | 79/18  | 79/25  | 79/36  | 80/18  |
| L0P5 000213  | 24/36  | 26/41  | 31/36  | 32/28  | 38/24  | 40/17  | 51/17  | 82/25  | 82/36  | 84/18  | 84/25  | 82/25  | 82/36  | 84/18  | 84/25  | 82/25  | 82/36  | 84/18  | 84/25  | 82/25  | 82/36  | 84/18  | 84/25  |
| L0P6 001131  | 45/18  | 46/17  | 47/17  | 48/17  | 49/18  | 50/17  | 61/17  | 85/39  | 87/01  | 88/20  | 88/27  | 85/39  | 87/01  | 88/20  | 88/27  | 85/39  | 87/01  | 88/20  | 88/27  | 85/39  | 87/01  | 88/20  | 88/27  |
| L0P7 000314  | 52/17  | 53/18  | 55/17  | 58/17  | 59/13  | 60/18  | 61/17  | 90/27  | 90/38  | 90/60  | 92/27  | 90/27  | 90/38  | 90/60  | 92/27  | 90/27  | 90/38  | 90/60  | 92/27  | 90/27  | 90/38  | 90/60  | 92/27  |
|              | 62/17  | 63/17  | 64/18  | 65/17  | 66/17  | 67/17  | 69/17  | 94/21  | 94/28  | 94/39  | 95/01  | 94/21  | 94/28  | 94/39  | 95/01  | 94/21  | 94/28  | 94/39  | 95/01  | 94/21  | 94/28  | 94/39  | 95/01  |
|              | 69/31  | 69/42  | 72/15  | 72/29  | 72/40  | 75/13  | 75/27  | 96/36  | 96/60  | 97/15  | 98/20  | 96/36  | 96/60  | 97/15  | 98/20  | 96/36  | 96/60  | 97/15  | 98/20  | 96/36  | 96/60  | 97/15  | 98/20  |
|              | 75/38  | 84/13  | 84/27  | 84/38  | 86/16  | 86/30  | 86/41  | 99/15  | 100/29 | 100/27 | 100/38 | 99/15  | 100/29 | 100/27 | 100/38 | 99/15  | 100/29 | 100/27 | 100/38 | 99/15  | 100/29 | 100/27 | 100/38 |
|              | 90/15  | 90/29  | 90/40  | 92/56  | 94/57  | 98/15  | 98/29  | 105/25 | 102/36 | 103/18 | 103/25 | 105/25 | 102/36 | 103/18 | 103/25 | 105/25 | 102/36 | 103/18 | 103/25 | 105/25 | 102/36 | 103/18 | 103/25 |
|              | 98/40  | 99/12  | 100/56 | 103/13 | 103/27 | 103/38 | 105/13 | 105/27 | 105/38 | 105/54 | 111/40 | 105/27 | 105/38 | 105/54 | 111/40 | 105/27 | 105/38 | 105/54 | 111/40 | 105/27 | 105/38 | 105/54 | 111/40 |
|              | 105/27 | 105/38 | 105/54 | 105/27 | 105/38 | 105/54 | 111/40 | 31/08  | 31/22  | 31/29  | 31/29  | 31/08  | 31/22  | 31/29  | 31/29  | 31/08  | 31/22  | 31/29  | 31/29  | 31/08  | 31/22  | 31/29  | 31/29  |
|              | 23/04  | 25/44  | 26/35  | 31/14  | 31/29  |        |        | 31/25  | 31/32  | 31/32  | 31/32  | 31/25  | 31/32  | 31/32  | 31/25  | 31/25  | 31/32  | 31/32  | 31/32  | 31/25  | 31/32  | 31/32  | 31/32  |
|              | 22/23  | 25/16  | 26/06  | 30/46  |        |        |        | 31/17  | 31/42  | 26/33  | 27/14  | 31/17  | 31/42  | 26/33  | 27/14  | 31/17  | 31/42  | 26/33  | 27/14  | 31/17  | 31/42  | 26/33  | 27/14  |
|              | 26/05  | 36/47  |        |        |        |        |        | 22/58  | 24/06  |        |        | 22/58  | 24/06  |        |        | 22/58  | 24/06  |        |        | 22/58  | 24/06  |        |        |
|              | 23/28  | 33/44  | 33/53  |        |        |        |        | 31/32  |        | 34/52  | 40/16  | 31/32  |        | 34/52  | 40/16  | 31/32  |        | 34/52  | 40/16  | 31/32  |        | 34/52  | 40/16  |
|              | 23/09  | 33/43  |        |        |        |        |        | 34/16  | 34/20  | 34/52  | 40/16  | 34/16  | 34/20  | 34/52  | 40/16  | 34/16  | 34/20  | 34/52  | 40/16  | 34/16  | 34/20  | 34/52  | 40/16  |
|              | 33/46  | 33/55  | 26/34  | 30/36  | 31/15  | 72/24  | 72/24  | 48/12  | 48/12  | 48/12  | 48/12  | 48/12  | 48/12  | 48/12  | 48/12  | 48/12  | 48/12  | 48/12  | 48/12  | 48/12  | 48/12  | 48/12  | 48/12  |
|              | 23/46  | 25/42  | 26/34  | 30/36  | 31/15  | 72/24  | 72/24  | 48/16  | 48/16  | 48/16  | 48/16  | 48/16  | 48/16  | 48/16  | 48/16  | 48/16  | 48/16  | 48/16  | 48/16  | 48/16  | 48/16  | 48/16  | 48/16  |
|              | 73/16  | 73/22  | 75/16  | 75/22  | 77/16  | 77/22  | 79/16  | 50/36  | 51/16  | 51/26  | 51/26  | 50/36  | 51/16  | 51/26  | 51/26  | 50/36  | 51/16  | 51/26  | 51/26  | 50/36  | 51/16  | 51/26  | 51/26  |
|              | 79/22  | 80/16  | 80/22  | 82/16  | 82/22  | 84/16  | 84/22  | 53/17  | 54/12  | 55/16  | 55/16  | 53/17  | 54/12  | 55/16  | 55/16  | 53/17  | 54/12  | 55/16  | 55/16  | 53/17  | 54/12  | 55/16  | 55/16  |
|              | 86/19  | 86/25  | 88/18  | 88/24  | 90/18  | 90/24  | 92/18  | 60/17  | 60/27  | 61/16  | 61/26  | 60/17  | 60/27  | 61/16  | 61/26  | 60/17  | 60/27  | 61/16  | 61/26  | 60/17  | 60/27  | 61/16  | 61/26  |
|              | 92/24  | 94/19  | 94/25  | 96/18  | 96/24  | 98/18  | 98/24  | 63/26  | 64/17  | 64/27  | 64/37  | 63/26  | 64/17  | 64/27  | 64/37  | 63/26  | 64/17  | 64/27  | 64/37  | 63/26  | 64/17  | 64/27  | 64/37  |
|              | 100/18 | 100/24 | 102/16 | 102/22 | 103/16 | 103/22 | 105/16 | 66/16  | 66/26  | 66/36  | 67/16  | 66/16  | 66/26  | 66/36  | 67/16  | 66/16  | 66/26  | 66/36  | 67/16  | 66/16  | 66/26  | 66/36  | 67/16  |
|              | 105/22 | 25/28  | 25/36  |        |        |        |        | 68/16  | 68/26  | 68/36  | 69/16  | 68/16  | 68/26  | 68/36  | 69/16  | 68/16  | 68/26  | 68/36  | 69/16  | 68/16  | 68/26  | 68/36  | 69/16  |
|              | 25/21  | 25/28  |        |        |        |        |        | 71/20  | 72/22  | 73/20  | 75/20  | 71/20  | 72/22  | 73/20  | 75/20  | 71/20  | 72/22  | 73/20  | 75/20  | 71/20  | 72/22  | 73/20  | 75/20  |
|              | 22/47  | 25/12  |        |        |        |        |        | 82/20  | 84/20  | 86/23  | 87/03  | 82/20  | 84/20  | 86/23  | 87/03  | 82/20  | 84/20  | 86/23  | 87/03  | 82/20  | 84/20  | 86/23  | 87/03  |
|              | 22/29  |        |        |        |        |        |        | 91/02  | 92/22  | 93/02  | 94/23  | 91/02  | 92/22  | 93/02  | 94/23  | 91/02  | 92/22  | 93/02  | 94/23  | 91/02  | 92/22  | 93/02  | 94/23  |

0123 ECL32

66/13 66/23 66/33 67/13 67/23 67/33 69/21  
71/17 72/19 73/17 75/17 77/17 79/17 80/17  
82/17 84/17 86/20 88/19 88/59 90/19  
90/59 92/19 92/59 94/20 94/50 95/16 96/19  
96/59 97/15 98/19 98/59 99/15 100/19 100/59  
101/15 102/17 103/17 105/17 105/53

RNDMA 001235  
RTAN 006613  
SETSW 001541  
SETUP 001207 MC

26/20 37/01 39/05 40/06 41/06 42/06 43/06  
44/05 45/07 46/06 47/06 48/06 49/07 50/06  
51/06 52/06 53/10 54/05 55/06 56/06 57/06  
58/06 59/05 60/07 61/06 62/06 63/06 64/07  
65/06 66/06 67/06 68/09 69/14 71/10 72/12  
73/10 75/10 77/10 80/18 82/10 84/10  
86/13 88/12 90/12 92/12 94/13 96/12 98/12  
100/12 102/10 103/10 105/10 107/05 108/05 109/05  
110/05

SIZE 001156  
START 000727  
STLOC 000675  
SWMES 000700  
SMREG 006614  
TBLSI 000262

23/10 30/36 30/47 36/28 26/30  
26/17 25/42 26/04 28/16 28/45 28/33 28/24 28/26 28/31  
23/24 73/24 73/24 73/24 73/24 73/24 73/24 73/24 73/24  
79/29 80/24 80/24 80/24 80/24 80/24 80/24 80/24 80/24  
86/27 86/32 88/26 88/31 90/26 90/31 92/26 92/26 92/26  
92/31 94/27 94/32 94/32 96/26 96/31 98/26 98/31 98/31  
100/26 100/31 102/24 102/29 103/24 103/29 105/24  
105/29

TSTLO 001112 MC

17/02 38/13 39/02 40/03 41/03 42/03 43/03  
44/02 45/04 46/03 47/03 48/03 49/04 50/03  
51/03 52/03 53/07 54/02 55/03 56/03 57/03  
58/03 59/02 60/04 61/03 62/03 63/03 64/04  
65/03 66/03 67/03 68/06 69/07 71/03 72/05  
73/03 75/03 77/03 79/03 80/03 82/03 84/03  
86/06 88/05 90/05 92/05 94/06 96/05 98/05  
100/05 102/03 103/03 105/03 107/02 108/02 109/02  
110/02

TSTTO 000234  
TTHOU 000245  
-EGGS 000010-  
-RAND 001126





LISTING

096-000251-04

PROGRAM

ECLIPSE EXTENDED MEMORY  
EXERCISER

TAPE

095-000247-04

ABSTRACT

THE ECLIPSE EXTENDED MEMORY EXERCISER IS INTENDED TO AUGMENT THE EXISTENCE OF THE OTHER MEMORY CHECKERBOARD TESTS. THIS CHECKERBOARD, HOWEVER, TAKES INTO CONSIDERATION THE EXISTENCE OF SUCH ITEMS AS THE MMPU AND INTERLEAVED MEMORIES. IT ALSO INCLUDES SPECIAL PATTERNS THAT ALLOW FOR WORST CASE PATTERNS IN THE 5 ERROR CORRECTION BITS IN THE ERCC MEMORIES.



0001 EXMEM MACRO REV 03.00 14129115 12/31/76

10002 EXMEM

```

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

```

: \*\*\*\*\*  
: NAME: EXMEM.SR PART NUMBER: 094-000648  
: \*\*\*\*\*  
: DESCRIPTION: ECLIPSE EXTENDED MEMORY EXERCISER  
: \*\*\*\*\*  
: REVISION HISTORY:  
: REV. DATE  
: 00 12/20/74  
: 01 04/11/75  
: 02 06/06/75  
: 03 08/06/76  
: 04 12/31/76  
: \*\*\*\*\*  
: COPYRIGHT (C) DATA GENERAL CORPORATION, 1974, 1975, 1976  
: ALL RIGHTS RESERVED.  
: \*\*\*\*\*

: ECLIPSE FAMILY OF COMPUTERS  
: EXTENDED MEMORY EXERCISER  
  
: ABSTRACT  
: THE ECLIPSE EXTENDED MEMORY EXERCISER IS  
: INTENDED TO AUGMENT THE EXISTENCE OF THE  
: OTHER MEMORY CHECKER BOARD TESTS.  
  
: THIS TEST WAS DESIGNED TO TEST 8K MEMORY  
: BOARDS BUT WILL ALSO HANDLE 16K SPLIT  
: SENSE MEMORIES.  
  
: THIS CHECKER BOARD TAKES INTO  
: CONSIDERATION THE EXISTENCE OF SUCH  
: ITEMS AS THE MMPU AND INTERLEAVED  
: MEMORIES. IT ALSO INCLUDES SPECIAL  
: PATTERNS THAT ALLOW FOR WORST CASE  
: PATTERNS IN THE 5 ERROR CORRECTION  
: BITS IN THE ERCC MEMORIES  
  
: MACHINE REQUIREMENTS  
: ECLIPSE PROCESSOR  
: 8K READ WRITE CORE MEMORY  
: (UP TO 128K WITH MMPU)  
: (UP TO 256K WITH MMPU)  
: 4010 CONTROL AND CONSOLE TELETYPE  
: OPTIONAL EQUIPMENT  
: RTC  
: 12.4.1  
: 12.4.2 ERROR CORRECTION MEMORIES

10003 EXMEM

01 SWITCH SETTINGS  
 02 STARTING ADDRESSES  
 03 200 RUN ALL AVAILABLE OPTIONS  
 04 202 ALLOW FOR DELETION OF OPTIONS  
 05 204 IGNORE MPU AND ONLY RUN UP  
 06 TO THE FIRST 32K OF MEMORY  
 07  
 08 SWITCH/KEY PACKAGE  
 09  
 10 KEY 0 PLACES PGM IN KEY ENTRY MODE.  
 11 ALLOWS SETTING OF OPTIONS FROM KEYBOARD.  
 12 TYPING ANY KEY FROM 1-7,A-F COMPLEMENTS  
 13 THE PREVIOUS STATE OF THE BIT IN SWREG.  
 14  
 15 TYPE A CR TO EXIT KEVENTRY MODE.  
 16  
 17  
 18 NOTE: THE ABOVE MANNER OF USING KEVENTRY ALLOWS  
 19 FOR THE ADDRESS MATCH FEATURE TO BE  
 20 UTILIZED IN DEBUGGING AS A SCOPE SYNC.  
 21  
 22 SW2=1 DELETE T10 TYPE OUTS  
 23 SW5=1 ENABLE OUTUP TO LPT  
 24  
 25 KEY 7 TYPING A 7 WILL PRINT THE  
 26 RUN TIME AND ACCUMULATED ERRORS  
 27  
 28 SW15=1 RING BELL ON ERROR  
 29  
 30 KEY M LIST THE CURRENT SWITCH REGISTER BITS  
 31  
 32 (C)D SETS SWREG TO DEFAULT MODE  
 33 AND RESTARTS THE PROGRAM  
 34  
 35 (C)R RESTARTS PROGRAM WITHOUT DISTURBING SWREG  
 36  
 (C) = CONTROL KEY

10004 EXMEM

01  
 02  
 03  
 04  
 05  
 06  
 07  
 08  
 09  
 10  
 11  
 12  
 13  
 14  
 15  
 16  
 17  
 18  
 19  
 20  
 21  
 22  
 23  
 24  
 25  
 26  
 27  
 28  
 29  
 30  
 31  
 32  
 33  
 34  
 35  
 36  
 37  
 38  
 39  
 40  
 41  
 42  
 43  
 44  
 45  
 46  
 47  
 48  
 49  
 50

OPERATING PROCEDURE  
 START AT ADDR. 200 TO RUN ALL OPTIONS  
 START AT ADDR. 202 IF YOU WANT TO DELETE  
 ERCC OR THE RTC  
 START AT ADDR. 204 IF YOU WANT TO RUN  
 WITHOUT THE MPU

THE TEST WILL THEN SIZE MEMORY AND THEN  
 RUN A SERIES OF TIMING TESTS TO DETERMINE  
 THE INTERLEAVING FACTORS OF THE AVAILABLE  
 MEMORY.

THE ENSUING TYPEOUT WILL INDICATE THE FOLLOWING:  
 TOTAL AMOUNT OF MEMORY FOUND  
 WHETHER OR NOT AN MPU OR MPU1 EXISTS

AFTER INDICATING THE EXISTANCE AND TYPE OF MPU  
 THE TEST WILL PROCEED TO TYPE AN INDICATION OF THE  
 TYPES OF MEMORIES AVAILABLE AND THEIR INTERLEAVE  
 FACTORS. THE TEST INDICATES THE EXISTENCE OF MEMORY  
 OF A SPECIFIC TYPE AND INTERLEAVING AND FOLLOWS ON  
 THE SAME LINE WITH EITHER ONE OR TWO 16 BIT CONSTANTS.  
 THE FIRST CONSTANT INDICATES EXISTENCE OF THAT TYPE  
 OF MEMORY WITHIN THE FIRST 128K AND THE 2ND INDICATES  
 THE EXISTENCE OF THAT TYPE OF MEMORY WITHIN THE 2ND  
 128K. IF THERE IS NO MEMORY OF THAT TYPE ABOVE 128K,  
 THE 2ND CONSTANT IS NOT TYPED.

I.E. IF A SYSTEM HAD THE FIRST 64K 4 WAY INTERLEAVED 16K CORE  
 THE NEXT 64K 2 WAY INTERLEAVED 32K SC AND THE UPPER  
 128K 8 WAY INTERLEAVED 32K SC'S THE TYPEOUT WOULD APPEAR  
 AS BELOW:

TYPE INTLF 128K 256K  
 CORE 4 WAY 177400  
 SC 2 WAY 000377  
 SC 8 WAY 000000 177777

NOTE:  
 THE TEST NO LONGER ATTEMPTS TO IDENTIFY THE INTERLEAVING  
 OF THE 8K CACHED SC MEMORIES. IT DOES HOWEVER,  
 IDENTIFY THEIR EXISTENCE. IF THE UPPER 64K OF A 128K SYSTEM  
 WAS 8K CACHED SC'S THE FOLLOWING TYPEOUT WOULD BE INCLUDED  
 NO MATTER HOW THE MEMORIES WERE INTERLEAVED:

8K SC 000377



10005 EXMEM

```

01 TEST EXECUTION
02 ALL MEMORY UNDER TEST IS FILLED WITH AN INHIBIT PATTERN
03 (OOOE VIA A 8LM )
04 THE WORST CASE CHECKERBOARD IS CREATED BY FILLING IN
05 THOSE AREAS THAT SHOULD NOT CONTAIN THE INHIBIT PAT.
06 WITH THE NON INHIBIT PATTERN
07
08 MEMORY IS THEN EXERCISED IN THE FOLLOWING MANNER(FOR 2 PASSES)
09 EACH GROUP OF 16 WORDS CHECKED TO SEE IF IT CONTAINS THE KEY
10 PATTERN.(IF NOT EACH WORD IS READ AND COMPLIMENTED TO IT)
11 EACH GROUP OF 16 WORDS IS THEN SHUFFLED 16 TIMES
12 (WORD 1 IS MOVED TO WORD 2 IS MOVED TO WORD 3 ETC.
13 UNTIL WORD 15 IS MOVED TO 16 AND 16 BACK TO 0)
14 NOTE: THE SHUFFLE LOOP "DISLP" IS MODIFIED TO
15 ACCOMODATE THE DIFFERENT INTERLEAVES
16 SO THAT DISTURB NOISE WILL BE CONCEN-
17 TRATED WITHIN EACH PHYSICAL 8K MODUAL.
18 ALSO, NONE OF THESE SHUFFLE LOOPS
19 ARE EXECUTED THE INITIAL PASS 0 OF.
20 EXMEMX FOR MEMORIES THAT ARE
21 OF A DIFFERENT INTERLEAVING THAN
22 THE LOOP.
23
24 THE THIRD PASS THROUGH MEMORY FOR EACH PATTERN,
25 THE SHUFFLE IS EXTENDED IF THE LOOP IS
26 CORRECTLY MODIFIED FOR INTERLEAVING.
27 BIT 15 OF THE FIRST WORD OF THE 16 IS COMPLIMENTED
28 THE 16 WORDS ARE THEN "SHUFFLED"
29 AND THE WORD IS CHECKED TO SEE IF IT STILL EQUALS THE
30 INHIBIT PATTERN.
31 THE ABOVE PROCESS IS THEN REPEATED WITH BIT 14 OF THE 2ND
32 WORD, 13 OF THE THIRD WORD, 12 OF THE 4TH WORD, ETC.
33 UNTIL BIT 0 OF THE 16TH WORD HAS BEEN COMPLIMENTED
34 SHUFFLED RECOMPLIMENTED AND CHECKED.
35 THE ENTIRE 16 WORDS ARE THEN RECOMPLIMENTED TO THE NONINHIBIT
36 PATTERN IF THEY WERE ORIGINALLY IN THAT STATE.
37 THE SAME PROCEDURE IS REPEATED FOR THE NEXT 16 WORDS
38 UNTIL THE END OF MEMORY.
39
40 THE ENTIRE MEMORY IS THEN CHECKED TO SEE IF IT CONTAINS
41 THE ORIGINAL WORST CASE PATTERN.
42
43 MEMORY IS THEN FILLED WITH THE COMPLIMENT WORST
44 CASE CHECKERBOARD AND THE WHOLE COMP -SHUFFLE-
45 PRECOMP. SEQUENCE IS AGAIN EXERCISED FOR 3
46 PASSES THROUGH MEMORY.
47

```

10006 EXMEM

```

01 TEST EXECUTION(CONTINUED)
02
03 THE ABOVE SEQUENCE OF PATTERN EXERCISING IS
04 THEN REPEATED FOR THE NEXT INTERLEAVING LEAF.
05 NEXT PATTERN PAIR IF UNINTERLEAVED
06 2 TIMES IF 2 WAY, 4 IF 4 WAY AND
07 8 TIMES IF 8WAY
08
09 THE TEST THEN PROCEEDS TO THE NEXT INHIBIT/NON INHIBIT
10 PAIR IF IT IS A 21 BIT MEMORY
11 IF A REGULAR 16 BIT MEM. OR BOTH ERCC PATTERNS HAVE BEEN RUN
12 THE TEST THEN PROCEEDS TO THE NEXT
13 INTERLEAVE SEQUENCE
14 AFTER ALL PATTERNS, THEIR COMPLIMENT PATTERNS
15 HAVE BEEN EXERCISED FOR ALL COMBINATIONS OF
16 INTERLEAVING THE PROGRAM WILL TYPE PASS
17
18 5.1 SUMMARY
19
20 TO SUMMARIZE THE SEQUENCE:
21 CB.TK= 0 FILL WITH KEY PATTERN
22 1 CREATE WORST CASE PAT
23 2 FAST SHUFFLE
24 3 FAST SHUFFLE
25 4 SLOW SHUFFLE
26 5 COMPARE MEMORY
27 6 FILL WITH KEY PATTERN
28 7 COMPLIMENT PAT
29 10 CONTINUE FILL COMPL. WORST CASE
30 11 FAST SHUFFLE
31 12 SLOW SHUFFLE
32 13 SLOW SHUFFLE
33 14 COMPARE MEMORY
34
35 THE ABOVE 15 OPERATIONS WILL BE REPEATED
36 FOR EACH PATTERN
37 INTLS= 1 ONCE(UNINTERLEAVED CORE TEST)
38 2 FOR INTL=0 AND 1 (2WAY CORE)
39 4 FOR INTL=0,1,2 AND 3 (4 WAY CORE)
40 10 FOR INTL=0 TO 7 (8 WAY CORE)
41
42 NOTE: FOR CB.TK= 4 OR 14 A FAST SHUFFLE OCCURS
43 IF INTLS DOES NOT AGREE WITH THE
44 INTERLEAVING OF THE 8K BEING EXERCISED
45 AFTER INTLS=INTLF STEP TO NEXT PATTERN
46 AND RESET INTLF=0
47 AFTER ALL PATTERNS STEP TO NEXT INTLS
48 MODIFY THE SHUFFLE LOOP AND RESTART PATTERNS.
49
50 I.E. IF INTLS IS =10 AND INTLF IS =5,
51 THEN, THE 8 WAY INTERLEAVE SHUFFLE IS
52 OCCURING ON ALL ADDRESSES THAT END
53 IN THE DIGIT 5.

```

10007 EXMEM

```

01 16.0
02 ERRORS
03 ALL LOGICAL ERRORS WILL CAUSE AN ERROR
04 TYPEOUT. SOME UNEXPECTED ERRORS WILL
05 CAUSE A HALT TO OCCUR.
16.1
06 DATA FAILURE TYPEOUTS
07 THE CHECKER BOARD TEST ITSELF INITIATES
08 TYPEOUTS WHEN IT DETECTS DATA FAILURES
09 THE INITIAL TYPE OUT IN A TEST LOOP
10 THAT FAILS WILL INCLUDE ALL 4 AC'S
11 TYPED ON ONE LINE.
12 AC0= DATA EXPECTED
13 AC1= INCORRECT DATA
14 AC2= LOGICAL ADDRESS OF FAILURE
15 AC3= ADDRESS+1 OF JSR @ERR!
16 IN ADDITION THE FOLLOWING IS TYPED
17 CB.TK= 0 TO 14 (SEE TEST SEQUENCE)
18 INTLS= 1,2,4 OR 10 (INDICATES THE
19 CURRENT INTERLEAVING
20 SEQUENCE IN THE SHUFFLE)
21 INTLF= 0 TO 7 (INDICATES THE LOW
22 ORDER BITS OF THE ADDRESSES
23 BEING SHUFFLED.)
24 I.E. IF INTLF=2/INTLS=1
25 THEN ONLY THE ODD ADDRESSES
26 WERE BEING SHUFFLED.
27 IN ADDITION, THE NEXT THREE FAILURES(DATA)
28 THAT OCCUR WILL BE TYPED
29 GOOD/BAD/LOGICAL ADDRESS
30
31 16.2
32 ERROR CORRECTION INTERRUPTS
33 IF ERCC IS ENABLED THE TYPE OUTS THAT
34 OCCUR WILL MORE LOGICALLY BE DUE TO
35 ERROR CORRECTION INTERRUPTS THAN
36 TO DATA FAILURES DETECTED BY
37 ANY OF THE GOOD/BAD COMPARES
38 AGAIN, THE INITIAL TYPEOUT WILL CONSIST
39 OF THE FOUR AC'S:
40 AC0= DATA WORD AFTER CORRECTION
41 AC1= 018 FROM ERCC OPTION
42 BITS 0 TO 4 ARE CORRECTION CODE
43 BIT 15 INDICATES LOWER 64K IF=0
44 UPPER 64K IF=1
45 AC2= PHYSICAL ADDRESS OF THE ERROR
46 (WITHIN 64K INDICATED)
47 AC3= ADDRESS(PC) OF THE INTERRUPT
48 (REFER TO LISTING)
49 CB.TK/INTLS/INTLF ARE INCLUDED
50 AND TYPED
51 THE CORRECTION CODE IS RIGHT JUSTIFIED
52 THE TOTAL NUMBER OF ERROR CORRECTION
53 INTERRUPTS THAT OCCURRED DURING THE LAST
54 TEST LOOP ARE TYPED

```

10008 EXMEM

```

01 16.3
02 POWER FAIL
03 IF A POWER FAILURE OCCURS WHILE THIS
04 TEST IS BEING EXECUTED, AND AUTO-
05 RESTART IS ENABLED, THE TEST WILL
06 SAVE ENOUGH INFORMATION TO RESTART
07 AT THE POINT THAT THE POWER FAIL
08 OCCURRED
09 ANY SC MEMORY FAILURES SHOULD BE
10 IGNORED IMMEDIATELY FOLLOWING
11 A POWER FAIL RESTART.
12
13 ERROR ANALYSIS
14 IN AN UNMAPPED/UNINTERLEAVED SYSTEM
15 THE ACTUAL MEMORY LOCATION(S) FAILING
16 WILL PROBABLY BE OBVIOUS. THE ERROR
17 TYPEOUTS WILL POINT DIRECTLY TO IT.
18
19 IN THE MORE COMPLEXLY ORGANIZED SYSTEMS
20 IT WILL PROBABLY REQUIRE A SERIES OF
21 TYPEOUTS BEFORE THE ACTUAL PHYSICAL MEM-
22 ORY THAT IS FAILING WILL BE ISOLATED.
23 (THE TEST WILL MAP MEMORY TO THE LOGICAL
24 ADDRESS THAT MOST CLOSELY RESEMBLES
25 THE PHYSICAL ADDRESS. THEREFORE, THE FIRST
26 32K WILL BE MAPPED LOGICAL TO PHYSICAL)
27
28 PAY PARTICULAR ATTENTION TO THE FOLLOWING:
29 HOW IS THE MEMORY INTERLEAVED?
30 WHAT INTLS/INTLF SEQUENCES FAIL THE MOST
31 FREQUENTLY.
32 (SEE TYPEOUTS AT INITIALIZATION)
33 IF A. THE MEMORY IS 8 WAY INTERLEAVED
34 B. INTLS=2/INTLF=1 FAILS
35 (AN ODD# MEMORY IS FAILING)
36 C. INTLS=10/INTLF=3 FAILS IN ADDITION TO B.
37 D. ONE CAN FAIRLY SAFELY ASSUME THAT
38 IT IS THE "3" MEMORY OF THE 8
39 THAT IS FAILING
40 TO SUPPORT THIS CONCLUSION HOWEVER, YOU
41 SHOULD ALSO ASCERTAIN THAT THE MAJORITY OF
42 ADDRESSES THAT FAILED HAVE A 3 IN THE
43 LEAST SIGNIFICANT DIGIT.
44
45 AS AN ASSIST IN DETERMINING EXACTLY WHAT
46 IS WRONG WITH A SPECIFIC MEMORY PAY
47 PARTICULAR ATTENTION TO THE ADDRESS OF THE
48 ERROR CALLS(AC3 AS TYPED) OR ADDRESS OF
49 THE INTERRUPTS IF ERCC TYPE ERRORS.

```

10009 EXMEM

```

: FOR CONVENIENCE (YOURS) THE FOLLOWING TABLE
: IS INCLUDED TO HELP DETERMINE WHICH BIT
: IS FAILING IN AN ERROR CORRECTION MEMORY
: COR.CODE
: 0
: 1 NO ERROR
: 2 CHECK BIT 4
: 3 CHECK BIT 3
: 4 DATA BIT 0
: 5 CHECK BIT 2
: 6 DATA BIT 1
: 7 MULTIPLE BIT
: 8 DATA BIT 3
: 9 CHECK BIT 1
: 10 DATA BIT 4
: 11 ALL 21 BITS WERE=1
: 12 DATA BIT 6
: 13 DATA BIT 7
: 14 DATA BIT 8
: 15 DATA BIT 9
: 16 MULT. BITS FAILED
: 17 CHECK BIT 0
: 18 DATA BIT 11
: 19 DATA BIT 12
: 20 DATA BIT 13
: 21 DATA BIT 14
: 22 ALL 21 BITS READ AS 0'S
: 23 MULTIPLE BIT'S
: 24 DATA BIT 10
: 25 MULTIPLE BITS
: 26 DATA BIT 5
: 27 MULTIPLE BITS
: 28 DATA BIT 15
: 29 MULTIPLE BITS
: 30 SAME
: 31 SAME
: 32 SAME
: 33 SAME
: 34 SAME
: 35 SAME
: 36 SAME
: 37 SAME
: 38
: 39
: 40
: 41
: 42
: 43
: 44
: 45
: 46
: 47
: 48
: 49
: 50
: 51

```

10010 EXMEM

```

01
03 000003 .TITL EXMEM
04 000040 .DUSR MAP=3
05 000040 .DUSR SSP=40
06 000005 LSTAP=SSP
07 000000 MSKRG=5
08 000000 .LOC 0
09 000000 DIRT ;FOR DTOS
10 000200 .LOC 200 ;SIZE AND GO
11 002000 002201 STRT1: JMP @+1
12 002010 006600 LINKR ;SIZE AND WAIT FOR SELECTIONS
13 002020 002203 STRT2: JMP @+1
14 002030 006601 LINKR*1
15 002040 002205 STRT3: JMP @+1
16 002050 006626 LAUTO*1
17
18 ;PAGE 0 LINKS FOR MAP OPTION TRAPS
19 ;PAGE 0 LINKS FOR CALLS
20 ;ASCRA MUST BE FIRST WITH ERROR LAST
21 ;ANY CALL EXPANSION MUST BE MADE BETWEEN THE TWO
22 00206 002047 ASCRA: ASSCR ;ASSIGN SCRATCH
23 00207 002104 ESCRA: EXSCR ;EXPAND SCRATCH
24 00210 002124 RSCRA: RLSCR ;RELEASE SCRATCH
25 00211 002337 PDECI: PDE? ;DECIMAL PRINT
26 00212 002274 ERXTX: ERXT ;TEXT TYPEOUT CALL
27 00213 001377 RETRN: LRETP ;RETURN FROM TEST
28 00214 002773 EINTS: EINTP ;ENTER INTR SERVICE
29 00216 002155 ERROI: ERROD ;INIT ERROR TYPEOUTS
30 00217 000176 -USTZM: LPGO ;APPEND TO ERR TYPEOUT
31 00220 000000 -USTSS: 0
32 00221 000000 -USTES: 0
33 00222 006173 -USTNM: LSYSE
34 00223 000224 CB.TS: ;*1
35 00224 000103 ;BLM INHIBIT PAT
36 00225 000000 ;FILL IN WORST CASE
37 00226 000121 ;FAST SHUFFLE
38 00227 000121 ;FAST SHUFFLE
39 00228 000121 ;SLOWER SHUFFLE
40 00229 000117 ;COMPARE PATTERN
41 00231 000041 ;BLM INHIBIT PAT
42 00232 000103 ;COMPLIMENT WORST CASE
43 00233 000033 ;CONTIN. FILL COMP WORST CASE
44 00234 000000 ;FAST SHUFFLE
45 00235 000121 ;FAST SHUFFLE
46 00236 000121 ;SLOW SHUFFLE
47 00237 000117 ;SLOW SHUFFLE
48 00240 000041 ;COMPARE PATTERN
49 00241 177777 ;

```



```

10013 EXMEM
01 00107 000000 ;CHECKER BOARD RELOCATE SWITCH
02 00110 000007 CMOVAL: 0 ;CURRENT MEMORY MODUAL
03 00111 177777 MDSIZ: 7 ;17 FOR 8K 17 FOR 16K 37 FOR 32K
04 00112 000000 CB.LK: 0
05 00113 000000 CB.ES: 0
06 00114 000000 CB.PX: 0
07 00115 000000 PATTB: 0
08 00116 000000 PATTB: 0
09 00117 000000 MPA36: 0
10 00120 000000 INHPR: 0
11 00121 000000 XORIP: 0
12 00122 000000 CB.37: 0
13 00123 004444 CM.BG: BEGIN
14 00124 000125 EPROG: DIRET
15 00125 000000 DIRET: 0
16 00126 000000 DIXCT: 0
17 00127 000000 DIXRT: 0
18 00130 000000 DIXNX: 0
19 00131 000000 DIXST: 0
20 00132 000000 DIXBT: 0
21 00133 000000 CB.EW: 0
22 00134 000000 CB.LC: 0
23 00135 000000 CB.II: 0
24 00136 000000 DTGL: 0
25 00140 000000 RETURN: 0
26 00141 000400 C400: 400
27 00142 000017 C17: 17
28 00143 000020 C20: 20
29 00144 000077 C77: 77
30 00145 001777 C1777: 1777
31 00146 177761 MINIS: -15-
32 00147 000101 C101: 101
33 00150 000136 CB.WK: TOTL
34 00151 004340 ERR1: CB.ER
35 00152 000000 PONES: 0
36 00153 004444 BPROG: BEGIN
37 00154 003747 INTL1: INTLO ;PAGE 0 FIXED LOCATIONS FOR INTERLEAVING CONTROL
38 00155 003747 INTL2: INTLO ;POINTS TO FIRST INTERLEAVING TABLE
39 00156 000005 CB.LK: INTL2-INTLO ;AND TO NEXT USED TABLE
40 00157 000169 INT90: INT90+1 ;TABLE LENGTH
41 00160 000017 OPADK: 17 ;CONSTANT FOR MOVE
42 00161 000000 AC2MD: 0 ;% FOR HIGHEST 17,56,154 OR 350
43 00162 007770 IMTBL: INST0+1B14 ;200 FOR 8 WAY=0 FOR ALL OTHERS
44 00163 010070 OPADI: RUMPO+1B14 ;BYTE PTR. TO INSTR. MOD TABLE
45 00164 000001 INTLS: 1 ;DITTO FOR ADPS. SEQUENCE
46 00165 000001 INTLF: 1 ;STOP EXER. WHEN INTLF=1,2,4 OR 10
47 00166 000000 WMDU: 0 ;CURRENT INTERLEAVING OFFSET
48 00167 000000 WMDI: 0 ;SCRATCH AREA SIZE PARAMETER LOCS FOR TEST USAGE
49 00170 000000 CURPR: 0 ;1ST 1K MODUAL#
50 00171 000000 SCRLO: 0 ;THIS 8K INTERLEAVE FACTOR
51 00172 000000 SCRHI: 0 ;CURRENT PROGRAM#
52 00173 000173 LZMAX=SCRHI+1 ;LOWEST LOGICAL SCRATCH ADPS
53 000173 LPG0=0 ;HIGHEST LOGICAL SCRATCH ADPS

```

```

10014 FXMEM
01 000004 .DUSR ISP=4
02 000006 .DUSR ISL=6
03 000007 .DUSR ISF=7
04 000005 .DUSR MK=5
05 000040 .DUSR USTKP=40
06
07
08
09
10
11 00600 102401 ;LOC 600
12 00601 102600 ;LINKER MAIN LINE DISPATCH ROUTINE
13 00602 126400 ;AUTO START ENTRY
14 00603 044062 SUB 1,1 ;MANUAL SELECT ENTRY
15 00604 040421 STA 1,USES# ;SET USE MAP SW
16 00605 102000 STA 0-LAUTO ;SET ENTRY TYPE SW
17 00606 040075 STA 0,PFAIS
18 00607 064407 JSR LOSPR ;START DISPATCH
19 00610 000631 LTR1 ;THROUGH INIT SEQ
20 00611 102520 SUBCL 0,0
21 00612 040170 STA 0,EURPR
22 00613 060177 INTEN
23 00614 004402 LRUNS: JSR LDISPR
24 00615 000647 LTR2 ;THROUGH RUN TABLE
25
26 ;DISPATCH ROUTINE
27 ;ENTER SUBROUTINES IN SEQ VIA TABLE SPEC BY (R3)
28 ;END OF EACH TABLE IS LRUNS WHICH WILL START US
29 ;BACK AT THE BEGINNING OF THE RUN TABLE
30 ;ADRS OF DISPATCH TABLE
31 LDISPR: LDA 0,0,3
32 STA 0,LDIS
33 LDA 3,ALDIS
34 JSR 0,3
35 JSZ LDIS
36 JMP LDS.1
37
38 LDIS: 0
39 LAUTO: 0
40 ;SET SWITCH SO MAP WILL NOT BE USED IF IT EXISTS
41 ADC 0,0
42 STA 0,USES#
43 JMP LINKR.4
44 ;DEFINE SYSTEM MACROS FOR INDIVIDUAL TESTS
45 .MACRO LCALL
46 JSR 0,1
47
48 .MACRO NEXTT
49 .LOC LPG0
50 .1
51 LPG0=.
52 .LOC LMEML
53 0
54 ;INTERRUPT WAIT TIME
55
56
57
58

```

!0015 EXMEM

01
02
03
04 00631 000655
05 00632 001061
06 00633 000773
07 00634 003004
08 00635 001074
09 00636 001217
10 00637 001262
11 00640 003160
12 00641 001231
13 00642 005230
14 00643 003174
15 00644 000611
16 00645 000000
17 00646 000000
18
19
20
21 00647 001313
22 00650 001556
23 00651 001354
24 00652 000614
25 00653 000000
26 00654 000000
27

!LTBL1=INIT SYSTEM DISPATCH TABLE
!END OF TABLE IS LRUNS
LTBL1:
LVSRR !RESET SYS
GPACK !DEI # TESTS LOADED
LWSET !SET UP RUNNING CONS.
LCINT !INIT INTR VECTORS
LSIZE !SIZE MEMORY
MVETA !MOVE EXISM TO AVALM
YINIT !INIT EACH TEST LOADED
NEWIO !SET UP FOR VECTORS
U8L32 !REMOVE USED 1K'S IN 32
JLPT !DETERMINE MEM INTERLEAVING
LPRSL !LISTS TESTS TO BE RUN
LRUNS-3
0
0

!LTBL2=RUN SYS DISPATCH TABLE
LTBL2:
LRANP !PRGG SELECT
LDMP !LOAD MAP OPTION
LSTRP !START TEST RUNNING
LRUNS
0
0

!0016 EXMEM

01
02
03
04
05
06 00655 062677
07 00656 102400
08 00657 040117
09 00660 040107
10 00661 040061
11 00662 040064
12 00663 040074
13 00664 040000
14 00665 040066
15 00666 040065
16 00667 040265
17 00670 024062
18 00671 125004
19 00672 001400
20
21 00673 062003
22 00674 026461
23 00675 125004
24 00676 001400
25 00677 12455
26 00700 020000
27 00701 046454
28 00702 100405
29 00703 001400
30 00704 040066
31 00705 024452
32 00706 066003
33
34 00707 102400
35 00710 063003
36 00711 062403
37 00712 026443
38 00713 123414
39 00714 065077
40 00715 126520
41 00716 125300
42 00717 067003
43 00720 062403
44 00721 024450
45 00722 123415
46 00723 102401
47 00724 102000
48 00725 040065
49
50 00726 101005
51 00727 000404
52 00730 102400
53 00731 040060
54 00732 001400
55 00733 054562
56 00734 102400
57 00735 024421
58 00736 044060
59 00737 024415
60 00740 030420

!LSYSR=RESET SYSTEM
!SET LOGICAL PAGE 37=PHYS 37
!SET MP5WT=0 NO MAP -1'S IF MAP OPTION
!SET PROCERRR PROTECT BIT
!0 ALL OTHER PROTECT BITS
LVSRR:
LSYSR:
SUB 0,0
STA 0,MPASK
STA 0,CHDUAL
STA 0,JUSTAT
STA 0,ECSEL
STA 0,ERTOT
STA 0,0
STA 0,MP5WT
STA 0,NEWMP
STA 0,CUR1K
STA 0,USESM
LDA 1,1,5ZR
MOV 1,1,5ZR
JMP 0,3
!NOI
!CHECK IF A MAP EXISTS
DO5 0,MAP
LDA 1,RLS,K2
MOV 1,1,5ZR
JMP 0,3
ISZ @LS,K2
LDA 0,0
STA 1,RLS,K2
NEG 0,0,SNR
JMP 0,3
STA 0,MP5WT
LLOMP: LDA 1,LS,K3
DO8 1,MAP
!MAP 37 TO 37
!CHECK WHICH MAP EXISTS
SUB 0,0
DCC 0,MAP
DCC 0,MAP
DCC 0,MAP
LDA 1,LS,K2
AND# 1,0,5ZR
HALT
SUBZL 1,1
MOVS 1,1
DCC 1,MAP
DCC 0,MAP
LDA 1,LS,K5
AND# 1,0,SNR
SUB 0,0,SKP
ADC 0,0
STA 0,NEWMP
!DON'T USE DEV. PROTECTION IF NEWMAP
MOV 0,0,SNR
JMP \*4
SUB 0,0
STA 0,USRMP
JMP 0,3
STA 3,LS-S3
SUB 0,0
LDA 1,LS,K6
LDA 1,USRMP
LDA 1,LS,K1
LDA 2,LS,K4

!SHD GET 0 IN BITS 1-5
!76000
!CHECK BITS 1-5
!MAP PROBLEM?
!BIT 7 = 1
!FMT BIT 7 ON
!SHD GET BIT 2 IF NEMMAP
!200000
!SKP IS NEW MAP
!OLD MAP
!NEW MAP
!SET MAP SWITCH
!SKP IS NEW MAP
!SET UP USRMP WORD
!EXIT , NEMMAP
!1000
!SET UP USRMP,OLD MAP

```

0017 EXMEM
01 00741 034414
02 00742 113410
03 00743 020426
04 00744 030414
05 00745 025000
06 00746 122415
07 00747 002546
08 00750 107000
09 00751 065003
10 00752 151400
11 00753 000772
12 00754 000010
13 00755 076000
14 00756 001000
15 00757 000037
16 00760 000761
17 00761 040037
18 00762 042066
19 00763 044137
20 00764 046347
21 00765 050377
22 00766 052377
23 00767 054377
24 00770 056376
25 00771 020000
26 00772 000000

LDA 3,LS,K2
LMP
LDA 0,LS,K5
LDA 2,LS,K4
LDA 1,0,2
SUB# 1,0,SNR
JMP 2LS,S5
ADD 0,1
OOA 1,MAP
INC 2,2
JMP LSY51
8.
LS,K1:
LS,K2: 76000
LS,K4: 1000
LS,K3: 37
LS,K4: *1
37*40000
66*42000
137*44000
347*46000
377*50000
377*52000
377*54000
376*56000
LS,K5: 20000
:END OF PROTECT MASK TABLE

```

```

10018 EXMEM
01
02
03
04
05 00773 022450
06 00774 040463
07 00775 054494
08 00776 054494
09 00777 024474
10 01000 124400
11 01001 044445
12 01002 030436
13 01003 102400
14 01004 041000
15 01005 151400
16 01006 041000
17 01007 151400
18 01010 024437
19 01011 100000
20 01012 041000
21 01013 151400
22 01014 125404
23 01015 000775
24 01016 050442
25 01017 010427
26 01020 000762
27 01021 014437
28 01022 024426
29 01023 030426
30 01024 100000
31 01025 041000
32 01026 151400
33 01027 125404
34 01030 000775

:LSSET-SET UP SYSTEM FOR RUNNING
:CLR MEM ALLOCATION TABLES EXISTM
:AND INIT MAP OPTION TRAP LOCATIONS
LSSET: LDA 0,0LW,K1
:SAVE RTN ADDR
LWS.1: STA 0,LSETB
STA 3,LW,S3
LDA 1,PROGK
MEG 1,1
STA 1,LW,C1
LDA 2,LSETB
SUB 0,0
STA 0,0,2
INC 2,2 :TABLES FOR 1 PROG
STA 0,0,2
INC 2,2
LDA 1,LW,K4
COM 0,0
STA 0,0,2
INC 2,2
INC 1,1,SZR
JMP LWS.2
STA 2,LSETB
ISZ LW,C1
JMP LWS.1
DSZ LSETB
LDA 1,LW,K6
LDA 2,LW,K7
COM 0,0
LWS.2: STA 0,0,2
INC 2,2
INC 1,1,SZR
JMP LWS.2
STA 2,LSETB
ISZ LW,C1
JMP LWS.1
DSZ LSETB
LDA 1,LW,K6
LDA 2,LW,K7
COM 0,0
LWS.3: STA 0,0,2
INC 2,2
INC 1,1,SZR
JMP LWS.3
:SET UP MEM ALLOC
:FOR 1 PROG
:1ST 2 WRDS = 0
:1ST 32=-1
:STORE -1 32 TIMES
:NEW END SYS TABLES
:DO ONE MORE TABLE
:REAL END OF SYS TABLES
:CLEAR CORE EXIST MAP

```

```

10020 EXMEM
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
:LSIZE-LINK THE 32K AND ABOVE 32K MEM SIZERS
:SET UP UP32L AND HIGHK
:EXISTM=0'S
LSIZE: STA 3,LS.S3
MS232 JSR
STA 0,UP32L :AC0=LAST ADRS IN 32K
LDA 1,LS.K2 :5 BITS PHYS PAGE 37
AND 0,1
MOVS 10 01101 125300
MOVZR 1,1 :AC1=LAST PHYS PAGE(32K)
MOVZR 1,1
STA 1,HIGHK :IN CASE NOT 32K OR NO MAP
INCZL# 0,0,SNC :SKP IS 32K
JMP 3LS.S3 :EXIT MEM <32K
LDA 1,MPSWT
MOV 1,1,SNR :#=0 IS NO MAP OPTION
JMP 3LS.S3
MS128 JSR
STA 0,HIGHK :# OF PHYS PAGES(1K)
JMP 3LS.S3
LS.S3: 0
UP32L: 0
HIGHK: 0

```

```

10019 EXMEM
01 000003 ECADRS=3
02 000002 SCADRS=2
03 :NOW SET UP JMP @'S IN MAP TRAPS
04 LDA 1,MP.K1
05 STA 1,ECADRS :LOAD PROTECT ADRS
06 LDA 1,MP.K2
07 STA 1,SCADRS :LOAD SUPER CALL
08 LDA 0,LW.09
09 LDA 0,SSP
10 LDA 1,LW.10
11 STA 1,SSP+2
12 JMP @LW.S3
13 LW.S3: 0
14 LDA 0,USTNM
15 LDA 0,USTSS
16 LW.K3: JUSTES
17 LW.K3: JUSTES
18 LW.C1: 0
19 LW.K4: =32.
20 LW.K6: =16.
21 LW.K7: EXISM
22 LW.K8: 176000
23 LW.09: 450
24 LW.10: 572
25 MP.K1: IOVAL
26 MP.K2:CDISP
27 LSTB: 0
28 LSETB: 0
29
30 :GPRK=GENERATE PROGRAM COUNT
31 :THE FOLLOWING SUBROUTINE SIMPLY DETERMINES
32 :HOW MANY TEST PROGRAMS ARE IN CORE
33 :ALONG WITH THE DIAGNOSTIC LINKER
34 :7MAX=LAST LINKER ZLOC=#TESTS INTO PROGK
35 :KLZMX :LAST LINKER ZLOC
36 :31STZM :LAST ZPAGE FILLED
37 LDA 0,2
38 MOV 0,2
39 SUB 1,0 :AC0=NUMBER TESTS
40 STA 0,PROBK
41 SUB 0,0
42 STA 0,0,2
43 JMP
44 KLZMX: LZMAX
45 ISTZM: LSTZM
46 PROGK: 0

```



10021 EXMEM

01 01 01120 054435 MSZ32: STA 3,XMS32 ;SAVE  
02 11 01121 126400 SUB 1,1 ;OFOR FIRST 1K  
03 12 01122 030431 LOA 2,K1K ;177 FOR END OF 1K  
04 13 01123 135000 ADD 1,2 ;CURRENT 1K FIELD  
05 14 01124 025000 LDA 1,0,2 ;GET CELL  
06 15 01125 120000 COM 1,0 ;CHNG BITS  
07 16 01126 041000 STA 0,0,2  
08 17 01127 021000 LDA 0,0,2 ;COM MEM EXISTS  
09 18 01130 122405 SUB 1,0,SNR ;AND WE'LL SKIP  
10 19 01131 009420 JMP M32SZ ;FLST WAS NONEXIST.  
11 20 01132 050422 STA 2,M32TEM ;RESTORE CELL  
12 21 01133 045000 STA 2,0  
13 22 01134 141300 MOVS 2,0  
14 23 01135 101200 MOVR 0,0  
15 24 01136 101200 MOVR 0,0  
16 25 01137 028417 LDA 1,K37 ;ACO=PHYS PAGE #  
17 26 01140 123400 AND 1,0  
18 27 01141 030416 LDA 2,KXIST ;ADRS EXIST TABLE  
19 28 01142 066416 JSR @MS.L1 ;SET EXIST BIT=1  
20 29 01143 101001 MOV 0,0,SKP  
21 30 01144 063077 HALT ;\*\*\*CAN'T HAPPEN EXIST BIT HAD TO =0  
22 31 01145 024407 LDA 1,M32TE ;ACI=LAST 1K TOP ADDRESS  
23 32 01146 125400 INC 1,1  
24 33 01147 125135 MOVZL#  
25 34 01150 007152 JMP MSZ32+2 ;NOT DONE SIZING  
26 35 01151 020403 M32SZ: LDA 0,M32TE ;ACO=HIGHEST AVAIL.  
27 36 01152 002403 JMP @XMS32  
28 37 01153 001777 K1K: 1777  
29 38 01154 000000 M32TE: 0  
30 39 01155 000000 XMS32: 0  
31 40 01156 00037 K37: 37  
32 41 01157 006133 KYIST: EXISM  
33 42 01160 001475 MS.L1: CMAPB

10022 EXMEM

01 08 01161 054774 MS128: STA 3,XMS32  
02 09 01162 024434 LDA 1,MS.1K ;=377  
03 10 01163 034065 LDA 3,MEMIP ;WHICH MAP?  
04 11 01164 175005 MOV 3,3,SNR ;SKR=MMPUI  
05 12 01165 125220 MOVZR 1,1 ;=177  
06 13 01166 020731 LDA 0,HIGHK  
07 14 01167 101400 INC 0,0  
08 15 01170 106433 SUBZ# 0,1,SNR ;SKP IS NOT SIZED ALL YET  
09 16 01171 000417 JMP M128S  
10 17 01172 062003 DDB 0,MAP  
11 18 01173 032421 LDA 2,AK32K ;GET CELL  
12 19 01174 154000 COM 2,3  
13 20 01175 056417 STA 3,AK32K  
14 21 01176 036415 LDA 3,AK32K  
15 22 01177 052415 STA 2,AK32K  
16 23 01500 158405 SUR 2,3,SNR ;SKIP IS CELL EXISTS  
17 24 01501 009407 JMP M128S ;MEM IS SIZED  
18 25 01502 040715 STA 0,HIGHK ;SAVE NEW TOP MEM  
19 26 01503 030754 LDA 2,KXIST ;EXIST MAP  
20 27 01504 066754 JSR @MS.L1 ;SET EXIST BIT  
21 28 01505 101001 MOV 0,0,SKP ;HAD TO GO 0-1  
22 29 01506 063077 HALT ;\*\*BIT HAD TO =0 CMAPB GOOF  
23 30 01207 000753 JMP MS128+1  
24 31 01210 020707 M128S: LDA 0,HIGHK  
25 32 01211 030745 LDA 2,K37  
26 33 01212 075003 DDB 0,MAP  
27 34 01213 002742 JMP @XMS32  
28 35 01214 077777 K32K: 77777  
29 36 01215 017400 MPF32: 17400  
30 37 01216 000377 MS.1K: 377

\*MS128=MEMORY SIZING ABOVE 32K  
\*MAP OPTION MUST EXIST  
;USES LOG PAGE# ALWAYS MAPPED TO SIZE  
;EXIT IS WITH ACO=LAST PHYSICAL PAGE#  
;EXISM(EXIST MAP) MUST =0'S ABOVE 32K  
;ASSUMES MEMORY TO BE CONTIGUOUS

10023 EXMEM

```

01
02
03 01217 050406
04 01220 030406
05 01221 034406
06 01222 024406
07 01223 133710
08 01224 024401
09 01225 001217
10 01226 06133
11 01227 06133
12 01230 000020
13
14
15
16
17
18
19 01231 050421
20 01232 102400
21 01233 026420
22 01234 066425
23 01235 026420
24 01236 026420
25 01237 066422
26 01240 022417
27 01241 026417
28 01242 066417
29 01243 020257
30 01244 101005
31 01245 002405
32
33 01246 022406
34 01247 105000
35 01250 066411
36 01251 002401
37 01252 000000
38 01253 000222
39 01254 001116
40 01255 000220
41 01256 000221
42 01257 001057
43 01260 001060
44 01261 001431

```

:MOVEA-MOVE THE EXIST MAP  
:INIT THE AVAILABLE MAP POSITION  
MVETA: STA 3,XMVET  
LDA 2,KEYMP :A2=STR EXIST  
LDA 3,KAVMP :A3=STR AVAILABLE  
LDA 1,K16 :I6 FOR COUNTING  
BLM  
JMP  
MVETA: MVETA  
EXISM  
KEYMP: EXISM  
KAVMP: AVALM  
K16: 16  
:UPL32=SET UP USABLE SCRATCH LIMITS IN 32K  
:ALL CORE ABOVE 32K IS ASSUMED TO BE USABLE SCRATCH  
:CLEAR AVAILABLE BITS FOR THOSE AREAS USED  
:SO THAT THEY WILL NOT BE ASSIGNED AS A SCRATCH AREA  
:TO ANY TEST  
:SUBR CBLIM IS USED TO CLEAR AVAILABLE BITS  
UPL32: STA 3,XUBL3 :SAVE RETURN  
SUB 0,0  
LDA 1,0KNMAX :0 TO NMAX  
JSR 0UBLIM :PROTECTS PROGRAMS  
LDA 1,0KSTSS :STRY SYMBOLS  
LDA 0,0KSTES :END SYMBOLS  
JSR 0UBLIM :PROTECTS SYMBOLS  
LDA 0,0KLETB :STRY LINKER TABLES  
LDA 1,0KLETB :END LINKER TABLES  
JSR 0UBLIM :PROTECTS LINKER TABLES  
LDA 0,EGGS  
MOV 0,0,SNR  
JMP 0XUBL3  
:PROTECT DTOS AREA  
LDA 0,0KUP32  
MOV 0,1  
JSR 0UBLIM  
JMP 0XUBL3 :RETURN  
0  
-USTIM  
UP32L  
:USTIS  
:USTES  
LSYTB  
LSETB  
UBLIM: CBLIM

10024 EXMEM

```

01
02
03
04
05
06
07 01262 054427
08 01263 102400
09 01264 040426
10 01265 026406
11 01266 034424
12 01267 166415
13 01270 002421
14 01271 020456
15 01272 117000
16 01273 031400
17 01274 102400
18 01275 041002
19 01276 041377
20
21 01377 021005
22 01380 024065
23 01301 125005
24 01302 009403
25 01303 143770
26
27 01305 041005
28 01306 007000
29 01307 010403
30 01310 000755
31 01311 000000
32 01312 000000
33
34
35

```

:INIT-TEST INITIALIZE  
:SEQUENCE THROUGH THE INITIALIZE ADDRESSES  
:FOR EACH TEST LOADED ALONG WITH LINKER  
:I POINTER FOR EACH TESTS PARAMETERS  
:I IS IN ALL USED LOCATIONS ABOVE ZLOC  
TINIT: STA 3,XTINI  
SUB 0,0  
STA 0,NPROG :PROG TO INIT  
LDA 1,PROG  
LDA 3,NPROG :NEXT PROG TO INIT  
SUB# 3,1,SNR :SKP IS NOT DONE ALL  
JMP 0XTINI :EXIT ALL PROGS INITED  
SUB 0,0  
LDA 0,LK.K1  
LDA 2,0,3 :GET INIT ADRS  
SUB 0,0  
STA 0,2,2 :CLEAR WAIT INT SW  
STA 0,1,2 :CLRS INTA WAIT CTR  
:REMOVE I/O PROTECT BIT AND DCH PROTECT BIT IF NEWMAP  
LDA 0,5,2  
LDA 1,NEWMAP  
MOV 1,1,SNR :SKP=NEW MAP  
ANDI 133,0  
STA 0,5,2 :RESTORE WORD  
JSR 00,2 :AND INIT THIS TEST  
ISZ NPROG :STEP TO NXT PROG  
TINIT+3 :AND DO AGN  
XTINI: 0  
NPROG: 0

```

10025 EXMEM
01 01313 054435 LPRG: STA 3,L,R,S3
02 01314 024170 LDA 1,CURPR
03 01315 104110 SBI L,1
04 01316 125112 MOVL# 1,1,SZC
05 01317 124520 NEGZL 1,1
06 01320 044170 LPRG0: STA 1,CURPR
07 01321 030426 LDA 2,L,R,K1
08 01322 133000 ADD 1,2
09 01323 035000 LDA 3,0,2
10 01324 054070 STA 3,PSTRT
11 01325 031001 LDA 2,1,2
12 01326 151005 MOV 2,2,SNR
13 01327 032423 LDA 2,BLN,K9
14 01330 050071 STA 2,PENDA
15 01331 014071 DSZ PENDA
16 01332 031402 LDA 2,2,3
17 01333 151132 MOVZL# 2,2,SZC
18 01334 000760 JMP LPRG+1
19 01335 020170 LPRG: LDA 0,CURPR
20 01336 103020 ADDZ 0,0
21 01337 105120 MOVZL 0,1
22 01340 127120 ADDZL 1,1
23 01341 125120 MOVZL 1,1
24 01342 107000 ADD 0,1
25 01343 022410 LDA 0,2,XYTB
26 01344 123000 ADD 1,0
27 01345 040057 JMP @L,R,S3
28 01346 002402 STA 0,ALTL
29 01347 000173 L,R,K1: LZMAX
30 01350 000000 L,R,S3: 0
31 01351 001073 LPRG: PROGK
32 01352 000222 LN,K9: *USTIM
33 01353 001057 XSYTB: LSTYB

10026 EXMEM
01
02
03
04 01354 054043 !LSTRP-START PROGRAM
05 01355 050070 !ENTER TEST SELECTED AT ITS EXECUTION ENTRY POINT
06 01356 020072 LSTRP: STA 3,L,SS3
07 01357 120010 LDA 2,PSTRT
08 01358 041377 LDA 0,RTIM
09 01359 102400 ADI 2,0
10 01360 041377 STA 0,1,2
11 01361 102400 SUB 0,0
12 01362 021001 LDA 0,1,2
13 01363 040435 !GET EXEC ADDR
14 01364 020066 !SKIP = MAPPED
15 01365 101095 MOV 0,0,SNR
16 01366 024432 JMP @L,S,I1
17 01367 025005 LDA 1,S,2
18 01370 020060 LDA 0,USRMP
19 01371 030065 LDA 2,NEWMP
20 01372 151004 MOV 2,2,SZ
21 01373 104410 IOR 0,1
22 01374 044061 DDA 1,USTAT
23 01375 045003 !SAVE FOR CDISP
24 01376 024232 !ENTER TEST
25
26
27
28
29 01377 020061 !LRETP-RETURN FROM TEST PROG CALL
30 01400 101220 LRETP: LDA 0,USTAT
31 01401 101120 MOVZL 0,0
32 01402 040061 STA 0,USTAT
33 01403 020075 LDA 0,PFAIS
34 01404 101004 MOV 0,0,SZ
35 01405 000403 JMP *3
36 01406 006100 JSR @L,MESS
37 01407 001421 PFAIT
38 01410 102000 ADC 0,0
39 01411 040075 STA 0,PFAIS
40 01412 034405 LDA 3,L,SS3
41 GETSP
42 01413 030063 LDA 2,XSSP
43 01414 031000 LDA 2,0,2
44 01415 055000 STA 3,0,2
45 01416 107710 POPB
46 01417 000000 L,SS3: 0
47 01420 000000 L,S,I1: 0
48 01421 177777 PFAIT: *TXTE !<177><177><15><12>POWER FAIL!
49 005215
50 147520
51 142727
52 120322
53 040706
54 146511
55 000000

```

10027 EXMEM

```

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
:CBLM-CLEAR AVAILABLE BITS BETWEEN LIMITS
:ACQ=LOWEST ADDR ACI=HIGHEST ADDRESS
: SINCE SOME USED AREAS MAY OVERLAP IN PAGES
: OCCASIONALLY 2 PASSES THROUGH CMAPP WILL BE REQ
: CBLM: STA 3,XCBLM
MOV 0,0
MOV 1,1
MOVZ 0,0
MOVZ 1,1
LDA 0,0
AND 2,0
AND 2,1
STA 0,CBLWR
STA 1,CBUPR
LDA 0,CBLWR
LDA 2,KAVLM
JSR 2,XCMPB
JMP -1
LDA 0,CBLWR
ISZ CBLWR
LDA 1,CBUPR
SUB 0,1,LSZR
JMP -1
JMP 2,XCBLM
XCBLM: 0
CMAPP: 0
KAVLM: 0
CBLWR: 0
CBUPR: 0
K3TC: 0
57

```

10028 EXMEM

```

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
:GETPA-GET A PHYSICAL ASSIGNMENT
:ACQ=ALLOCATION TABLE POSITION
: RETURN WITH ACI=PHYSICAL PAGE ASSIGNMENT
: SKIP EXIT IF THE ALLOC ASSIGN DOES NOT =-1
: THE INTEGRITY OF ACQ IS PRESERVED
GETPA: PSH 0,0
LDA 2,ALTR
LDA 2,ALTR
ADD 0,2
LDA 1,2,2
COM# 1,1,LSZR
INC 3,3
POP 0,0
JMP 0,3
:CMAPP-COMPLIMENT MAP BIT
:COMPLIMENT THE STATE OF MEMORY MAP BIT
:THE START ADDR OF THE MAP IS INAC2
:ACQ CONTAINS UP TO 7 BITS OF ADDRESS WITH
:BITS 12 TO 15 =# BITS TO SHIFT RIGHT
:BITS 9 TO 11 =WORD POSITION IN MAP
:THOSE 7 BITS ARE THE PHYSICAL PG # OF A 1K OF MEM
:SKIP ON EXIT IF THE BIT IS GOING 1-0
CMAPP: SZB 2,0
MOV 0,0,SKP
JMP 0,3
BTZ 2,0
JMP 1,3

```

10029 EXMEM

```

01 :CDISP-LINKER CALL DISPATCH ROUTINE
02 :DIRECTS MEM ALLOCATION AND OTHER CALLS
03 :TO THE CORRECT HANDLER FOR PROCESSING
04 :CALL WILL BE TO ONE OF THE FOLLOWING
05 :ASCR A SCRATCH AREA
06 :ESCR EXPAND SCRATCH AREA
07 :RSCR RELEASE SCRATCH AREA
08 :POECL PRINT DECIMAL
09 :ERRTX PRINT TEXT MESSAGE
10 :RETRN RETURN FROM TEST EXECUTION
11 :PARNG RANDOM # GENERATION
12 :ADIVI LINK TO SOFTWARE DIVIDE
13 :ENTER DEVICE INTO INTERRUPT SERV
14 :ERRDI INIT. ERROR TYPEDITS
15 :ERRRC APPEND TO ERR TYPEDOUT
16 :CDISP: LDA 3,XSSP ;STACK POINTER ADRS
17 :LDA 3,0+3 ;STK PTR.
18 :STA 3,CD,T2 ;SAVE IT.
19 :LDA 2,0+3 ;CALL ADRS +1
20 :STA 2,CD,T3 ;SAVE RET. ADRS
21 :LDA 1,USTAT ;LDA 1,USTAT
22 :MOVZR 1,1,SZC ;MOVZR 1,1,SZC
23 :NIOP MAP ;NIOP MAP
24 :LDA 0,0+2 ;LDA 0,0+2
25 :CLM 0,0 ;CLM 0,0
26 :JSR @ASCR ;JSR @ASCR
27 :JSR @ERRRC ;JSR @ERRRC
28 :HALT ;HALT
29 :LDA 1,-3+3 ;LDA 1,-3+3
30 :LDA 2,-2+3 ;LDA 2,-2+3
31 :LDA 3,-4+3 ;LDA 3,-4+3
32 :XCH 0,3 ;XCH 0,3
33 :STA 3,+1 ;STA 3,+1
34 :JSR @ASCR ;JSR @ASCR
35 :MOV 0,0,SKP ;MOV 0,0,SKP
36 :ISZ CD,T5 ;ISZ CD,T5
37 :I*3 RET ;I*3 RET
38 :LDA 3,CD,T2 ;LDA 3,CD,T2
39 :STA 0,-4+3 ;STA 0,-4+3
40 :LDA 1,-3+3 ;LDA 1,-3+3
41 :LDA 2,-2+3 ;LDA 2,-2+3
42 :LDA 2,CD,T3 ;LDA 2,CD,T3
43 :STA 2,0+3 ;STA 2,0+3
44 :LDA 1,USTAT ;LDA 1,USTAT
45 :DOA 1,MAP ;DOA 1,MAP
46 :POPB ;POPB
47 :HALT ;HALT
48 :;TEMP STORES AND CONSTANTS
49 :CO,T1: 0
50 :CO,T2: 0
51 :CO,T3: 0

```

10030 EXMEM

```

01 :ENTPA-ENTER PHYSICAL ASSIGNMENT
02 :ENTPA: PSH 1,3
03 :1,0ALTB:GET # ENTRIES
04 :2,ALTB:ADRS OF ALLOC TBL
05 :1,2 :WRD #
06 :0,2,2 :ENTER ASSIGNMENT
07 :3,1,1 :MAP
08 :3,1,1 :MAP
09 :3,1,1 :MAP
10 :3,1,1 :MAP
11 :3,1,1 :MAP
12 :3,1,1 :MAP
13 :3,1,1 :MAP
14 :3,1,1 :MAP
15 :3,1,1 :MAP
16 :3,1,1 :MAP
17 :3,1,1 :MAP
18 :3,1,1 :MAP
19 :3,1,1 :MAP
20 :3,1,1 :MAP
21 :3,1,1 :MAP
22 :3,1,1 :MAP
23 :3,1,1 :MAP
24 :3,1,1 :MAP
25 :3,1,1 :MAP
26 :3,1,1 :MAP
27 :3,1,1 :MAP
28 :3,1,1 :MAP
29 :3,1,1 :MAP
30 :3,1,1 :MAP
31 :3,1,1 :MAP
32 :3,1,1 :MAP
33 :3,1,1 :MAP
34 :3,1,1 :MAP
35 :3,1,1 :MAP
36 :3,1,1 :MAP
37 :3,1,1 :MAP
38 :3,1,1 :MAP
39 :3,1,1 :MAP
40 :3,1,1 :MAP
41 :3,1,1 :MAP
42 :3,1,1 :MAP
43 :3,1,1 :MAP
44 :3,1,1 :MAP
45 :3,1,1 :MAP
46 :3,1,1 :MAP
47 :3,1,1 :MAP
48 :3,1,1 :MAP
49 :3,1,1 :MAP
50 :3,1,1 :MAP
51 :3,1,1 :MAP
52 :3,1,1 :MAP
53 :3,1,1 :MAP
54 :3,1,1 :MAP
55 :3,1,1 :MAP
56 :3,1,1 :MAP
57 :3,1,1 :MAP
58 :3,1,1 :MAP
59 :3,1,1 :MAP
60 :3,1,1 :MAP

```

```

0031 EXMEM
01 01631 123400 AND 1,0
02 :WHICH MAP?
03 01632 030065 LDA 2,NEWMP
04 01633 151005 MOV 2,2,SNR
05 01634 000404 JMP .+4
06 01635 030531 LDA 2,MPTBL
07 01636 113000 ADD 0,2
08 01637 050526 STA 2,MPTBP
09 01640 105300 MOV# 0,1
10 01641 127120 ADDZL 1,1
11 01642 107000 ADD 0,1
12 01643 030065 LDA 2,NEWMP
13 01644 151005 MOV 2,2,SNR
14 01645 000403 JMP .+3
15 01646 046517 STA 1,MPTBP
16 01647 000404 JMP LM.LB
17 :OLD MAP ENTRY
18 01650 034060 LDA 3,USRMP
19 01651 167000 ADD 3,1
20 01652 045003 DOA
21 01653 105700 LM.LB: INCS
22 01654 127120 ADDZL 1,1
23 01655 030071 LDA
24 01656 132432 SUBZ# 1,2,SZC
25 01657 000746 JMP LM.L2
:ENTER NEXT SEQUENCE WITH AC0=HIGHEST PAGE USED FOR TEST

```

```

:IF TEST HAS SCRATCH AREA ASSIGNED MAP AS MUCH AS POSSIBLE
:STARTING AT THE FIRST 1K ABOVE PROGRAM STORAGE
: # 1K FIELDS ASSIGNED
10032 EXMEM
01
02
03 01660 026067 LDA 1,@ALTLB
04 01661 125005 MOV 1,1,SNR
05 01662 000455 JMP LM.PP
06 01663 102400 SUB 0,0
07 01664 006477 JSR @LGTPA
08 01665 063077 HALT
09 01666 147770 ANDI 37,1
10
11 01670 131004 MOV 1,2,SZR
12 01671 000403 JMP .+3
13 01672 172470 ELEM 2,10
14
15 01674 145300 MOV# 2,1
16 01675 127120 ADDZL 1,1
17 01676 034041 LDA 3,SSP+1
18 01677 045401 STA 1,1,3
19 01700 051403 STA 2,3,3
20 01701 102400 SUB 0,0
21 01702 006461 JSR @LGTPA
22 01703 000434 JMP LM.PP
23 01704 034041 LDA 3,SSP+1
24 01705 031403 LDA 2,3,3
25 01706 034065 LDA 3,NEWMP
26 01707 175005 MOV 3,3,SNR
27 01710 000404 JMP .+4
28 01711 034455 LDA 3,MPTBL
29 01712 157000 ADD 2,3
30 01713 054452 STA 3,MPTBP
31 01714 151300 MOV# 2,2
32 01715 153120 ADDZL 2,2
33 01716 147000 ADD 2,1
34 :WHICH MAP?
35 01717 034065 LDA 3,NEWMP
36 01720 175005 MOV 3,3,SNR
37 01721 000403 JMP LM.LC
38 01722 046443 STA 1,MPTBP
39 01723 000404 JMP LM.LD
40 01724 034060 LM.LC: LDA 3,USRMP
41 01725 167000 DOA
42 01726 065003 LM.LD: LDA
43 01727 024432 LM.LD: LDA
44 01730 133000 ADD 1,2
45 01731 034041 LDA 3,SSP+1
46 01732 051402 STA 2,2,3
47 01733 101400 INC 0,0
48 01734 011403 ISZ 3,3
49 01735 151523 INCL 2,2,SNR
50 01736 000744 JMP LM.L3
51 01737 020065 LM.PP: LDA 0,NEWMP
52 01740 101005 MOV 0,0,SNR
53 01741 000410 JMP LM.DN
54 01742 020060 LDA 0,USRMP
55 01743 063003 DOA
56 01744 102400 SUB 0,0
57 01745 024413 LDA 1,LM.37
58 01746 030420 STA 2,MPTBL
59 01747 125400 INC 1,1
60 01750 113410 LMP

```

```

:GET FIRST PHYS 1K#
:ALTLB CAN'T BE EMPTY
:AC2=LOGICAL PAGE#
:CAN'T MAP TO 0
:1ST 8K EACH 32K AT 20000
:CREATE SCRLO
:GET FRAME PTR
:START OF ACCESSABLE SCRAT
:NEXT PHYS PAGE ASSIGNED
:DOONE ALL EXIT
:AC2=LOGICAL PG. #
:WHICH MAP?
:SKP=NEW MAP
:SET POINTER TO THIS ENTRY
:MAP LOGICAL TO PHYS
:NEW HI SCRATCH LIMIT
:+1 ALLOC TBL POS
:+1 LOGICAL PAGE
:C=1 IF HI IS 77777(32K)
:SKP=NEWMAP
:SET UP FMT BITS
:AC1=37
:AC2=ADDR OF TBL
:AC1=32. = # OF WORDS TO LMP
:LOAD MAP TBL'S

```

```

0033 EXMEM
01 01751 036041 LM.DN: LDA 3,SSP+1
02 01752 021401 LDA 0,1,3
03 01753 025402 LDA 1,2,3
04 01754 040171 STA 0,SCRLO
05 01755 044172 STA 1,SCRHI
06 01756 127110 RTN
07 01757 000377 LM.K1: 377
08 01760 000377 LM.K2: 377
09 01761 001777 LM.K2: 1777
10 01762 000000 LM.TM: 0
11 01763 001465 LGTPA: GETPA
12 01764 101777 LM.AP: 101777
13 01765 001767 MPTBP: *2
14 01766 001767 MPTBL: *1
15 01767 000040 .BLK 32.

```

```

10034 EXMEM
01
02
03 02027 006734
04 02030 000721
05 02031 129300
06 02032 127120
07 02033 034041
08 02034 045401
09 02035 006726 LM.L4:
10 02036 000715
11 02037 125300
12 02040 127120
13 02041 030720
14 02042 133000
15 02043 034041
16 02044 051402
17 02045 101400
18 02046 000767

```

```

*MAP OPTION DOES NOT EXIST
*SIMPLY SET LIMITS TO SCRATCH AREA ASSIGNED
LM.NM:
ALGTPA *SKP=AC1 *PHYS PAGE#
LM.DN *EXIT NO SCRATCH
MOVZ 1,1
ADDZL 1,1
LDA 3,SSP+1
STA 1,1,3 *LOW=FIRST PHYS 1K
ALGTPA *SKP=AC1=PHYS PG#
LM.DN *EXIT SCRHI ADJUSTED
MOVZ 1,1 *PG# POSITIONED TO PHYS
ADDZL 1,1
LDA 2,LM.K2
ADD 1,2
LDA 3,SSP+1
STA 2,2,3 *NO TERT CAN HAVE 32K IF
INC 0,0
JMP LM.L4 *MAP OPTION NONEEXIST

```

```

10036 EXMEM
01
02
03
04
05
06 02047 11310
07 02050 05432
08 02051 022067
09 02052 101004
10 02053 063077
11 02054 020107
12 02055 026415
13 02056 123032
14 02057 008407
15 02060 010107
16 02061 030412
17 02062 006441
18 02063 006440
19 02064 000411
20 02065 000767
21 02066 102400
22 02067 040107
23 02070 008007
24 02071 000000
25 02072 001117
26 02073 006153
27 02074 000403
28 02075 006454
29 02076 010404
30 02100 143210
31 02101 002401
32 02102 000000
33 02103 001556
34
35
36
37
38
39
40
41
42 02104 11310
43 02105 054775
44 02106 022067
45 02107 101005
46 02110 063077
47 02111 020107
48 02112 024110
49 02113 123415
50 02114 000763
51 02115 010107
52 02116 030755
53 02117 006404
54 02120 006403
55 02121 000754
56 02122 000755
57 02123 001475

```

```

PASSCR-ASIGN A SCRATCH AREA
!ASSIGN SCRLO TO THE NEXT AVAILABLE 1K OF CORE
!IF HIGHEST MEMORY HAS BEEN ASSIGNED EXIT WITHOUT
!SKIPPING-THIS TELLS CBOARD WHEN END OF PASS
ASSCR: PSH 0,2
STA 3,AS,S3
LDA 0,0,ALTLBL
MOV 0,0,SNR
HALT
AS,SL:
LDA 0,CMDUAL
LDA 1,0,AS,G1+1
ADZ# 1,0,SEC
JMP AS,NO
ISZ CMDUAL
LDA 2,AS,G1+2
JSR 0EX,11
JMP AS,XT-2
JMP AS,SL
AS,ND:
SUP 0,0
STA 0,CMDUAL
JMP AS,XT
AS,G1:
HIGHK
AVALM
JMP
JSR
ISZ
JSR
POP 2,0
JMP
AS,S3:
0
LDMPI LDMAP
!ASSIGN NEXT SEQUENTIAL 1K OF CORE
!SKP EXIT IF CORE TO ASSIGN
!NO SKIP IF NO CORE OR IF AT END OF
!MODUAL SIZE
!RETURN IS TO CALL +1 NO SCRATCH ASSIGNED
!RETURN CALL +2 IF SCRATCH WAS EXPANDED
EXSCR: PSH 0,2
STA 3,AS,S3
LDA 0,0,ALTLBL
MOV 0,0,SNR
HALT
LDA 0,CMDUAL
LDA 1,0,SNR
AND# 1,0,SNR
JMP AS,XT
ISZ CMDUAL
LDA 2,AS,G1+2
JSR 0EX,11
JMP AS,XT-2
JMP AS,XT
EX.II: CMAPB

```

```

!PASSCR-RELEASE SCRATCH AREA
!REMOVE 1 K SCRATCH FROM MEM ALLOCATION
!EXIT IS TO CALL +1 ALL SCRATCH RELEASED
!EXIT TO CALL +2 IF STILL SCRATCH LEFT
RLSCR: PSH 0,2
STA 3,AS,S3
LDA 0,0,ALTLBL
NEG 0,0,SNR
JMP AS,XT
COM 0,0
STA 0,0,ALTLBL
JSR 0IGTPA
HALT
MOV 1,0
LDA 2,AS,G1+2
SEX,11
MOV 0,0,SKP
HALT
LDA 0,EX,K1
JSR 0NTPA
DSZ 0,ALTLBL
JMP AS,XT-1
JMP AS,XT
EX,K1: 177777
IGTPA: GETPA
NTPA: ENTPA
!LDSCR-LOAD SCRATCH LIMITS INTO ACO AND ACI
LDSCR: LDA 0,SCRLO
LDA 1,SCRHI
JMP 0,3

```



```

10037 EXMEM
01
02
03
04
05 02155 040513
06 02156 044513
07 02157 050513
08 02160 054513
09 02161 010074
10 02162 102400
11 02163 102400
12 02164 040073
13 02165 061100
14 02166 005132
15
16
17 02167 032411
18 02170 020412
19 02171 040410
20 02172 025001
21 02173 061102
22 02174 151400
23 02175 010404
24 02176 000774
25 02177 000414
26 02200 001053
27 02201 000000
28 02202 177774
29 02203 000207
30 02204 020264
31 02205 101233
32 02206 000491
33 02207 020774
34 02210 065411
35 02211 061111
36 02212 000435

:ERROR - ERROR HANDLER - PRINT ALL ERR INFO
:FIRST PRINT PRG# AND (AC'S)
ERR0H:
STA 0,ER.S0
STA 1,ER.S1
STA 2,ER.S2
STA 3,ER.S3
ISZ ER.TOT
MOV 0,0
MOV 0,0
STA 0,TIMSH
JSR 0,MESS
TXT.1
:PRINT THE CURRENT CONTENTS OF THE USER STACK
:HERE SHOULD ONLY BE A TOTAL OF 4 ENTRIES
LDA 2,0ERRK9
LDA 0,ERK4
STA 0,ERCTR
LDA 1,1,2
ISZ 0,POCT
INC 2,2
ISZ ERCTR
JMP *-4
ERRK9: LW.09
ERCTR: 0
ERK4: -4
:BELL CODE
ERRBELL: LDA 0,SWRES
MOVZ 0,0,SNC
JMP ERXIT,ERRBELL-1
LDA 0,ERBELL-1
SKPBN TIO
DDAS 0,TIO
JMP ERXIT

:PRINT MEM ALLOCATION ASSIGNMENTS
ERMP:
LDA 0,ER.C1
JSR 0,ER.K4
JMP ERBELL
:PRINT PHYS PAGES
:ALSO IN MODULO 1K
:AND LOGICAL MEM
LDA 0,ER.C1
JSR 0,ER.K4
JMP ERBELL
:PRINT SCRATCH LIMITS
LDA 1,SCRLO
JSR 0,POCT
LDA 1,SCRHI
JSR 0,POCT
LDA 1,SPSWT
MOV 1,J,SNR
JMP ERBELL
:TYPE MEM ALLOC IF MAP
:NO MAP FORGET REST OF TYPE
:SEE IF ANY SCRA
:SKP=SCRA EXISTS
:EXIT NO SCRA TO TYPE
SUB 0,0
STA 0,ER.C1
JSR 0,MESS
TXT.4

:PRINT MEM ALLOCATION ASSIGNMENTS
ERMP:
LDA 0,ER.C1
JSR 0,ER.K4
JMP ERBELL
:PRINT PHYS PAGES
:ALSO IN MODULO 1K
:AND LOGICAL MEM
LDA 0,ER.C1
JSR 0,ER.K4
JMP ERBELL
:PRINT SCRATCH LIMITS
LDA 1,SCRLO
JSR 0,POCT
LDA 1,SCRHI
JSR 0,POCT
LDA 1,SPSWT
MOV 1,J,SNR
JMP ERBELL
:TYPE MEM ALLOC IF MAP
:NO MAP FORGET REST OF TYPE
:SEE IF ANY SCRA
:SKP=SCRA EXISTS
:EXIT NO SCRA TO TYPE
SUB 0,0
STA 0,ER.C1
JSR 0,MESS
TXT.4

10038 EXMEM
01
02 02213 006100
03 02214 003137
04 02215 024171
05 02216 006102
06 02217 024172
07 02220 006102
08 02221 024086
09 02222 125005
10 02223 000761
11 02224 020172
12 02225 101005
13 02226 000756
14 02227 102400
15 02230 040415
16 02231 006100
17 02232 003145
18
19 02233 020412
20 02234 060412
21 02235 000747
22 02236 006101
23 02237 006103
24 02240 006104
25 02241 024171
26 02242 006102
27 02243 000741
28
29 02244 000170
30 02245 000000
31 02246 001465
ER.K1: CURPR
ER.C1: 0
ER.K4: GETPA

```

10039 EXMEM

```

01
02
03 02247 020421 EREXI: LDA 0,ER.S0
04 LDA 1,ER.S1
05 02250 024421 LDA 2,ER.S2
06 02252 034421 LDA 3,ER.S3
07 02253 001400 JMP 0,3
08
09
10
11
12
13
14 02254 040414 ERROE: STA 0,ER.S0
15 02255 044414 STA 1,ER.S1
16 02256 050414 STA 2,ER.S2
17 02257 054414 STA 3,ER.S3
18 02260 006101 JSR 0LCRLF
19 02261 028407 JSR 0LPOCT
20 02262 006102 LDA 1,ER.S1
21 02263 024406 JSR 0LPOCT
22 02264 006102 LDA 1,ER.S2
23 02265 024405 JSR 0LPOCT
24 02266 006102 JSR 0LPOCT
25 02267 000760 JMP EREXI
26
27 02270 000000 ER.S0: 0
28 02271 000000 ER.S1: 0
29 02272 000000 ER.S2: 0
30 02273 000000 ER.S3: 0
31
32
33 02274 040774 ERTXT: STA 0,ER.S0
34 02275 040405 STA 1,ER.S1
35 02276 044773 STA 2,ER.S2
36 02277 050773 STA 3,ER.S3
37 02300 054773 JSR 0LMESS
38 02301 006100 ER.TP: 0
39 02302 000000 ER.TP: 0
40 02303 000744 JMP EREXI

```

```

:TEXT CALL ADRS OF TEXT IS IN ACO
:CALL MUST ONLY BE MADE WHILE IN FIRST LEVEL TEST
:TEXT ADRS STORED HERE

```

10040 EXMEM

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
:FILENAME=TTYIO
:TELETYPE NON INTERRUPT PACKAGE
:AC1,AC2 SAVED
:AMES?" PRINTS ASCII MESSAGES AS SPECIFIED BY ASSEMBLER
:CLF?" PRINTS A CARRIAGE RETURN
:POC?" PRINTS C(1) IN OCTAL
:ZOC?" PRINTS C(1) IN OCTAL, LEADING ZEROS SUPPRESSED,
:THE ABOVE THREE ARE FOLLOWED BY A TAB UNLESS LOCATION PTB? IS
:ALTERED IN WHICH CASE CONTENTS OF PTB? WILL BE PRINTED AFTER
:THE NUMBER.
:TI0?" ACCEPTS OCTAL, AND
:TI1?" ACCEPTS DECIMAL SINGLE PRECISION SIGNED INTEGERS
:INTO AC1 FROM THE TTY. LEADING NULLS, TABS,
:AND SPACES ARE IGNORED. A 16 BIT UNSIGNED INTEGER IS
:FORMED, THEN NEGATED IF A MINUS SIGN IS TYPED.
:EXIT AT CALL+1 IF INPUT ERROR WITH ACO=BAD CHARACTER.
:(NOT A LEGAL DIGIT OR TERMINATING CHARACTER)
:EXIT AT CALL+2 UPON TERMINATING CHARACTER
:WITH ACO=0, 0, 40, 12, 55
:FOR NULL, TAB, SPACE, CARRIAGE RETURN, COMMA
:THE ABOVE WAIT FOR TTY DONE, THEN CLEAR TTY.
:CHC?" PRINTS ASCII CHARACTER IN C(0)R; C(0) MUST BE 0.
:EXITS CALL +2 IF C(0)R=0; SIMULATES TAB
:TYPE?" PRINTS C(0)R. EXITS AT CALL+1. REPLACE "TYPE?" WITH
:INTERRUPT TYP? IF DESIRED.
:TPS?" PRINTS A SPACE AND EXITS AT CALL+1 WITH ACO = 40
3, MES?R
1, PAC?1
2, PAC?2
MES?R
: C(2) POINTS TO MESSAGE
: A 8 BIT MASK
: C(2) = DATA WORD
: C(0) = DATA CHARACTER RIGHT
: INC TO NEXT WORD
: FLTP MASK
: PRINT
: ANOTHER
: LAST
: PRINT C(1) IN OCTAL
: PRINT C(1) IN DECIMAL
: BOTH ENTRIES PRINT NUMBER
02304 054523 MES?: STA
02305 044865 STA
02306 050465 STA
02307 010520 ISZ
02310 031400 LDA
02311 024463 LDA
02312 021000 LDA
02313 125112 MOVL#
02314 123701 ANDS
02315 123401 AND
02316 151400 INC
02317 124000 COM
02320 004455 JSR
02321 000771 JMP
02322 000402 JMP
02323 004452 CHC?T
02324 024446 JSR
02325 030446 LDA
02326 063511 LDA
02327 000777 TTO
02328 060211 SKPBZ
02329 060211 JMP
02330 060211 TIO
02331 002476 NIOC
02332 102401 JMP
02333 020562 SUB
02334 050437 LDA
02335 030564 POCT?
02336 000404 LDA
02337 050434 JMP
02338 030571 STA
02339 102400 LDA
02342 054465 STA

```

```

0041 EXMEM
01 02343 044427 STA 1,PAC?1
02 02344 040425 STA 0,ZSU?P
03 02345 050401 STA 2,+*1
04 02346 000000 DEC?OC: 0
05 02347 010777 ISZ *-1
06 02350 020570 LDA 0,PTB?
07 02351 151005 MOV 2,R2,SNR
08 02352 000751 JMP PLS?T
09 02353 034416 LDA 3,ZSU?P
10 02354 102400 SUB 0,0
11 02355 146452 DEO?T: 0
12 02356 000405 JMP DEC?P
13 02357 146400 SUB 2,1
14 02360 034535 LDA 3,PC6?0
15 02361 101400 INC 0,0
16 02362 000775 JMP DEO?T
17 02363 151235 DEC?P: MOVZR#
18 02364 034531 LDA 3,PC6?0
19 02365 054404 STA 3,ZSU?P
20 02366 163004 ADD 3,0,SZR
21 02367 004406 JSR CHC?T
22 02370 000756 JMP DEC?OC
23
24 02371 000000 ZSU?P: 0
25 02372 000000 PAC?1: 0
26 02373 000000 PAC?2: 0
27 02374 000377 P3?T: 377
28 02375 054831 STA MOVZR#
29 02376 101315 JMP 1,3
30 02377 001401 LDA 3,PC1?1
31 02400 034425 JSR 0,3,SNR
32 02401 116415 SUB#
33
34 02402 000403 JMP CHA?3
35 02403 004537 JSR TYP?P
36 02404 024222 JMP @CHR?T
37 02405 004534 JSR TPB?
38 02406 020416 LDA 0,CHR?2
39 02407 034410 LDA 0,PC?7
40 02410 163404 AND 3,0,SZR
41 02411 000774 JMP CHA?3
42
43 02412 040412 STA 0,CHR?2
44 02413 002413 JMP @CHR?T
45
46 02414 054413 CLF?F: STA 3,MES?R
47 02415 044755 STA 1,PAC?1
48 02416 050755 STA 2,PAC?2
49 02417 020561 LDA 0,PC1?5

```

```

10042 EXMEM
01 02420 000755 JSR CHC?T
02 02421 020514 LDA 0,PC1?2
03 02422 000701 JMP PLS?T
04
05 02423 000007 PC?F: 7
06 02424 000000 CHR?Z: 0
07 02425 000011 PC1?1: 11
08 02426 000000 CHR?T: 0
09 02427 000000 WES?R: 0
10 02430 020505 TIC?F: LDA
11 02431 004511 JSR TYP?P
12 02432 010775 TIX?F: ISZ
13 02433 024737 TIR?F: LDA
14 02434 054735 LDA 3,ZSU?P
15 02435 175102 MOV# 3,3,SZC
16 02436 124400 MEG 1,1
17 02437 000666 JMP PEX?T+1
18
19 02440 102121 ADCZL 0,0,SKP
20 02441 102440 SUBO 0,0
21 02442 054765 STA 3,MES?R
22 02443 050730 STA 2,PAC?2
23 02444 030471 LDA 2,PC1?2
24 02445 115000 ADD 0,2
25 02446 102440 SUBO 0,0
26 02447 040722 STA 0,ZSU?P
27
28 02450 034721 TIS?F: LDA
29 02451 175004 MOV 3,3,SZR
30 02452 000760 JMP TIX?F
31 02453 054717 STA 3,PAC?1
32 02454 063610 SKPDN TTI
33 02455 000777 JMP *-1
34 02456 060610 CIAC 0,TTI
35 02457 004716 JSR CHC?T
36 02460 034436 LDA 3,PC8?0
37 02461 116414 SUB# 0,3,SZR
38 02462 101015 MOV# 0,0,SNR
39 02463 000765 JMP TIS?F
40 02464 024434 LDA 1,TIM?2
41 02465 106015 ADC# 0,1,SNR
42 02466 060744 JMP TIX?F
43 02467 106424 SUBZ 0,1,SZR
44 02470 000405 JMP TIM?S
45 02471 034700 LDA 3,ZSU?P
46 02472 177200 ADDR 3,3
47 02473 054676 STA 3,ZSU?P
48 02474 000760 JMP TIM?+1
49 02475 136415 SUB# 1,3,SNR
50 02476 000732 JMP TIC?F
51 02477 024420 LDA 1,TIM?1
52 02500 107022 ADDZ 0,1,SZC
53 02501 146513 SUBL# 2,1,SNR
54 02502 000731 JMP TIR?F
55 02503 010666 ISZ ZSU?P
56 02504 020666 LDA 0,PAC?1
57 02505 101120 MOVZL 0,0
58 02506 115120 MOVZL 0,3
59 02507 175120 MOVZL 3,3
60 02510 137000 ADD 1,3

```

```

10042 EXMEM
01 02420 000755 JSR CHC?T
02 02421 020514 LDA 0,PC1?2
03 02422 000701 JMP PLS?T
04
05 02423 000007 PC?F: 7
06 02424 000000 CHR?Z: 0
07 02425 000011 PC1?1: 11
08 02426 000000 CHR?T: 0
09 02427 000000 WES?R: 0
10 02430 020505 TIC?F: LDA
11 02431 004511 JSR TYP?P
12 02432 010775 TIX?F: ISZ
13 02433 024737 TIR?F: LDA
14 02434 054735 LDA 3,ZSU?P
15 02435 175102 MOV# 3,3,SZC
16 02436 124400 MEG 1,1
17 02437 000666 JMP PEX?T+1
18
19 02440 102121 ADCZL 0,0,SKP
20 02441 102440 SUBO 0,0
21 02442 054765 STA 3,MES?R
22 02443 050730 STA 2,PAC?2
23 02444 030471 LDA 2,PC1?2
24 02445 115000 ADD 0,2
25 02446 102440 SUBO 0,0
26 02447 040722 STA 0,ZSU?P
27
28 02450 034721 TIS?F: LDA
29 02451 175004 MOV 3,3,SZR
30 02452 000760 JMP TIX?F
31 02453 054717 STA 3,PAC?1
32 02454 063610 SKPDN TTI
33 02455 000777 JMP *-1
34 02456 060610 CIAC 0,TTI
35 02457 004716 JSR CHC?T
36 02460 034436 LDA 3,PC8?0
37 02461 116414 SUB# 0,3,SZR
38 02462 101015 MOV# 0,0,SNR
39 02463 000765 JMP TIS?F
40 02464 024434 LDA 1,TIM?2
41 02465 106015 ADC# 0,1,SNR
42 02466 060744 JMP TIX?F
43 02467 106424 SUBZ 0,1,SZR
44 02470 000405 JMP TIM?S
45 02471 034700 LDA 3,ZSU?P
46 02472 177200 ADDR 3,3
47 02473 054676 STA 3,ZSU?P
48 02474 000760 JMP TIM?+1
49 02475 136415 SUB# 1,3,SNR
50 02476 000732 JMP TIC?F
51 02477 024420 LDA 1,TIM?1
52 02500 107022 ADDZ 0,1,SZC
53 02501 146513 SUBL# 2,1,SNR
54 02502 000731 JMP TIR?F
55 02503 010666 ISZ ZSU?P
56 02504 020666 LDA 0,PAC?1
57 02505 101120 MOVZL 0,0
58 02506 115120 MOVZL 0,3
59 02507 175120 MOVZL 3,3
60 02510 137000 ADD 1,3

```

```

!PRINT CARRIAGE AND LF
!THEN TAB TO NEXT POSITION
!*LDA 2, TABLE" INSTRUCTION
!IF TABLE ENTRY=0
!EXIT WITH A SPACE/TAB
!ZEROS SUPPRESS STUF
!FORM THE DIGIT
!C(0)=DIGIT
!MAKE ASCII
!GET NEXT DIGIT
!PRINT C(0) RIGHT
!RETURN +2 IF NULL
!AC3 = 11
!SKIP IF A TAB IS NOT TO
!BE SIMULATED
!PRINT IT
!EXIT
!PRINT A SPACE
!AC3 = 7
!SIMULATE A TABE WITH 1
!TO 7 SPACES
!SAVE RETURN
!SPACE, TAB, OR NULL
!CDMMA
!OR
!MINUS
!IF NEITHER THEN GO TO TIM?
!COMPLEMENT SIGN
!IS IT A CARRIAGE RETURN?
!IF CR THEN GO TO TIC?
!SKIP IF NOT A DIGIT
!SKIP IF DIGIT
!OUT OF LEADING SPACES
!8 OLD PAC?1'S + NEW DIGIT

```

0043 EXMEM  
 01 02511 145220  
 02 02512 125232  
 03 02513 117000  
 04 02514 000737  
 05 02515 000060  
 06 02516 000040  
 07 02517 17720  
 08 02520 000055  
 09 02521 030554  
 10 02522 100000  
 11 02523 010000  
 12 02524 001000  
 13 02525 000100  
 14 02526 000010  
 15 02527 000001  
 16 02530 000000  
 17 02531 030564  
 18 02532 023420  
 19 02533 001750  
 20 02534 000144  
 21 02535 000012  
 22 02536 000001  
 23 02537 000000  
 24 02538 000000  
 25 02539 000010  
 26 02540 000011  
 27 02541 020755  
 28 02542 054437  
 29 02543 034434  
 30 02544 175004  
 31 02545 034264  
 32 02546 175120  
 33 02547 177122  
 34 02550 000404  
 35 02551 063511  
 36 02552 000777  
 37 02553 061111  
 38 02554 143770  
 39 02556 175120  
 40 02557 177123  
 41 02560 000405  
 42 02561 061117  
 43 02562 063517  
 44 02563 000777  
 45 02564 060217  
 46 02565 034411  
 47 02566 116043  
 48 02567 024727  
 49 02570 162432  
 50 02571 010633  
 51 02572 034406  
 52 02573 116445  
 53 02574 054630

0044 EXMEM  
 01 02575 002404  
 02 02576 000177  
 03 02577 177777  
 04 02578 000015  
 05 02579 000000  
 06 02580 000000

JMP  
 177  
 -1  
 15  
 0

@TYP2R

|    |         |            |                           |  |  |  |  |  |  |
|----|---------|------------|---------------------------|--|--|--|--|--|--|
| 01 | MOVZR   | 2,1        |                           |  |  |  |  |  |  |
| 02 | MOVZR#  | 1,1,8ZC    |                           |  |  |  |  |  |  |
| 03 | ADD     | 0,3        |                           |  |  |  |  |  |  |
| 04 | JMP     | TIW?       |                           |  |  |  |  |  |  |
| 05 | PC420:  |            |                           |  |  |  |  |  |  |
| 06 | PC430:  |            |                           |  |  |  |  |  |  |
| 07 | PC430:  |            |                           |  |  |  |  |  |  |
| 08 | TIW2:   |            |                           |  |  |  |  |  |  |
| 09 | TIW2:   |            |                           |  |  |  |  |  |  |
| 10 | OC174B: | LDA        | 2,.*+1,--DEC?0C           |  |  |  |  |  |  |
| 11 |         |            |                           |  |  |  |  |  |  |
| 12 |         |            |                           |  |  |  |  |  |  |
| 13 | C10?0:  |            |                           |  |  |  |  |  |  |
| 14 |         |            |                           |  |  |  |  |  |  |
| 15 |         |            |                           |  |  |  |  |  |  |
| 16 |         |            |                           |  |  |  |  |  |  |
| 17 | DEC?7B: | LDA        | 2,.*+1,--DEC?0C           |  |  |  |  |  |  |
| 18 | .ROX 10 |            |                           |  |  |  |  |  |  |
| 19 |         |            |                           |  |  |  |  |  |  |
| 20 |         |            |                           |  |  |  |  |  |  |
| 21 |         |            |                           |  |  |  |  |  |  |
| 22 | PC1?2:  |            |                           |  |  |  |  |  |  |
| 23 |         |            |                           |  |  |  |  |  |  |
| 24 |         |            |                           |  |  |  |  |  |  |
| 25 | .ROX 8  |            |                           |  |  |  |  |  |  |
| 26 |         |            |                           |  |  |  |  |  |  |
| 27 | PTB?:   |            |                           |  |  |  |  |  |  |
| 28 |         |            |                           |  |  |  |  |  |  |
| 29 | TPS?:   | LDA        | 0,PC4?0                   |  |  |  |  |  |  |
| 30 | STA     | 3,TYP?R    | !PREPARE TO PRINT A SPACE |  |  |  |  |  |  |
| 31 | TYP?:   | LDA        | 3,INT?                    |  |  |  |  |  |  |
| 32 |         |            |                           |  |  |  |  |  |  |
| 33 |         |            |                           |  |  |  |  |  |  |
| 34 |         |            |                           |  |  |  |  |  |  |
| 35 |         |            |                           |  |  |  |  |  |  |
| 36 |         |            |                           |  |  |  |  |  |  |
| 37 |         |            |                           |  |  |  |  |  |  |
| 38 |         |            |                           |  |  |  |  |  |  |
| 39 | SKPBZ   | TT0        |                           |  |  |  |  |  |  |
| 40 | JMP     | .-1        |                           |  |  |  |  |  |  |
| 41 | DOAS    | 0,TT0      |                           |  |  |  |  |  |  |
| 42 | PLP?T:  | ANDI 177,0 |                           |  |  |  |  |  |  |
| 43 |         |            |                           |  |  |  |  |  |  |
| 44 |         |            |                           |  |  |  |  |  |  |
| 45 |         |            |                           |  |  |  |  |  |  |
| 46 |         |            |                           |  |  |  |  |  |  |
| 47 |         |            |                           |  |  |  |  |  |  |
| 48 |         |            |                           |  |  |  |  |  |  |
| 49 | SKPBZ   | LPT        |                           |  |  |  |  |  |  |
| 50 | JMP     | .-1        |                           |  |  |  |  |  |  |
| 51 | NI0C    | LPT        |                           |  |  |  |  |  |  |
| 52 | TPR?T:  | LDA        | 3,PI1?77                  |  |  |  |  |  |  |
| 53 |         | ADCO       | 0,3,SNC                   |  |  |  |  |  |  |
| 54 |         | LDA        | 3,PC4?0                   |  |  |  |  |  |  |
| 55 |         | SUBZ#      | 3,0,SZC                   |  |  |  |  |  |  |
| 56 | ISZ     | CHR?Z      |                           |  |  |  |  |  |  |
| 57 | LDA     | 3,PC1?95   |                           |  |  |  |  |  |  |
| 58 | SUB0    | 0,3,SMR    |                           |  |  |  |  |  |  |
| 59 | STA     | 3,CHR?Z    |                           |  |  |  |  |  |  |
| 60 |         |            |                           |  |  |  |  |  |  |

!SKIP IF OCTAL MODE  
 !ADD 2 OLD PAC?1'S  
 !CHECK THE SWITCHES  
 !SKIP IF IT WAS A RUBOUT  
 !SKIP IF IT WAS NOT A "CR"  
 !CLR PARITY BIT  
 !CHECK FOR BIT 5  
 !SKIP IF THE OUTPUT IS  
 !REQUIRED ON THE LPT  
 !OUTPUT THE CHARACTER TO LPT  
 !WAIT FOR LPT  
 !CLEAR THE DONE FLAG FOR LPT  
 !AC3 = 177  
 !SKIP IF IT WAS A RUBOUT  
 !AC3 = 40  
 !SKIP FOR NON PRINTING  
 !CHARACTERS  
 !AC3 = 15  
 !SKIP IF IT WAS NOT A "CR"  
 !CLEAR MORZ POS

10045 EXMEM

```

01 :FILENAME= SWRPACK
02
03 :THIS PACKAGE IS USED TO CHANGE THE SETTINGS OF LOCATION
04 :$WREG" OF PAGE 0. THE PROGRAM CONTROL SHOULD ENTER "INP?"
05 :WITH AC3 HAVING THE RETURN ADDRESS. THE CHARACTER INPUTED
06 :BY THE OPERATOR IS ECHOED AFTER A "CR". IF THE COMMAND IS
07 :NOT A LEGAL ONE THEN THE CONTROL IS RETURNED WITHOUT DOING
08 :ANYTHING, OTHERWISE ONE OF THE FOLLOWING COMMANDS IS
09 :EXECUTED:
10 :KEYS I-9 AND A-F ARE USED TO COMPLEMENT THE CURRENT VALUE
11 :OF BITS 1-15 OF "SWREG". IF ONE OF THESE KEYS IS HIT THE
12 :CORRESPONDING BIT OF "SWREG" IS COMPLEMENTED AND THE CONTROL
13 :IS RETURNED TO THE STATE PROGRAM HAD BEFORE HITTING THE KEY
14 :TYPING OF A "0" WILL LOCK THE PROGRAM IN A SWITCH MODIFICATION
15 :MODE IN WHICH CASE MORE THAN ONE BITS CAN BE CHANGED BEFORE
16 :THE CONTROL IS ALLOWED TO RETURN TO THE MAIN PROGRAM. HITTING
17 :THE "CS" KEY WILL UNLOCK THE PROGRAM FROM THIS MODE.
18 :A "0" THIS COMMAND GIVEN AT ANY TIME WILL RESET THE "SWREG"
19 :TO DEFAULT MODE (ALL ZEROS) AND RESTART THE PROGRAM AT ADD.
20 :STORED IN LOCATION "INS?".
21 :R" THIS COMMAND GIVEN AT ANY TIME WILL RESTART THE PROG.
22 :AT ADDRESS STORED IN LOCATION "INS?".
23 :M" THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE CURRENT
24 :OPERATING MODES.
25 :BEFORE THE CONTROL IS RETURNED TO THE MAIN PROGRAM BIT 0 WILL
26 :BE SET IF ANY OF THE OTHER BITS OF "SWREG" IS SET, OTHERWISE
27 :IT WILL BE CLEARED
28
29 :THIS PACKAGE EXITS WITH C(AC3) = CHARACTER TYPED IN (PARITY
30 :STRIPPED).
31
32 02602 000136 INI?36: 136
33 02603 000104 INI?04: 104
34 02604 000122 INI?22: 122
35 02605 000033 INC?33: 33
36 02606 000000 INL?M: 0
37 02607 000000 INR?T: 0
38 02610 000000 INS?0: 0
39 02611 000000 INS?1: 0
40 02612 000000 INS?2: 0
41 02613 000000 INS?3: 0
42 02614 000000 INS?C: 0
43 02615 000202 INR?T: STRT2
44 02616 000000 SVP?TB: 0
45
46 02617 050770 INP?: STA 3,INS?T
47 02620 040770 STA 0,INS?0
48 02621 040770 STA 1,INS?1
49 02622 050770 STA 2,INS?2
50 02623 175200 MOV# 3,3
51 02624 054770 STA 3,INS?C
52 02625 176400 SUB 3,3
53 02626 054751 STA 3,INT?
54 02627 066101 JSR @ICLF?
55 02630 040750 STA 0,INL?K
56 02631 040610 INO?: DIAC 0,IT
57 02632 034744 LDA 3,PI?T?
58 02633 163400 AND 3,0
59 02634 024744 LDA 1,PC1?S
60 02635 106415 SUB# 0,1,SNR

```

0046 EXMEM

```

01
02 02636 000516 JMP INR?
03 02637 040754 STA 0,INS?3
04 02640 024665 LDA 1,C10?0
05 02641 034674 LDA 3,PC1?2
06 02642 116414 SUB# 0,3,SZR
07 02643 034742 LDA 3,INC?3
08 02644 164452 SUBO# 3,0,SZC
09
10 02645 000471 JMP INS?
11 02646 004674 JSR TYP?
12 02647 034646 LDA 3,PC6?0
13 02650 152620 SUBZR 2,2
14 02651 116405 SUB 0,3,SNR
15
16 02652 000447 JMP INI?
17
18 02653 151221 IN2?: MOVZR 2,2,SKP
19 02654 128520 SUBZL 1,1
20 02655 175405 INC 3,3,SKM
21 02656 000445 JMP IN3?+2
22 02657 147415 AND# 2,1,SNR
23
24 02660 000773 JMP IN2?
25 02661 106400 SUB 0,1
26
27 02662 135000 MOV 1,3
28 02663 151225 MOVZR 2,2,SNR
29 02664 000444 JMP IN4?
30 02665 024713 LDA 1,PC1?S
31 02666 167004 ADD 3,1,SZR
32 02667 000765 JMP INR?+1
33
34 02670 034650 INM?: LDA 3,PTB?
35 02671 054725 STA 3,SVP?TB
36 02672 034624 LDA 3,PC4?0
37
38 02673 054645 STA 3,PTB?
39 02674 006101 JSR @ICLF?
40 02675 006104 JSR @IPDE?
41 02676 034637 LDA 3,PC1?2
42 02677 125400 INC 1,1
43 02700 166452 SUBO# 3,1,SZC
44
45 02701 004640 JSR TPS?
46 02702 101220 MOVZR 0,0
47 02703 128414 SUB# 1,0,SZR
48 02704 000771 JMP INM?+5
49 02705 006101 JSR @ICLF?
50 02706 030264 LDA 2,SWREG
51 02707 151140 MOVOL 2,2
52 02710 128560 SUBCL 1,1
53 02711 006103 JSR @IZOC?
54 02712 004627 JSR TPS?
55 02713 151124 MOVZL 2,2,SZR
56
57 02714 000774 JMP -4
58 02715 006101 JSR @ICLF?
59 02716 034700 LDA 3,SVP?TB
60 02717 054621 STA 3,PTB?

```

```

:WAS NOT "CR"
:SAVE CHARACTER
:AC1 = 100
:AC3 = 12
:SKIP IF IT IS A LINE FEED
:SKIP IF AC0 IS EQUAL OR MORE
:THAN AC3
:ECHO THE CHARACTER
:AC2 = 60
:AC2 = 100000
:SKIP IF THE DIGIT TYPED WAS
:NOT 0
:SHIFT AC2 TO RIGHT
:AC1 = 1
:STAY IN LOOP UNTIL ALL BITS
:OF SWREG ARE CHECKED
:WHEN THE CONTROL COMES HERE
:FOR THE FIRST TIME AC1 = 100
:AC1 = 15
:SKIP IF THE COMMAND WAS "M"
:SAVE PTR?
:PREPARE TO PRINT A SPACE
:AFTER EACH NUMBER
:TYPE A "CR" AND "LF"
:PRINT THE CONTENTS OF AC1
:AC3 = 12
:SKIP IF AC1 IS GREATER OR EQUAL
:TO AC3
:TYPE A SPACE
:AC0 = 20
:SKIP AFTER TYPING # 15
:AC2 HAD SWITCH SETTINGS
:BRING THE CARRY BIT IN AC1
:TYPE THE CONTENTS OF AC1
:TYPE A SPACE
:SKIP AFTER TYPING ALL THE 16
:BITS
:PRESTONE PTB?

```

```

0047 EXMEM
01 02720 009410
02
03 02721 176000 IN3?:
04 02722 054664
05 02723 024264
06
07 02724 133414
08 02725 146401
09 02726 147000
10 02727 044264 IN4?:
11 02730 010656 IN4?:
12
13 02731 000423
14 02732 014654
15 02733 063610 SKPDN
16 02734 000777
17 02735 000674
18
19 02736 107000 IN5?:
20
21 02737 020643
22 02740 006432
23 02741 121000 MOV
24 02742 006430 JSR
25 02743 034641 LDA
26 02744 114405 SUB
27 02745 009405 JMP
28 02746 034635 LDA
29 02747 118404
30 02750 000760 JMP
31 02751 054264 STA
32
33 02752 034643 IN6?:
34
35 02753 054634
36 02754 006101 INR?:
37 02755 010622
38 02756 030264
39 02757 176220
40 02760 173404
41
42 02761 172000
43 02762 050264
44 02763 020631
45 02764 101100 MOVL
46 02765 020623
47 02766 024623
48 02767 030623
49 02770 034623
50 02771 002616
51
52 02772 002542 I1Y?: TYP?

10048 EXMEM
01
02
03
04
05 02773 054410
06 02774 034444
07 02775 117000
08 02776 051400
09 02777 034436
10 03000 117000
11 03001 045400
12 03002 002401
13 03003 000000 EI.S3: 0
14
15
16
17
18 03004 020437
19 03005 030433
20 03006 041000
21 03007 151400
22 03010 145300
23 03011 125224
24 03012 000774
25 03013 024427
26 03014 032425
27 03015 050420
28 03016 102000
29 03017 041000
30 03020 151400
31 03021 107004
32 03022 000775
33 03023 052416
34 03024 102400
35 03025 040005
36 03026 020502
37 03027 040043
38 03030 020406
39 03031 040310
40 03032 020405
41 03033 040311
42 03034 001400
43 03035 000000 LMSKS: 0
44 03036 005210 TT.II
45 03037 005222 TT.TO
46 03040 000300 LK300: 300
47 03041 001060 LC.K1: LSETB
48 03042 000100 LC100: 100

;EINTP - ENTER INTERRUPT SERVICE PARAMETERS
;(AC0)=DEV# JSR @EINTS
;(AC1)=MSKO
;(AC2)=ADDRESS OF DEV INTR SERV
EINTP: STA 3,EI.S3
LDA 3,LK300
ADD 0,3
STA 2,0,3
LDA 3,LMSKS
ADD 0,3
STA 1,0,3
JMP @EI.S3
EI.S3: 0

;LCINT - INITIALIZE INTERRUPT SERVICE TABLES
;VECTOR ADDRESSES ARE SET TO ILLEGAL INT
;AND MSKO'S ARE SET TO -1
;LSTMP IS SET TO START 1 AFTER MSKO'S
LCINT: LDA 0,LILLI
LDA 2,LK300
STA 0,0,2
INC 2,2
MOVZ 2,1
MOVZ 1,1,SZ
JMP LCINT+2
LDA 1,LC100
LDA 2,ALC.K1
STA 2,LMSKS
ADC 0,0
STA 0,0,2
INC 2,2
ADD 0,1,SZ
JMP -3
STA 2,ALC.K1
SUB 0,0
STA 0,MSKRG
LDA 0,USTKF
STA 0,USTKP+3
LDA 0,LMSKS+1
STA 0,300+TTI
LDA 0,LMSKS+2
STA 0,300+TTO
JMP 0,3

;FILL SERVICE
;VECTORS WITH
;AORS ILLEGAL INTR
;=100
;RESERVE 100 WORDS
;100 FOR MSKO
;STRT INT MSK = 0
;IN CASE STACK OFLOW

```

```

10049 EXMEM
01 03043 003044 LILLI+1
02 03044 107110 PSH 0,1
03 03045 061477 INTA 0
04 03046 024402 LDA 1,0,+2
05 03047 123001 ADD 1,0,SKP
06 03050 060200 NIOC 0
07 03051 040401 STA 0,+1
08 03052 060200 NIOC 0
09 03053 060403 DIA 0,MAP
10 03054 061003 DDA 0,MAP
11 03055 123210 POP 1,0
12 03056 060177 INTEN
13 03057 002000 JMP @0
14
15
16 03060 063077 IOVAL: HALT ;IO/VALIDITY TRAP
17
18 ;POWER FAIL SAVE AND RESTART ROUTINES
19 03061 117110 PSH 0,3 ;BUILD A STATE BLOCK
20 03062 102660 SUBCR 0,0
21 03063 024000 LDA 1,0
22 03064 123000 ADD 1,0
23 03065 103110 PSH 0,0
24 03066 060403 DIA 0,MAP
25 03067 103110 PSH 0,0 ;SAVE STATE OF MAP
26 03070 020404 LDA 0,0,PWRU-1
27 03071 040000 STA 0,0 ;FOR POWER UP
28 03072 063777 SKPDZ CPU
29 03073 000777 JMP *-1 ;WAIT UP
30 03074 002077 JMP @PWRUP

10050 EXMEM
01
02 ;POWER OK RESTART THE WORLD
03 03075 102400 PWRU: SUB 0,0
04 03076 040073 STA 0,TIMSN
05 03077 040073 STA 0,PEAIS
06 03100 020066 LDA 0,MPSWT
07 03101 101005 MOV 0,0,SNR ;SKP=MAP IN USE
08 03102 000406 JMP +*6 ;NOT USING IT
09 ;RELOAD MAP
10 03103 020065 LDA 0,NEWMP ;WHICH MAP?
11 03104 101004 MOV 0,0,SZR ;SKP=OLD MAP
12 03105 000402 JMP +*2 ;NEW MAP DOESN'T USE DEV. PROTECTION TBL'S
13 03106 006416 JSR @PWRK ;DEVICE PROTECT
14 03107 006416 JSR @PWRK+1 ;AND USER MAP
15 ;HAS RTC ENABLED?
16 03110 022416 LDA 0,@PWRK+2
17 03111 126520 SUBZL 1,1
18 03112 101005 MOV 0,0,SNR ;SKP=RTC NOT
19 03113 065114 DDAS 1,RTC
20 ;NOW CHECK ERROR CORRECTION ENABLE
21 03114 022413 LDA 0,@PWRK+3
22 03115 124010 ADI 2,+1
23 03116 101005 MOV 0,0,SNR ;SKP =NOT ERCC ENABLE
24 03117 065102 DDAS 1,ERCC
25 ;RESTORE AND RETURN TO WHERE INTERRUPTED
26 03120 103210 POP 0,0 ;MAP STATUS
27 03121 061005 DDA 0,MAP
28 03122 060177 INTEN
29 03123 107710 POPB
30 03124 000705 PWRK: LDDMP
31 03125 001556 LDMAP
32 03126 005011 RT,00+2
33 03127 003405 EC,00+2
34
35
36 ;DNCHK - USR STACK OVERFLOW
37 03130 003131 USTKF: +*1
38 03131 063077 DNCHK: HALT

```

```

10051 EXMEM
01 03132 005215 TXT.1: ;TXTE (<15><12>
02 03133 141501 AC.8: (
03 03137 005215 TXT.2: ;TXTE (<15><12>
04 03140 141523 SCRL0/HI (
05 03145 005215 TXT.4: ;TXTE (<15><12>
06 03146 044120 PHYS: DECI LOGICAL(
07 ;NOLDC 0
08 ;SET UP DEVICE TABLES FOR
09 ;INTERRUPT DISPATCHING VIA THE VECTOR INST
10 03160 024403 NEWID: LDA 1,NEWKI ;ADRS @ VECTORS
11 03161 044001 STA 1,1 ;FOR INTR'S
12 03162 001400 JMP 0,3
13 03163 003164 NEWKI: FRSTI ;FIRST LEVEL INTR
14
15 ;FIRST LEVEL INTERRUPT DISPATCH
16 03164 063777 FRSTI: SVPDZ CPU ;WAS IT A PWR FAIL INTR?
17 03165 002076 JMP @PWRDN ;YEP GO DIRECT
18 03166 061777 VCT VTBL
19 000300
20 000300 ;OUSR VTBL=300

10052 EXMEM
01 ;LPRSL-LINKER PROGRAM SELECT 00,
-8.
ILCTB+1 ;INTERLEAVE FACTOR TABLE
LAUTO
LPRSL: STA 3,LPSV3
SUB 0,0
STA 0,LPRGN ;FIRST TXT ADRS
LDA 0,LPT1
JSR @ERRTX ;GET HIGHK PHYS.
LDA 1,@LPHIK ;TOTAL # 'K'S
INC 1,1 ;PRINT SUPR 0'S
JSR @LPDEC
LDA 1,MPSWT ;WHICH MAP?
LDA 2,NEWMP
MOV 2,2,SNR
JMP +3
LDA 0,MP2TX ;MMPU1 TXT
JMP +2
LDA 0,MPYTX ;MAP EXIST TEXT
MOV 1,1,SNR ;SKP IF MAP EXIS
LDA 0,MPPTX ;NO MAP TEXT
JSR @ERRTX
JSR @LCRLF
ELEF 0,JHOR2,0
JSR @ERRTX ;DESCRIBE MEMORY SYSTEM
EJSR JMEXT
LDA 1,@LPRSL-1
LDA 0,LPT2 ;SKP IF AUTO STRT
MOV 1,1,JSZR ;NOT AUTO USE OTHR HDR
LDA 0,LPT3
JSR @ERRTX
LDA 0,LPT4 ;PRT PRG HDR
JSR @ERRTX ;CARRET LFEED TST 0
JSR @LCRLF

```





```

10053 EXMEM
01 03234 020447 LPRLP: LDA 0,LPRCN
02 03235 03443 ADD 0,3
03 03236 117000 LDA 2,0,3
04 03237 051400 MOV 2,0,SNR
05 03240 151005 JMP @LPSV3
06 03241 002841 MOV 0,1
07 03242 105000 LDA 0,2,2
08 03243 021002 MOV 0,0,SZR
09 03244 101004 JMP LPR1E
10 03245 000431 STA 2,LPIDX
11 03246 050442 JSR @LZOCI
12 03247 006103 LDA 0,LPIDX
13 03250 020440 LDA 1,LPR10
14 03251 024436 ADD 1,0
15 03252 125000 JSR @ERRTX
16 03253 006212 LDA 1,@LPRSL-1
17 03254 026717 MOV 1,1,SNR
18 03255 125005 JMP LPR1E-1
19 03256 000417

:CUR PROG #
:SKP IF NOT LAST
:SET PRG WAIT SW
:DEV MUST NOT EXIST
:TYPE PRG #
:CALC ADNS DESC TXT
:SKP IS LET OPR SELECT
:CR/LF AND DO NXT PRG

10054 EXMEM
01
02
03 03257 060277 INTDS
04 03260 063610 SKPDN TTI
05 03261 000777 JMP -1
06 03262 064610 DIAC 1,TTI
07 03263 030422 LDA 2,LPR77
08 03264 147400 AND 2,1
09 03265 030421 SUB# 2,1,SNR
10 03266 146415 JMP LPR1E-1
11 03267 000406 LDA 2,LPIDX
12 03270 030420 ADC 0,0
13 03271 102000 STA 0,2,2
14 03272 041002 LDA 0,LPIDX
15 03273 020503 JSR @ERTY
16 03274 006212 JSR @LCRLF
17 03275 006101 JSR @LRLP
18 03276 010405 LPR1E: ISZ LPRGN
19 03277 000755 JMP LPRLP
20 03300 000175 LPLZM: LMAX
21 03301 003370 LPR74: LPR4T
22 03302 000000 LPSV3: 0
23 03303 000000 LPRGN: 0
24 03304 001117 LPHIK: MIGHK
25 03305 000077 LPR77: 77
26 03306 000040 LPR40: 40
27 03307 000010 LPR10: 10
28 03310 000000 LPIDX: 0
29 03311 003312 LPR71:
30 03312 005215 .TXTE (<15><12>EXMEMX 04 #1K'S=(
31 03325 003326 MPATX:
32 03326 046640 .TXTE ( MHPUI(
33 03331 003332 NMPTX:
34 03332 047240 .TXTE ( NO MAP(
35 03333 003337 MP2TX:
36 03337 046640 .TXTE I MHPUI!
37 03343 005344 LPR72:
38 03344 005215 .TXTE (<15><12>RUN LIST(
39 03352 003353 LPR73:
40 03353 005215 .TXTE (<15><12>SFA. SELS-OTHERS DEL.(
41 03370 005215 LPR4T: .TXTE (<15><12>PRG# DES.(
42 03376 003377 LP0TX:
43 03377 042240 .TXTE ( DEL.(
44 000000 .NOLOC 0

```

10055 EXMEM

```

02 .L1L1 ERCCR
03 .L1L1 ERCCR
04 .L1L1 ERCCR
05 .L1L1 ERCCR
06 .L1L1 ERCCR
07 .L1L1 ERCCR
08 .L1L1 ERCCR
09 .L1L1 ERCCR
10 .L1L1 ERCCR
11 .L1L1 ERCCR
12 .L1L1 ERCCR
13 .L1L1 ERCCR
14 .L1L1 ERCCR
15 .L1L1 ERCCR
16 .L1L1 ERCCR
17 .L1L1 ERCCR
18 .L1L1 ERCCR
19 .L1L1 ERCCR
20 .L1L1 ERCCR
21 .L1L1 ERCCR
22 .L1L1 ERCCR
23 .L1L1 ERCCR
24 .L1L1 ERCCR
25 .L1L1 ERCCR

```

10056 EXMEM

```

01 .L1L1 ERCCR
02 .L1L1 ERCCR
03 .L1L1 ERCCR
04 .L1L1 ERCCR
05 .L1L1 ERCCR
06 .L1L1 ERCCR
07 .L1L1 ERCCR
08 .L1L1 ERCCR
09 .L1L1 ERCCR
10 .L1L1 ERCCR
11 .L1L1 ERCCR
12 .L1L1 ERCCR
13 .L1L1 ERCCR
14 .L1L1 ERCCR
15 .L1L1 ERCCR
16 .L1L1 ERCCR
17 .L1L1 ERCCR
18 .L1L1 ERCCR
19 .L1L1 ERCCR
20 .L1L1 ERCCR
21 .L1L1 ERCCR
22 .L1L1 ERCCR
23 .L1L1 ERCCR
24 .L1L1 ERCCR
25 .L1L1 ERCCR
26 .L1L1 ERCCR
27 .L1L1 ERCCR
28 .L1L1 ERCCR
29 .L1L1 ERCCR
30 .L1L1 ERCCR
31 .L1L1 ERCCR
32 .L1L1 ERCCR
33 .L1L1 ERCCR
34 .L1L1 ERCCR
35 .L1L1 ERCCR
36 .L1L1 ERCCR
37 .L1L1 ERCCR
38 .L1L1 ERCCR
39 .L1L1 ERCCR
40 .L1L1 ERCCR
41 .L1L1 ERCCR
42 .L1L1 ERCCR
43 .L1L1 ERCCR

```



10059 EXMEM

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

```

EC.MP= SYSTEM IS MAPPED  
 RETRIEVE DATA THAT FAILED USING THE 32K  
 ALWAYS MAPPED FEATURE  
 EC.MP: LDA 0,EC.M6  
 LDA 1,EC.FC  
 HIGHER ADDR BITS IN AC1  
 FACI=PHYS PG #  
 PREPO 10 ADDR BITS  
 LSH 0,1  
 NEG 0,0  
 LSH 0,2  
 LDA 3,EC.K76K  
 ADD 3,2  
 DOB 1,MAP  
 LDA 0,0,2  
 JMP EC.XD  
 ECKM6: 6  
 03660 076000 ECK76K: 76000  
 03661 03662 ERCTX: +1  
 03662 05215 .TXTE 1<15><12>ERC CODE=1  
 120303  
 142504  
 006275  
 03670 03671 ECTTX: +1  
 151305 .TXTE IERRS=1  
 051722  
 000275  
 040712 EC.XD: STA 0,EC.FD  
 SUB 0,0  
 STA 0,TIMSW  
 ISZ ECTOT  
 MOV 0,0  
 LDA 1,SWREG  
 MOVZR 1,1,SNC  
 JMP .,4  
 03703 000404 LDA 1,ECBELL  
 SKPBN TIO  
 DOAS 1,TIO  
 ADI 3,0  
 DOAS 0,ERCC  
 POP 3,1  
 DIA 0,MAP  
 DOA 0,MAP  
 POP 0,0  
 INTEN  
 JMP 30  
 03717 000207 ECBELL: 207

10060 EXMEM

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

```

.TITL CBROS  
 MEMORY CHECKERBOARD VI  
 PATTERN GENERATION AND CHECKING IS NOT  
 MOVED INTO THE SELECTED SCRATCH AREAS  
 FOR EXECUTION  
 :  
 :DEF'S TO LINKER PARAMETER FILE FOLLOWS  
 NEXT CB.00  
 .LOC LPS0  
 .LOC LMEML  
 0  
 :INTERRUPT WAIT TIME  
 CB.1 :INIT ENTRY ADRS.  
 CB.02 :EXECUTE ENTRY ADRS  
 0 :NO I/0 WAITS  
 0 :RAND SEL LIMITS  
 -1 :ALWAYS ENTER  
 77 :EVERY PROTECT BIT ON  
 CB.EC :NO I/O VALIDITY TRAPS  
 CB.EC :NO WRITE OR DEFER TRAPS  
 .TXTE (  
 040303 CNKRBRD VI(  
 15113  
 151102  
 120104  
 144526  
 000000  
 000000  
 03737 003744 CB.2  
 LDA 0,-1  
 STA 0,CB.00+1  
 JMP 0,3  
 03742 001400  
 03743 004153 CB.02  
 03744 03777 CB.2:  
 LDA 3,-1  
 STA 3,CB.00+1  
 JMP CB.01  
 03746 000353

10061 EXMEM  
01  
02  
03

04 :CONSTANTS AND TABLES FOR EXERCISING INTERLEAVED MEM  
05 :RUN INTERLEAVED  
06 :DIXST+17 HIGHEST ADRS SHUFFLED  
07 :ADDED TO DIXST FOR SHUFFLE  
08 :TABLE FOR INSTR. MOD  
09 :INST\*1814  
10 :BUMP\*1814  
11 :F. FOR ADRS SEQUENCE  
12 :INTERLV. FACTOR FOR PASS

13 :2WAY INTERLEAVED  
14 :INSTL2: 56 :DIXST+56=HIGH ADRS IN SHUFFLE  
15 :ADDED FOR SHUFFLE OFFSET  
16 :INST\*1814  
17 :INST. MODIF. TABLE  
18 :BUMP\*1814  
19 :ADRS. SEQUENCE  
20 :ILV. FACTOR FOR PASS

21 :4 WAY INTERLEAVED  
22 :INSTL4: 154 :DIXST+154=HIGH ADRS SHUFFLED  
23 :ADDED FOR SHUFFL OFFSET  
24 :INST\*1814  
25 :INST. MODIF. BYTES  
26 :BUMP\*1814  
27 :ADRS. SEQ. BYTES  
28 :ILV FACTOR = PASS

29 :8 WAY INTERLEAVED  
30 :INSTL8: 380+0  
31 :DIXST+380 HIGHEST SHUFFLED  
32 :ADDED TO DIXST FOR SHUFFLE  
33 :INST. MOD. BYTE TABLE  
34 :ADRS. SEQ. BYTES  
35 :ILV FACTOR = PASS  
36 :END OF TABLES

37 :16 WAY INTERLEAVED  
38 :INSTL16: 760+0  
39 :DIXST+760 HIGHEST SHUFFLED  
40 :ADDED TO DIXST FOR SHUFFLE  
41 :INST. MOD. BYTE TABLE  
42 :ADRS. SEQ. BYTES  
43 :ILV FACTOR = PASS  
44 :END OF TABLES

45 :32 WAY INTERLEAVED  
46 :INSTL32: 1520+0  
47 :DIXST+1520 HIGHEST SHUFFLED  
48 :ADDED TO DIXST FOR SHUFFLE  
49 :INST. MOD. BYTE TABLE  
50 :ADRS. SEQ. BYTES  
51 :ILV FACTOR = PASS  
52 :END OF TABLES

53 :64 WAY INTERLEAVED  
54 :INSTL64: 3040+0  
55 :DIXST+3040 HIGHEST SHUFFLED  
56 :ADDED TO DIXST FOR SHUFFLE  
57 :INST. MOD. BYTE TABLE  
58 :ADRS. SEQ. BYTES  
59 :ILV FACTOR = PASS  
60 :END OF TABLES

10062 EXMEM  
01  
02  
03

04 :INSTRUCTION MODIFICATION TABLES  
05 :RUN INTERLEAVED  
06 :DIXST+17 HIGHEST ADRS SHUFFLED  
07 :ADDED TO DIXST FOR SHUFFLE  
08 :TABLE FOR INSTR. MOD  
09 :INST\*1814  
10 :BUMP\*1814  
11 :F. FOR ADRS SEQUENCE  
12 :INTERLV. FACTOR FOR PASS

13 :2WAY INTERLEAVED  
14 :INSTL2: 56 :DIXST+56=HIGH ADRS IN SHUFFLE  
15 :ADDED FOR SHUFFLE OFFSET  
16 :INST\*1814  
17 :INST. MODIF. TABLE  
18 :BUMP\*1814  
19 :ADRS. SEQUENCE  
20 :ILV. FACTOR FOR PASS

21 :4 WAY INTERLEAVED  
22 :INSTL4: 154 :DIXST+154=HIGH ADRS SHUFFLED  
23 :ADDED FOR SHUFFL OFFSET  
24 :INST\*1814  
25 :INST. MODIF. BYTES  
26 :BUMP\*1814  
27 :ADRS. SEQ. BYTES  
28 :ILV FACTOR = PASS

29 :8 WAY INTERLEAVED  
30 :INSTL8: 380+0  
31 :DIXST+380 HIGHEST SHUFFLED  
32 :ADDED TO DIXST FOR SHUFFLE  
33 :INST. MOD. BYTE TABLE  
34 :ADRS. SEQ. BYTES  
35 :ILV FACTOR = PASS  
36 :END OF TABLES

37 :16 WAY INTERLEAVED  
38 :INSTL16: 760+0  
39 :DIXST+760 HIGHEST SHUFFLED  
40 :ADDED TO DIXST FOR SHUFFLE  
41 :INST. MOD. BYTE TABLE  
42 :ADRS. SEQ. BYTES  
43 :ILV FACTOR = PASS  
44 :END OF TABLES

45 :32 WAY INTERLEAVED  
46 :INSTL32: 1520+0  
47 :DIXST+1520 HIGHEST SHUFFLED  
48 :ADDED TO DIXST FOR SHUFFLE  
49 :INST. MOD. BYTE TABLE  
50 :ADRS. SEQ. BYTES  
51 :ILV FACTOR = PASS  
52 :END OF TABLES

53 :64 WAY INTERLEAVED  
54 :INSTL64: 3040+0  
55 :DIXST+3040 HIGHEST SHUFFLED  
56 :ADDED TO DIXST FOR SHUFFLE  
57 :INST. MOD. BYTE TABLE  
58 :ADRS. SEQ. BYTES  
59 :ILV FACTOR = PASS  
60 :END OF TABLES

10063 EXMEM

```

01
02
03
04 04034 000401
05 04035 000401
06 04036 000401
07 04037 000401
08 04040 000401
09 04041 000401
10 04042 000401
11 04043 000400
12
13 04044 001002
14 04045 001002
15 04046 001002
16 04047 001002
17 04050 001002
18 04051 001002
19 04052 001002
20 04053 001000
21
22 04054 002004
23 04055 002024
24 04056 002004
25 04057 002024
26 04060 002004
27 04061 002024
28 04062 002004
29 04063 002000
30
31 04064 004030
32 04065 004030
33 04066 004030
34 04067 004030
35 04070 004030
36 04071 004030
37 04072 004030
38 04073 004000

```

10064 EXMEM

```

01
02
03
04 04074 011675
05 04075 054125
06 04076 034776
07 04077 142710
08 04100 163010
09 04101 040130
10 04102 024146
11 04103 151400
12 04104 134010
13 04105 142710
14 04106 165010
15 04107 134010
16 04110 163010
17 04111 125404
18 04112 000771
19 04113 134010
20 04114 020130
21 04115 163010
22 04116 102400
23 04117 040165
24 04120 002125
25
26 04121 102400
27 04122 040112
28 04123 020154
29 04124 040155
30 04125 054020
31 04126 030155
32 04127 034157
33 04130 024156
34 04131 137110
35 04132 050155
36 04133 030162
37 04134 004741
38 04135 034020
39 04136 030243
40 04137 020064
41 04140 101004
42 04141 130010
43 04142 024000
44 04143 025001
45 04144 040115
46 04145 044116
47 04146 040120
48 04147 120510
49 04150 040121
50 04151 050114
51 04152 001400

```

```

;ADDRESS SEQUENCE BYTE TABLES USED TO INCR DIXNX
;UNINTERLEAVED
BUMP0: 187+1
187+1
187+1
187+1
187+1
187+1
187+1
187+0
;2 WAY INTL
RUMP2: 287+2
287+2
287+2
287+2
287+2
287+2
287+2
287+0
;4 WAY INTLEAVED
BUMP4: 487+4
487+4
487+4
487+4
487+4
487+4
487+4
487+0
;8 WAY INTERLEAVED
BUMP8: 1087+30
1087+30
1087+30
1087+30
1087+30
1087+30
1087+30
1087+0

```

```

;MODIFY THE SHUFFLE LOOP FOR THE CURRENT INTERLEAVING
DISL8+1814+1
18 BYTE PTR TO INSTR.
STA 3,DIRET
LDA 3,INMOD-1
LDB 2,0
;1ST MODIFY
;SAVE FOR LAST
;NXT BYTE IN TABLE
;NEXT INST.
;MOD'S 1ST OF PAIR
;THEN 2ND OF PAIR
;DO NEXT PAIR
;STEP TO LAST
;HAS SAME AS 1ST
;RESET INLEAVE OFFSET
JMP SOISET
;INITIALIZE CHECKERBOARD TEST SEQUENCE
CB.01: SUB 0,0
STA 0,CB.TK
LDA 0,INTLX
STA 0,INTLX
STA 3,20
LDA 2,INTLX
LDA 3,INTP0
LDA 1,CBITX
BLW
STA 2,INTLX
LDA 2,INTBL
JSR INMOD
LDA 3,20
LDA 2,CBTBL
LDA 0,ECSEL
MOV 0,0,SZR
ADI 2,2
LDA 0,0,2
LDA 1,1,2
STA 0,PAITA
STA 1,PAITB
STA 0,INHPA
XDR 1,0
STA 0,XORIP
STA 2,CBPTX
JMP 0,3 ;RETURN TO LINKER TEST INIT

```

```

;MOVE INTERLEAVING SET
;POINTS TO NEXT
;INSTR. MOD. BYTE POINTER
;SET UP SHUFFLE LOOP
;SKP =NOT ERCC
;DONT 16 BIT PAT
;GET INHIBIT PAT
;GET 2ND WRD
;CONSTANT TO INHIBIT
;SAVE POINTER
;RETURN TO LINKER TEST INIT

```

```

10065 EXMEM
01
02
03
04
05 06 04153 177510
07 04154 062206
08 04155 006470
09
10 04156 020107
11 04157 100110
12 04160 040166
13 04161 101220
14 04162 101220
15 04163 101220
16 04164 024282
17 04165 152520
18 04166 122210
19 04167 000404
20 04170 124010
21 04171 151400
22 04172 122210
23 04173 000404
24 04174 124010
25 04175 150010
26 04176 122210
27 04177 000404
28 04200 124010
29 04201 170010
30 04202 122210
31 04203 151001
32 04204 152400
33 04205 050167
34 04206 102000
35 04207 040113
36
37 04210 177510
38 04211 006207
39 04212 000402
40 04213 000775

;EXECUTE ENTRY POINT
CB.02:  LCALL ASCRA
;TRY TO GET IK
SYC 3,3
JSR 8ESCRA
;DETERMINE WHICH INTER LEAVING TABLE
LDA 0,CNDUAL
SBI 1,0
STA 0,MODU
MOVZR 0,0
MOVZR 0,0
MOVZR 0,0
LDA 1,CBTRL-1
SUBZL 2,2
SXB 1,0
JMP *+4
ADI 2,1
INC 2,2
SXB 1,0
JMP *+4
ADI 2,1
MOV 2,2,SXP
SUB 2,2
STA 2,MODI
ADC 0,0
STA 0,CB.ES
CB.2L:  LCALL ESCRA
SYC 3,3
JSR 8ESCRA
JMP CB.03
JMP CB.2L

;EXECUTE NEXT TEST IN SEQUENCE
CB.03:  SUB 0,0
STA 0,FSHUF
LDA 2,CB.TK
LDA 1,CB.TS
ADD 1,2
LDA 3,0,2
LDA 1,CB.BG
ADD 1,3
STA 2,CB.TI
STA 3,CB.SE
JSR 0,3
JMP CB.X1
;***GO TO TEST ***
;NO ERROR RETURN
;SKIP ON RETURN IS GROSS ERROR
;FAST CHECKSUM OF MEMORY WAS NOT CORRECT
;BUT THE SECOND PASS THROUGH THE DATA CHECK
;DID NOT FIND ANY ERRORS IN PATTERN GENERATED
;FAST SUM RESULT
LDA 0,OTDIL
LDA 2,CB.EN
LDA 1,-4,2
JSR CB.ER
CB.X1:  LCALL RSCRA
SYC 3,3
JSR 8RSCRA
MOV 0,0,SXP
JMP CB.X1
LDA 0,TIMSW
MOV 0,0,SZR
JMP CB.02
LCALL RETRN
SYC 3,3
JSR 8RETRN

10066 EXMEM
01
02
03
04 04214 102400
05 04215 040106
06 04216 030112
07 04217 024223
08 04220 133000
09 04221 035000
10 04222 024123
11 04223 137000
12 04224 050135
13 04225 054137
14 04226 005400
15 04227 000405
16
17
18
19 04230 020136
20 04231 030133
21 04232 025374
22 04233 044505
23
24 04234 177510
25 04235 062210
26 04236 101001
27 04237 000775
28 04240 020073
29 04241 101004
30 04242 000711
31
32 04243 177510
33 04244 062213

;EXECUTE NEXT TEST IN SEQUENCE
CB.03:  SUB 0,0
STA 0,FSHUF
LDA 2,CB.TK
LDA 1,CB.TS
ADD 1,2
LDA 3,0,2
LDA 1,CB.BG
ADD 1,3
STA 2,CB.TI
STA 3,CB.SE
JSR 0,3
JMP CB.X1
;***GO TO TEST ***
;NO ERROR RETURN
;SKIP ON RETURN IS GROSS ERROR
;FAST CHECKSUM OF MEMORY WAS NOT CORRECT
;BUT THE SECOND PASS THROUGH THE DATA CHECK
;DID NOT FIND ANY ERRORS IN PATTERN GENERATED
;FAST SUM RESULT
LDA 0,OTDIL
LDA 2,CB.EN
LDA 1,-4,2
JSR CB.ER
CB.X1:  LCALL RSCRA
SYC 3,3
JSR 8RSCRA
MOV 0,0,SXP
JMP CB.X1
LDA 0,TIMSW
MOV 0,0,SZR
JMP CB.02
LCALL RETRN
SYC 3,3
JSR 8RETRN

;TRY TO GET IK
;LAST PASS WAS END OF CORE
;START THIS 8K
PADRS OF INTL TBL
;SKP=NOT UNINTL.
;CORE UNINTL.
;SKP IF NOT 2 WAY
;CORE 2 WAY
;SKP=NOT 4 WAY
;CORE 4WAY
;SKP=NOT CORE
;CORE 8 WAY
;PROPLY. SC MEM
;SAV INTL. FACTOR
;=1 TO
;NO ERRSW
;EXPAND SCRATCH IK
;HAVE REACHED END OF MODUAL
;KEEP EXPANDING

```

```

10067 EXMEM
01
02
03 04245 010112
04 04246 010135
05 04247 022135
06 04250 100004
07 04251 000763
08 04252 102400
09 04253 040112
10 04254 030120
11 04255 020115
12 04256 024116
13 04257 040116
14 04260 044115
15 04261 146414
16 04262 000752
17 04263 010165
18 04264 020165
19 04265 024164
20 04266 122404
21 04267 000745
22 04270 040165
23 04271 020064
24 04272 101005
25 04273 000411
26 04274 030114
27 04275 130010
28 04276 021000
29 04277 101005
30 04300 000405
31 04301 004642
32 04302 000732
33 04303 004633
34 04304 022155
35 04305 101005
36 04306 000403
37 04307 006420
38 04310 000724
39 04311 034257
40 04312 175004
41 04313 000416
42 04314 020523
43
44 04315 177510
45 04316 006212
46 04317 006411
47
48 04320 024117
49
50 04321 177510
51 04322 006211
52 04323 010117
53 04324 102400
54 04325 040073
55 04326 000706
56 04327 004125
57 04330 004121

;THE END OF ONE PASS THROUGH ALL AVAILABLE CORE
;SEE IF ALL OPTIONS HAVE BEEN EXERCISED
CB.04:
ISZ CB.TK
ISZ CB.TI
LDA 0,CB.TI
JMP 0,CB.TI
JMP 0,CB.TI
JMP 0,CB.TI
JMP 0,CB.TI
JMP 0,CB.TK
LDA 2,INPRA
LDA 0,PATIA
LDA 1,PATIB
STA 0,PATIB
STA 1,PATIA
SUB# 2,1-SZR
JMP CB.X1
ISZ INTLF
LDA 0,INTLF
LDA 1,INTLS
SUB 1,0,SZR
JMP CB.X1
STA 0,INTLF
LDA 0,ECSEL
MOV 0,0,SNR
JMP *+11
LDA 2,CBPTX
ADI 2,-2
LDA 0,0,2
MOV 0,0,SNR
JMP *+5
JMP CB.1A
JMP CB.X1
JSR CB.1A-5
LDA 0,INTLX
MOV 0,0,SNR
JMP *+3
JSR 0,CB.4E-2
JMP CB.X1
LDA 3,EGGS
MOV 3,3,SZR
JMP CB.4E
LDA 0,CB.PX
LCALL ERRTX
SYC 3,3
JSR 0ERRTX
JSR 0,CB.4E-1
;REINIT THE WORLD
LDA 1,WPASK
LCALL POECL
SYC 3,3
JSR 0PECCI
ISZ MPASK
SUB 0,0
STA 0,TIMSW
JMP CB.X1
CB.01

;SKP=ERR COR
;PATRN INDEX
;NXT INH. WORD
;SKP=NOT AT END
;SET UP NEXT PAT.
;RESTART PAT'S
;SKP IF NOT ALL INTL. TSTD
;SET UP NEXT INTLV.
;PASS TEXT
;PRINT IT
;FORCE TIME TYPE

```

```

10068 EXMEM
01 04331 014262 CB.4E: DSZ EGGS+3
02 04332 000702 JMP CB.X1
03 04333 062677 TORST
04 04334 034263 LDA 3,EGGS+4
05 04335 102400 SUB 0,0
06 04336 041776 STA 0,-2,3
07 04337 001400 JMP 0,3
08 ;PATTERN CHECK FOUND AN ERROR
09 04340 117110 CB.ER: PSH 0,3
10 04341 010113 ISZ CB.ES ;SKP=FIRST ERR
11 04342 000431 JMP CB.E2 ;NOT FIRST ERR
12 04343 163210 POP 3,0
13 LCALL ERROI
14 04344 177510 SYC 3,3
15 04345 006215 JSR 0ERRI
16 04346 117110 CB.EC: PSH 0,3
17 04347 020443 LDA 0,CBXTX
18 LCALL ERRTX
19 04350 177510 SYC 3,3
20 04351 006212 JSR 0ERRTX
21 04352 020112 LDA 0,CB.TK
22 04353 024164 LDA 1,INTLS
23 04354 030165 LDA 2,INTLF
24 LCALL ERROC
25 04355 177510 SYC 3,3
26 04356 006216 JSR 0ERRC
27 04357 020257 CH.EX: LDA 0,EGGS
28 04360 101004 MOV 0,0,SZR
29 04361 000426 JMP CB.EE
30 04362 020264 LDA 0,SWREG
31 04363 101223 MOVZR 0,0,SNC
32 04364 000404 JMP *+4
33 04365 020424 LDA 0,CBELL
34 04366 063411 SKP8N TTD
35 04367 061111 DOAS 0,TTO
36 04370 163210 POP 3,0
37 04371 041000 STA 0,0,2
38 04372 001400 JMP 0,3
39 04373 020113 CB.E2: LDA 0,CB.ES
40 04374 101224 MOVZR 0,0,SZR
41 04375 000405 JMP CB.E3
42 04376 020427 LDA 0,CBXTX
43 LCALL ERRTX
44 04377 177510 SYC 3,3
45 04400 006212 JSR 0ERRTX
46 04401 000403 JMP CB.E4
47 04402 101224 CB.E3: MOVZR 0,0,SZR
48 04403 000754 JMP CB.E5
49 04404 163210 CB.E4: POP 3,0
50 04405 117110 PSH 0,3
51 04406 000747 JMP CB.E5
52 04407 062677 CB.EE: TORST
53 04410 002263 JMP 0EGGS+4
54 04411 000207 CBELL: 207

```

```

;INTERLEAVE TEST #
;CURRENT INTER LEAVE FACTOR
;RESTORE COR. DATA
;RETURN TO COMPARE
;SKP=2ND ERR
;SKP=LESS THAN 3 ERRS
;ONLY TYPE 3

```



10069 EXMEM

```

01 04412 004413 CBXT: *.1
02 04413 005215 CBXT: *.1TXE (<15><12>CB.TK INTLS INTLP(
03 04425 004426 CBXT: *.1TXE (<15><12>6000 BAD ADRS.:!
04 04426 005215 CB.PX: *.1TXE (<15><12>PASS 1
05 04437 004440 .NDLOC 0
06 04440 005215
07 04440 000000
08
09
10
11
12 04444 054140
13 04445 004502
14 04446 030171 JSR IFILL
15 04447 024141 LDA 1,C400
16 04450 020115 LDA 0,PATTA
17 04451 147405 AND 2,1,SNR
18 04452 000405 JMP FILL-1
19 04453 024115 LDA 1,PATTA
20 04454 122415 SUB# 1,0,SNR
21 04455 024116 LDA 1,PATTB
22 04456 121000 MOV 1,0
23 04457 024142 LDA 1,C17
24 04460 034120 LDA 3,INHPPA
25 04461 116415 SUB# 0,3,SNR
26 04462 133001 ADD 1,2,SKP
27 04463 041000 STA 0,0,2
28 04464 151400 INC 2,2
29 04465 133414 AND# 1,2,SZR
30 04466 000775 JMP FILL+3
31 04467 034144 LDA 3,C77
32 04470 157414 AND# 2,3,SZR
33 04471 000762 JMP IPAT1
34 04472 024172 LDA 1,SCRHI
35 04473 125400 INC 1,1
36 04474 146434 SUB# 2,1,SZR
37 04475 000752 JMP IPAT+1
38 04476 002140 JMP @RETURN
39 04477 010112 BEG2:
40 04500 020115
41 04501 024116
42 04502 040116
43 04503 044115
44 04504 000740

```

10070 EXMEM

```

01
02 04505 054140
03 04506 030171
04 04507 024141 LDA 1,C400
05 04510 020115 LDA 0,PATTA
06 04511 133415 AND# 1,2,SNR
07 04512 000405 JMP CHECK
08 04513 024115 LDA 1,PATTA
09 04514 122415 SUB# 1,0,SNR
10 04515 024116 LDA 1,PATTB
11 04516 121000 MOV 1,0
12 04517 025000 CHECK:
13 04520 106414 SUB# 0,1,SZR
14 04521 006151 JSR @ERR1
15 04522 034121 LDA 3,XORIP
16 04523 164510 XOR 3,1
17 04524 045000 STA 1,0,2
18 04525 025000 LDA 1,0,2
19 04526 164510 XOR 3,1
20 04527 045000 STA 1,0,2
21 04530 025000 LDA 1,0,2
22 04531 122414 SUB# 1,0,SZR
23 04532 006151 JSR @ERR1
24 04533 151400 INC 2,2
25 04534 024142 LDA 1,C17
26 04535 133414 AND# 1,2,SZR
27 04536 000761 JMP CHECK
28 04537 034144 LDA 3,C77
29 04540 157414 AND# 2,3,SZR
30 04541 000752 JMP ICK
31 04542 024172 LDA 1,SCRHI
32 04543 125400 INC 1,1
33 04544 146434 SUB# 2,1,SZR
34 04545 000742 JMP ICKCHK+2
35 04546 002140 JMP @RETURN
36
37
38
39 04547 054125
40 04550 030171
41 04551 158400
42 04552 024172 LDA 1,SCRHI
43 04553 166000 ADC 3,1
44 04554 134010 ADI 2,3
45 04555 020120 LDA 0,INHPPA
46 04556 041000 STA 0,0,2
47 04557 041001 STA 0,1,2
48 04560 041002 STA 0,2,2
49 04561 133710 BLM
50 04562 002125 JMP @DIRET

```

!CHECK PATTERN IN SCRATCH AREA AGAINST GENERATED

!CHECK: STA 3,RETURN ;INIT PAT

LDA 2,SCRLO ;

LDA 1,C400 ;

LDA 0,PATTA ;

AND# 1,2,SNR ;

JMP CHECK ;

LDA 1,PATTA ;

SUB# 1,0,SNR ;

LDA 1,PATTB ;

MOV 1,0 ;

LDA 1,0,2 ;

SUB# 0,1,SZR ;

JSR @ERR1 ;

LDA 3,XORIP ;

XOR 3,1 ;

STA 1,0,2 ;

LDA 1,0,2 ;

XOR 3,1 ;

STA 1,0,2 ;

LDA 1,0,2 ;

SUB# 1,0,SZR ;

JSR @ERR1 ;

INC 2,2 ;

LDA 1,C17 ;

AND# 1,2,SZR ;

JMP CHECK ;

LDA 3,C77 ;

AND# 2,3,SZR ;

JMP ICK ;

LDA 1,SCRHI ;

INC 1,1 ;

SUB# 2,1,SZR ;

JMP ICKCHK+2 ;

JMP @RETURN ;

!CHK END OF LINE

!CHK END OF CORE

!FILL SCRATCH AREA WITH INHIBIT PATTERN

!THE OFFSET OF 3 RUNS FASTEST IN ANY INTERLEAVING

!FILL: STA 3,DIRET

LDA 2,SCRLO

INC 2,3

LDA 1,SCRHI

ADC 3,1

ADI 2,3

LDA 0,INHPPA

STA 0,0,2

STA 0,1,2

STA 0,2,2

BLM

JMP @DIRET

```
10071 EXMEM
01
02
03
04 04563 020117
05 04564 040106
06 04565 054140 DISTUR: STA 3,RETURN
07 04566 020117 LDA 0,MPASK
08 04567 101004 MOV 0,0,SZR
09 04570 000405 JMP *5
10 04571 020164 LDA 0,INTLS
11 04572 024167 LDA 1,MODI
12 04573 106414 SUB# 0,1,SZR
13 04574 002140 JMP @RETURN
14 04575 030171 LDA 2,SCRELO
15 04576 020165 LDA 0,INTLF
16 04577 113000 ADD 0,2
17 04600 145000 DISTL: MOV 2,1
18 04601 020160 LDA 0,OPADK
19 04602 107000 ADD 0,1
20 04603 034172 LDA 3,SCRHI
21 04604 164436 SUB# 3,1,SEZ
22 04605 002140 JMP @RETURN
23
24
25 04606 020141
26 04607 144510 XOR 2,1
27 04610 107405 AND 0,1,SNR
28 04611 000404 JMP DISDO
29 04612 020143 LDA 0,C20
30 04613 113000 ADD 0,2
31 04614 000764 JMP DISTL
32 04615 050131 DIS00: STA 2,DIYST
33 04616 050130 STA 2,DIYNX
34 04617 102320 SUBZL 0,0
35 04620 040132 STA 0,DIRBT
36 04621 020163 LDA 0,OPADI
37 04622 040126 STA 0,DIYCT
38 04623 021000 LDA 0,0,2
39 04624 024120 LDA 1,INHPA
40 04625 122414 SUB# 1,0,SZR
41 04626 000406 JMP DISCM
42 04627 004420 JSR DISEX
43 04630 030151 LDA 2,DIYST
44 04631 024143 LDA 1,C20
45 04632 133000 ADD 1,2
46 04633 000745 JMP DISTL
47 04634 004455 DISCM: JSR DIXCM
48 04635 004412 JSR DISEX
49 04636 004453 JSR DIXCM
50 04637 000771 JMP DISCM+4

; SHUFFLE MEM BY FLOATING A BIT OR NO BIT THROUGH 16 WORD
LDA 0,MPASK
STA 3,RETURN
LDA 0,MPASK
MOV 0,0,SZR
JMP *5
LDA 0,INTLS
LDA 1,MODI
SUB# 0,1,SZR
JMP @RETURN
LDA 2,SCRELO
LDA 0,INTLF
ADD 0,2
;2=START OF 16
;MAKE I=END+1
LDA 0,OPADK
ADD 0,1
LDA 3,SCRHI
SUB# 3,1,SEZ
JMP @RETURN
;YES EXIT JOB DONE
;MAKE SURE THE WHOLE INTERLEAVE SHUFFLE
;IS WITHIN THE SAME 256 WORDS
LDA 0,C400
XOR 2,1
AND 0,1,SNR
JMP DISDO
LDA 0,C20
ADD 0,2
JMP DISTL
STA 2,DIYST
STA 2,DIYNX
SUBZL 0,0
STA 0,DIRBT
LDA 0,OPADI
STA 0,DIYCT
LDA 0,0,2
LDA 1,INHPA
SUB# 1,0,SZR
JMP DISCM
JSR DISEX
LDA 2,DIYST
LDA 1,C20
ADD 1,2
JMP DISTL
JSR DIXCM
JSR DISEX
JMP DISCM+4
;SKIP IF B7'S DIFFR
;SAVE STRT 16 WORDS
;NEXT WORD TO DO
;NEXT BIT TO DO
;GET FIRST WORD
;INHIBIT PAT
;SKP=ALRDY INHIBIT
;COMPLIMENT PAT FIRST
;SHUFFLE
;DO NEXT 16 WORDS
;COMPLIMENT NEXT 16
;SHUFFLE THEM
```

10072 EXMEM

```
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
;THE NEXT TWO ROUTINES ARE NOT NORMALLY EXECUTED
;BUT ARE INCLUDE TO HELP BEAT UP MEMORY'S
;THAT ARE SUSPECT BUT POSSIBLY NOT
;FAILING AT A RATE HIGH ENOUGH TO ISOLSTE
;THE PROBLEM OR PROBLEMS OCCURRING
;CHANGE THE DISTURB ENTRIES IN THE DISPATCH
;TABLE TO = EITHER OF THE FOLLOWING CONSTANTS
;THE TEST DISPATCH TABLE STARTS WITH THE LABEL
;CB.TS THE DISTURB ENTRIES ARE DISTUR-BEGIN
;A GOOD SEQUENCE TO MORE THOROUGHLY
;EXERCISE MEMORIES WOULD BE TO CHANGE
;BOTH SETS OF THREE DISTURBS TO =
;
;
000174 OIXXX=-BEGIN
000176 OIYYY=*2-BEGIN
18 04640 020405 DISLO: LDA 0,DISEX-2
19 04641 101001 MOV 0,0,SKP
20 04642 020404 DISFA: LDA 0,DISEX-1
21 04643 040431 STA 0,DISXL
22 04644 000717 JMP DISTUR-2
;JMP'S TO MODIFY THE DISTURB LOOP
;
;
04645 000763 JMP *-13.
04646 000401 JMP *+1
;SLGW SHUFFLE
;FAST SHUFFLE
```

```

10073 EXMEM
01 04647 054127 OISEX: STA 3,DISRT
02 04650 034164 LDA 3,INTLS
03 04651 030167 LDA 2,-MODI
04 04652 172414 SUB# 1,2,SZR
05 04653 006453 JMP DISLA
06 04654 030106 LDA 2,IFSHUF
07 04655 151005 MOV 2,2,SNR
08 04656 006450 LDA 2,DISRT
09 04657 030130 DISXX: JMD 2,DIXNX
10 04660 004521 JSR DIXOR
11 04661 030131 LDA 2,DIXST
12 04662 020143 LDA 0,C20
13 04663 024161 LDA 1,AC2MD
14 04664 004447 JSR DISDL
15 04665 030130 LDA 2,DIXNX
16 04666 004515 JSR DIXOR
17 04667 020132 LDA 0,DIXBT
18 04670 101125 MOVZL 0,0,SNR
19 04671 101400 INC 0,0
20 04672 040132 STA 0,DIXBT
21 04673 101003 MOV 0,0,SNC
22 04674 000401 DISXL: JMP +1,DR JMP
23 04675 020120 LDA 0,INHPA
24 04676 122414 SUB# 1,0,SZR
25 04677 006151 JSR SERRI
26 04700 034126 LDA 3,DIXCT
27 04701 101026 ISZ DIXCT
28 04702 030130 LDA 2,DIXNX
29 04703 162710 LDB 3,0
30 04704 113000 ADD 0,2
31 04705 050130 STA 2,DIXNX
32 04706 101004 MOV 0,0,SZR
33 04707 000750 JMP DISXX
34 04710 002127 JMP 0,DISRT
35 04711 030131 ;COMPLIMENT NEXT 16 WORDS OF MEMORY TO
36 04712 054125 DIXCM: LDA 2,DIXST
37 04713 034163 STA 3,DIRET
38 04714 024121 LDA 3,OPADI
39 04715 021000 LDA 1,XORIP
40 04716 120510 LDA 0,0,2
41 04717 041000 XOR 1,0
42 04718 162710 STA 0,0,2
43 04720 162710 LDB 3,0
44 04721 113000 ADD 0,2
45 04722 175400 INC 5,3
46 04723 101004 MOV 0,0,SZR
47 04724 000771 JMP DIXCM+4
48 04725 002125 JMP 0,IRET
49 04726 020143 DISXL: LDA 0,C20
50 04727 024161 LDA 1,AC2MD
51 04730 030131 LDA 2,DIXST
52 04731 004402 JSK DISDL
53 04732 002127 JMP 0,DISRT

10074 EXMEM
01 ; THIS LOOP SHUFFLES SINGLE BIT MODIFIED WORDS
02 ; THROUGH A GROUP OF 16 WORDS
03 ; THE LOWER BYTE OFFSETS ARE MODIFIED TO
04 ; REFLECT THE INTERLEAVING PAAS
05 DISOL: STA 3,DIRET
06 04733 054125 NEG 0,3
07 04734 114400 ADD 1,2
08 04735 133000 ; 18 WAY AC1=S 200 (0 ALL OTHERS)
09 04736 021000 ;WORD 0
10 04737 025001 ;GOES TO WORD 1
11 04740 041001 LDA 0,0,2
12 04741 021002 STA 0,1,2
13 04742 025002 LDA 0,2,2
14 04743 025003 STA 1,2,2
15 04744 041003 LDA 1,3,2
16 04745 021004 STA 0,3,2
17 04746 045004 LDA 1,4,2
18 04747 025005 STA 1,5,2
19 04750 041005 STA 0,5,2
20 04751 021006 LDA 0,6,2
21 04752 025006 STA 1,6,2
22 04753 025007 LDA 1,7,2
23 04754 041007 STA 0,7,2
24 04755 021010 LDA 0,10,2
25 04756 045010 STA 1,10,2
26 04757 025011 LDA 1,11,2
27 04760 041011 STA 0,11,2
28 04761 021012 LDA 0,12,2
29 04762 045012 STA 1,12,2
30 04763 025013 LDA 1,13,2
31 04764 041013 STA 0,13,2
32 04765 021014 LDA 0,14,2
33 04766 045014 STA 1,14,2
34 04767 025015 LDA 1,15,2
35 04770 041015 STA 0,15,2
36 04771 021016 LDA 0,16,2
37 04772 045016 STA 1,16,2
38 04773 025017 LDA 1,17,2
39 04774 041017 STA 0,17,2
40 04775 045000 STA 1,0,2
41 04776 175404 INC 3,3,SZR
42 04777 000757 JMP DISLP
43 05000 002125 JMP 0,DIRET
44 ;COMS A SINGLE BIT IN ONE OF NEXT 16 WORDS
45 05001 020132 DIXOR: LDA 0,DIXBT
46 05002 025000 LDA 1,0,2
47 05003 104510 XOR 0,1
48 05004 045000 STA 1,0,2
49 05005 001400 JMP 0,3

;SKP=END OF 16
;ONLY SHUFFLE ONCE

```

!0075 EXMEM

```

02 *REAL TIME CLOCK TEST TO RUN WITH LINKER
03 *LOADED 10/01/72 MPT16,SR
04 *IF A REAL TIME CLOCK EXISTS
05 *RUNTIME WILL BE TYPED AT 5 MINUTES
06 *15 MINUTES AND ON EACH HALF HOUR.
07 *ALSO, FOLLOWING ANY ERROR TYPEDOUT
08 *FOR ANY TYPE IN WITH ACS#=1
09 * (SEE TTY TEST TO CLR TIMSW)
10 *NEXT RT.00
11 005006 LMEWL=.
12 000175 -LOC LPGO
13 00175 005007 RT.00
14 000176 LPGO=.
15 005006
16 05006 000000 -LOC LMEWL
17 05007 005027 RT.00 ;INTERRUPT WAIT TIME
18 05010 005057 RT.01
19 05011 000000 RT.02
20 05012 000000 0
21 05013 177777 -1
22 05014 000077 77
23 05015 005103 RT.03
24 05016 005105 RT.03
25 05017 142722 *.TATE (REAL TIME CLOCK(
26 146101
27 152240
28 046711
29 120305
30 146303
31 141717
32 000113

```

!0076 EXMEM

```

01 *DETERMINE WHETHER OR NOT A REAL TIME CLOCK EXISTS
02 *ENABLE OR DISABLE TEST ACCORDINGLY
03 RT.01: INTOS
04 05027 060277 AOC 0,0 ;DISABLES TEST
05 05030 102000 STA 0,RT.00+2 ;TURN CLOCK ON
06 05031 040760 NIOS RTC ;TURN CLOCK ON
07 05032 040114 SKPBZ RTC ;SKTP MAYBE NO CLOCK
08 05033 065514 JMP *+2 ;CLOCK EXISTS
09 05034 004402 SKPDZ RTC ;TRY FOR DOME =1
10 05035 063714 JMP *+2 ;CLOCK EXISTS
11 *NO RTC ON SYSTEM EXIT WITH CLOCK DISABLED
12 05037 001400 JMP 0,3
13 05040 102400 SUB 0,0
14 05041 040750 STA 0,RT.00+2 ;ENABLE CLOCK TEST
15 05042 040073 STA 0,TIMSW ;FORCE RTC ENTER
16 05043 020413 LDA 0,RT.02-1
17 05044 040744 STA 0,RT.00+1
18 05045 020405 LDA 0,RT.K1
19 05046 024405 LDA 1,RT.K2
20 05047 030405 LDA 2,RT.K3
21 05050 060214 NIOC RTC
22 05051 002214 JMP #REINTS ;TURN CLOCK OFF
23 05052 000014 RT.K1: RTC
24 05053 000007 RT.K2: 7
25 05054 005144 RT.K3: RT.ID
26 05055 000005 RTFIV: 5

```

```

10077 EXMEM
01 05056 00507
02 05057 020426 RT.02
03 05060 040730 RT.02:
04 05061 102420 LDA 0,RT,K4
05 05062 101500 SUBZ 0,0
06 05063 024452 INCL 0,0
07 05064 044452 LDA 1,RT,K5
08 05065 044452 STA 1,RTSEC
09 05066 024447 LDA 1,RT,K6
10 05066 044451 STA 1,RTMIN
11 05067 024766 LDA 1,RTFIV
12 05070 131120 MOVZL 1,2
13 05071 044447 STA 1,RTCTR
14 05072 050447 STA 2,RTCTR+1
15 05073 133000 ADD 1,2
16 05074 166400 SUB 1,1
17 05075 044072 STA 1,RTTIM
18 05076 126000 ADC 1,1
19 05077 044073 STA 1,TIMSW
20 05100 050442 STA 2,RTCTR+2
21 05101 102520 SUBZL 0,0
22 05102 061114 COAS 0,RTC
23 05103 177510 LCALL RETRN
24 05104 06213 SYC 3,3
25 05105 005106 JSR @RETRN
26 05106 020073 RT.K4: RT.04
27 05107 101004 LDA 0,TIMSW
28 05110 000773 JMP RT.03
29 05111 102000 ADC 0,0
30 05112 040073 STA 0,TIMSW
31 05112 040073 LDA 0,RTTEX
32 05113 020462 LCALL ERRTX
33 05114 177510 SYC 3,3
34 05115 06212 JSR @ERTX
35 05116 024072 LDA 1,RTTIM,
36 05117 177510 LCALL PDECI
37 05120 006211 SYC 3,3
38 05121 024413 JSR @PDECI
39 05122 020415 LDA 1,RT,K6
40 05123 106400 LDA 0,RTMIN
41 05124 177510 SUB 0,1
42 05125 006211 LCALL PDECI
43 05126 024074 SYC 3,3
44 05127 125005 JSR @PDECI
45 05128 000753 MOV 1,1,SNR
46 05129 177510 LCALL PDECI
47 05130 006211 SYC 3,3
48 05131 177510 JSR @PDECI
49 05132 006211 JMP RT.03
50 05133 000750

:START CLOCK TEST IS NOT DELETED
:TO COUNT 1 SECOND
:60 SEC'S =1MIN.
:10 COUNT DOWN 1ST
:2ND AFTER 10 MORE
:SET RT=0 INHIBIT TIME
:3RD AT HALF HOUR
:TURN CLOCK ON 10 MZ
:PRINT TIME
:NOT YET
:PRESET INH. SW
:ELAPSED TIME IN MINUTES
:PRINT SECONDS
:# ERR TYPEOUTS
:PRINT ERROR TOTAL

```

```

10078 EXMEM
01 05134 000074 RT.K6: 60.
02 05135 040012 RT.K5: 10.
03 05136 000000 RTSEC: 0
04 05137 000000 RTMIN: 0
05 05140 000000 RTCTR: 0
06 05141 000000 0
07 05142 000000 0
08 05143 000036 30.
09 05144 117110 RT.ID:
10 05145 060114 PSH 0,3
11 05146 014770 DSZ RTSEC
12 05147 000420 JMP RTSTR
13 05150 020765 LDA 0,RT.K5
14 05151 040765 DSZ RTMIN
15 05152 014765 JMP RTSTR
16 05153 000414 ISZ RTTIM
17 05154 010072 LDA 0,RT.K6
18 05155 020757 STA 0,RTMIN
19 05156 040761 DSZ RTCTR
20 05157 014761 JMP RTSTR
21 05161 000407 LDA 0,RTCTR+1
22 05162 024760 LDA 1,RTCTR+2
23 05163 030760 STA 2,RTCTR+3
24 05164 040754 STA 1,RTCTR+1
25 05165 044754 STA 2,RTCTR+2
26 05166 050754 POP 3,1
27 05167 167210 RTSTR:
28 05170 060403 DIA 0,MAP
29 05171 061003 DOA 0,MAP
30 05172 103210 POP 0,0
31 05172 103210 INTEN
32 05173 060177 JMP @0
33 05174 002000
34 05175 005176 RTTEX: *1
35 05176 005215 *1TXIE (<15><12>MIN. SEC. ERRS<15><12>(<
36 05176 000000 *NOLOC 0

```

10079 EXMEM

```
01  
02  
03 05210 117110 :TTY INTR SERVICE  
04 05211 006407 TT.TI: PSH 0,3  
05 05212 024407 :JSR @INP?  
06 05213 136404 LDA 1,TT-67  
07 05214 000402 SUB 1,3,SZR  
08 05215 054073 JMP TT,3A  
09 05216 060210 STA 3,IIMSW  
10 05217 000750 NIOC TTI  
11 05220 002617 IIMP?: INP?  
12 05221 000067 TT-67: 67  
13  
14 :TTY OUTPUT INTR HANDLER  
15 05222 117110 TT.TO: PSH 0,3  
16 05223 060211 NIOC TIO  
17 05224 000743 JMP RTSTR
```

10080 EXMEM

```
01  
02  
03  
04 :THE INTERLEAVE TIMING SEQUENCES HAVE BEEN CHANGED TO  
05 :HELP IN IDENTIFYING 32K SC MEMORIES AND THEIR INTERLEAVE  
06 :FACTORS. ALTHOUGH THE LOOPS WILL ALLOW IDENTIFICATION OF  
07 :8K CACHED SC MEMORIES, THE TEST NO LONGER ATTEMPTS TO  
08 :DETERMINE THEIR INTERLEAVING FACTORS.  
09 :  
10 : THE FOLLOWING LOOPS ARE TIMED  
11 : EACH LOOP CONSISTS OF A JSR,+1 13 LDA'S OR STA'S AN INC AND A JMP  
12 :  
13 : LDA 2,+*1 INTERFERES ALWAYS BUT FASTER THAN STA'S IF SC MEM  
14 : RUNS SLOWER THAN LDA +*1 IF CACHED 8K SC'S  
15 :  
16 : STA 2,+*161 ESSENTIALLY STA +*1 ALWAYS INTERFERES  
17 : ONLY INTERFERES IF UN OR 2 WAY INTERLEAVED  
18 : DOES NOT INTERFERE IF 8 WAY INTERLEAVED  
19 : SWE AS THE +*143 LOOP  
20 : SAME AS +*161 LOOP  
21 : STA - DOES NOT INTERFERE WITH FETCHING THE NEXT  
22 : INSTRUCTION AS MUCH AS THE ODD +*1'S UNLESS MEM  
23 : IS UNINTERLEAVED.
```

```

:0081 EXMEM
01
02 05225 005616
03 05226 005544
04 05227 005565
05 05228 005586
06 05229 005607
07 05230 177110
08 05231 060114
09 05232 063514
10 05233 000403
11 05234 063614
12 05235 060406
13 05236 060214
14 05237 004410
15 05240 065114
16 05241 065614
17 05242 000001
18 05243 064404
19 05244 065111
20 05245 063611
21 05246 000377
22 05247 021400
23 05250 040520
24 05251 040510
25 05252 021401
26 05253 040507
27 05254 021402
28 05255 040516
29 05256 126400
30 05257 172070
31 05261 155400
32 05262 045000
33 05263 187770
34 05265 000022
35 05265 135710
36 05266 125400
37 05267 146070
38 05267 005657
39 05271 101001
40 05272 010265
41 05273 006733
42 05274 000776

:DETERMINE THE INTERLEAVE FACTORS AND TYPE
:OF MEMORY'S AVAILABLE
JLPT: PSH 3,3
GNX1K
JLPT:
:SAV RTN
:RTC EXISTS
:DOONE COULD HAVE SET
:RNO RTC USE TIO
JLOPT:
SKP8 RTC
JMP *+3
SKPN RTC
JMP *+6
NIC RTC
JSR JLOPM
DAS 1, RTC
SKPN RTC
1
JSR JLOPM
DOAS 1, TIO
SKPN TIO
377
JLOPM: LDA 0,0,3
STA 0,JLOPI+3
LDA 0,1,3
STA 0,JLOPI+1
LDA 0,2,3
STA 0,TCHAR
SUB 1,1
ELEF 2,ILCTB+1,0
INC 2,3
STA 1,0,2
ADDI 18,1
BLM
EST 1,1
MOV 0,0,SKP
ISZ CURIK
JSR 3JLOPT+2
JMP *-2

:0082 EXMEM
01
02 05275 122070
03 05276 005501
04 05277 049020
05 05278 005501
06 05279 049020
07 05300 103240
08 05301 100110
09 05302 040022
10
11 05303 030171
12 05304 052020
13 05305 006720
14 05306 030401
15
16
17
18 05307 155400
19 05310 051000
20 05311 166070
21 05312 000017
22 05313 133710
23 05314 155000
24 05315 172070
25 05317 056020
26 05320 166070
27 05320 166070
28
29 05322 133710
30 05323 171000
31 05324 052020
32 05325 006700
33 05326 050561
34 05327 052020
35 05330 006675
36 05331 050543
37 05332 052020
38 05333 006672
39 05334 050525
40 05335 052020
41 05336 006667
42 05337 050507
43 05340 052020
44 05341 006664
45 05342 050061
46 05343 052020
47 05344 006661
48 05345 050440
49 05346 162070
50
51 05350 042020

:FILL THE CURRENT 1K WITH THE LDA STA
:OFFSET SEQUENCES AND TIME
:THEM INDIVIDUALLY
JLOPD: ELDA 0,INSTR,0 :START ADR TBL
STA 0,20 :20 TO FORM IT
ADDR 0,0 :SET 180
SRI 1,0
STA 0,22 :FOR EXECUTE @
:NOW SET UP THE SEQUENCES
LDA 2,SCRL0 :SAV STRT ADR.
STA 2,20
JSR 3JLOPT+3
LDA 2,*1.
:BLM THE LDA *+ SEQUENCE THAT IDENTIFIES 8K CACHE SC
:FIRST BLM IS TO ALLOW ROOM FOR THE *-31'S AND TO GET
:THE STARTING ADDRESS OF THE LOOP TO SCRL0+37
INC 2,3
WRITE TO MEM
STA 2,0,2
ELEF 1,15,*0 :IN CASE JUST POWER
BLM
MOV 2,3
ELEF 2,JLSE0,0 :ADRS OF SPEC. LDA SLO.
STA 3,20
:START ADRS TO RUN TABLE
ELEF 1,35,*0 :WRITES EXTRA FOR LDA GOOD DATA
BLM
MOV 3,2
:STRT OF STA *+161
STA 2,20
JSR 3JLOPT+3
STA 2,*161
:SHOULD INTERFER SAME AS *+1.
STA 2,20
JSR 3JLOPT+3
STA 2,143
:INTERFERS SAME AS *+3
STA 2,20
JSR 3JLOPT+3
STA 2,*125
:INTERFERS SAME AS *+5
STA 2,20
JSR 3JLOPT+3
STA 2,*107
:INTERFERS SAME AS *+7
STA 2,20
JSR 3JLOPT+3
STA 2,*61
:INTERFERS SAME AS THE *+161 LOOP
STA 2,20
JSR 3JLOPT+3
STA 2,*40
:INTERFERS SAME AS STA 2*.
ELEF 0,JLOPE,0
:AGN, 13 IN LOOP
STA 0,20

```

10083 EXMEM

```

01 05351 020001
02 05352 040566
03 05353 020412
04 05354 040001
05 05355 020416
06 05356 105520
07 05357 101120
08 05360 062077
09 05361 065114
10 05362 063614
11 05363 000777
12 05364 000403
13 05365 005366
14 05366 103110
15 05367 102400
16 05370 065114
17 05371 060177
18 05372 002022
19 05373 000003
20 05374 062677
21 05375 030040
22 05376 176400
23 05377 054930
24 05400 021371
25 05401 025372
26 05402 004461
27 05403 000455
28 05404 176070
29 05405 005515
30 05406 054416
31 05407 023373
32 05410 004453
33 05411 014413
34 05412 021000
35 05413 004450
36 05414 010413
37 05415 021376
38 05416 004445
39 05417 019410
40 05420 021375
41 05421 004442
42 05422 010405
43 05423 172070
44 05425 031000
45 05426 166070
46 05430 125120
47 05431 133000
48 05432 020265
49 05433 101220
50 05434 101220
51 05435 101220
52 05436 142010
53 05437 030503
54 05440 101001
55 05441 000365
56 05442 113370
57 05443 030475
58 05444 050001
59 05445 101520
60 05446 103120
61 05447 040265
62 05450 024871
63 05451 066000
64 05452 101300
65 05453 101202
66 05454 117710
67 05455 004467
68 05456 117710
69 05457 000616

```

10084 EXMEM

```

01 05373 000003
02 05374 062677
03 05375 030040
04 05376 176400
05 05377 054930
06 05400 021371
07 05401 025372
08 05402 004461
09 05403 000455
10 05404 176070
11 05405 005515
12 05406 054416
13 05407 023373
14 05410 004453
15 05411 014413
16 05412 021000
17 05413 004450
18 05414 010413
19 05415 021376
20 05416 004445
21 05417 019410
22 05420 021375
23 05421 004442
24 05422 010405
25 05423 172070
26 05425 031000
27 05426 166070
28 05430 125120
29 05431 133000
30 05432 020265
31 05433 101220
32 05434 101220
33 05435 101220
34 05436 142010
35 05437 030503
36 05440 101001
37 05441 000365
38 05442 113370
39 05443 030475
40 05444 050001
41 05445 101520
42 05446 103120
43 05447 040265
44 05450 024871
45 05451 066000
46 05452 101300
47 05453 101202
48 05454 117710
49 05455 004467
50 05456 117710
51 05457 000616

```

```

;NOW SYNCH WITH THE RTC OR T10
;AND CREATE THE LOOP COUNT FOR EACH SEQ
LDA 0,1
STA 0,SVONE
LDA 0,JLOPI
STA 0,1
LDA 1,1TCHAR
SUBZL 0,0
MOVZL 0,0
MSKO 0
;SYNC WITH THE FIRST DONE FLAG TO ENABLE INTA'S
;SECOND DONE FLAG INTERRUPTS BUT NO TIMING IS
;PERFORMED, IT IS SIMPLY TO ALLOW FOR OVERHEAD
;CREATED DURING THE INTERRUPT AND PUSH TO STACK
JLOPI: DOAS 1,RTC
SKPDC RTC
JMP *-1
;INTERRUPT SAVE LOOP COUNT RESTART
JLOPI: +-1
PSH 0,0
SUB 0,0
DOAS 1,RTC
INTEN
JMP @22
;DOO NEXT LOOP
;ALL LOOPS HAVE BEEN TIMED
;DETERMINE THE INTERLEAVE FACTOR
JLOPE: IORST
LDA 2,SSV
STA 3,JCTXX+1
;SET INTERLEAVE FACTOR TO 0
;GETS LDA +-1 LOOP COUNT
;GETS THE LDA SEQUENC COUNT
;SKP N.I IF CORE OR 32K SC
;APPARENTLY 8K CACHED SC
;POINTS TO TABLE POINTERS
ELEF 3,IICLB,0
STA 3,JCTBX+1
LDA 1,-5,2
;GETS THE STA +-161 COUNT
;SKP N.I. IS CORE LDA=5 STA
;NOW POINTS TO 32K SC TABLES
DSZ JCTBX+1
;GETS STA +-40 LOOP COUNT
;SKP N.I. IF UNINTERLEAVED
;MEM IS AT LEAST 2 WAY INTERLV'D
;GETS THE STA +-107 LOOP COUNT
;SKP IF NOT 4 OR 8 WAY INTERLEAVED
;AT LEAST 4 WAY INTERLEAVED
;GET THE +-125 LOOP COUNT
;SKP NEXT INSTRU. IF NOT 8 WAY
;+1 AGAIN IF 8 WAY INTERLEAVED
;EFA GETS -1'D IF 32K SC
ELEF 2,IICLB,0
JCTBX:
LDA 2,0,2
;ADDRESS OF TYPE OF MEM
;0,1,2 OR 3=8 1,2,4 OR 8 WAY INT.
ELEF 1,0,0
JCTXX:
MOVZL 1,1
ADD 1,2
LDA 0,CURIK
MOVZR 0,0
MOVZR 0,0
BTO 2,0
LDA 2,JL186+1
MOV 0,0,SKP
JMP JLOPX
MSP 2
LDA 2,SVONE
STA 2,1
INCL 0,0
ADDZL 0,0
STA 0,CURIK
LDA 1,JL186
DOB 1,0 ;REENABLE NEW MODE
MOVS 0,0
MOV 0,0,SZC
POPJ
JSR GNXIK
POPJ
JMP JLOPD
;DO NEXT 1K

```





```

10087 EXMEM
01
02
03
04
05
06
07 05565 025400
08 05566 175400
09 05567 177110
10 05570 020422
11 05571 123000
12 05572 041000
13 05573 020420
14 05574 125000
15 05575 034414
16 05576 133000
17 05577 041000
18 05600 175404
19 05601 000775
20 05602 133000
21 05603 020412
22 05604 041000
23 05605 020407
24 05606 041001
25 05607 130010
26 05610 117710
27 05611 177763
28 05612 004400
29 05613 000400
30 05614 001777
31 05615 101400
32
33
34 05616 021400
35 05617 175400
36 05620 177110
37 05621 024771
38 05622 125400
39 05623 045000
40 05624 126520
41 05625 000750

: FILL THE CURRENT 1K WITH THE 4
: JMP +OFFSET SEQ'S
: AC2=ADRS OF 1ST WORD
: JSR+1 = THE CURRENT OFFSET
: OFFSET=1,2,4 OR 8.
: JFILL: LDA 1,0,3
: INC 3,3
: PSH 3,3 :SAVE RTN AC1=OFFSET
: LDA 0,JJSR
: ADD 1,0
: STA 0,0,2
: LDA 0,JJMP
: ADD 1,0
: JFJLY: LDA 3,JKM13
: ADD 1,2
: STA 0,0,2
: INC 3,3,SZR
: JMP +3
: ADD 1,2
: LDA 0,JINCR
: STA 0,0,2
: LDA 0,JJMP1
: STA 0,1,2
: ADI 2,2
: POPJ
: JKM13: -13.
: JSR .
: JJSR: JMP .
: JJMP: JMP -1,3
: JJMP1: JMP 0,0
: JINCR: INC 0,0

: MOVE EITHER 13 LDA +17 OR STA +17
: JFJLY: LDA 0,0,5
: INC 3,3
: PSH 3,3
: LDA 1,JJSR
: INC 1,1
: STA 1,0,2
: SUBZL 1,1
: JMP JFJLY

: CREATE JMP.+ OFFSET
: STOR 13 OF THEM
: INC 0,0
: JMP -1,3 END OF LOOP

```

```

10088 EXMEM
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

: ALL 8 LOOPS HAVE BEEN TIMED
: TYPE THEIR LOOP COUNTS FOR ANALYSIS
: JLOPX: IORST
: DSZ J1STK
: JMP +3
: JSR 2LMESS
: JHDT
: JSR 2LCRLF
: LDA 1,CURLK
: JSR 2LPDEC
: ELEF 1,8,,0
: STA 1,JLCTR
: POP 1,1
: JSR 2LPCT
: DSZ JLCTR
: JMP +3
: POP 0,0
: LDA 0,SVONE
: STA 0,1
: LDA 0,CURLK
: MOVZR 0,0
: MOVZR 0,0
: EJMPL JLOP2,0
: JLCTR: 0
: TO COUNT 8 TYPEOUTS
: JLCTR: 0
: JHDT: -TXTE !<15><12>1K#
: JSR: +125
: JSR: +143
: LDA SEQ LDA +1,1
: NOLOC 0

```

```

04 05626 062677
05 05627 014430
06 05630 000403
07 05631 006100
08 05632 095660
09 05633 006101
10 05634 024265
11 05635 006104
12 05636 166070
13 00000 000010
14 05640 044416
15 05641 127210
16 05642 006102
17 05643 014413
18 05644 000775
19 05645 103210
20 05646 020672
21 05647 040001
22 05650 020265
23 05651 101220
24 05652 101220
25 05653 101220
26 05654 102070
27 005443
28 05656 000000
29 05657 000000
30 05660 005215
31 05673 025456
32 000000 -NOLOC 0

```

```

: TXTE !<15><12>1K#
: +143
: LDA SEQ LDA +1,1
: +107
: +61
: +40

```

10084 EXMEM

```

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
00715 004401
00716 030401
00717 030501
00720 030541
00721 030741
00722 030441
00723 030501
00724 030541
00725 030741
00726 030441
00727 030501
00730 030541
00731 030741
00732 030571
00733 101400
00734 001777

```

;: THE FOLLOWING SEQUENCE OF LDA INSTRUCTIONS IS  
 ; RELOCATED TO DETERMIN IF 8K CACHED SC MEMORIES  
 ; ARE AVAILABLE IN THE SYSTEM, FOR CORE OR 32K SC'S  
 ; THE LOOP RUNS THE SAME AS THE LDA +1.  
 ; FOR 8K CACHED SC'S 12 OF THE 13 LDA'S WILL INVALIDATE  
 ; AN LRU AFTER THE FIRST PASS AND CONSEQUENTLY THE LOOP  
 ; WILL RUN SIGNIFICANTLY SLOWER THAN THE LDA +1 LOOP  
 ; NOTE THAT THE JSR IS STORED AT SCRLD+37  
 JLSRD: JSR +1  
 ; STORED AT SCRLD+37  
 LDA 2, +33. ; 4 LDA'S + THE 3 EFA'S  
 LDA 2, +65. ; FILL ALL 4 CACHE'S  
 LDA 2, +97. ; FOR THE MEMORY  
 LDA 2, -31. ; THIS EFA IS FOR NEXT CACHE  
 ; \*\*\*\*\*NEW CACHE\*\*\*\*\* INSTRUCTIONS +4 EFA'S FORCE SLOW DOWN  
 LDA 2, +33.  
 LDA 2, +65.  
 LDA 2, +97.  
 LDA 2, -31.  
 ; \*\*\*\*\*NEW CACHE\*\*\*\*\*  
 LDA 2, +33.  
 LDA 2, +65.  
 LDA 2, +97.  
 LDA 2, -31.  
 ; \*\*\*\*\*NEW CACHE\*\*\*\*\* THE LDA .\*121 INVALIDS CACHE OF FIRST .\*33  
 LDA 2, +121.  
 INC 0,0  
 JMP -1,3

```

10090 EXMEM
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
00735 177110
00736 172070
00737 005516
00740 025000
00741 021001
00742 102015
00743 000420
00744 153110
00745 107110
00746 024771
00747 132640
00750 173770
00752 006101
00753 021000
00754 006212
00755 127210
00756 006102
00757 127210
00760 125004
00761 006102
00762 153210
00763 130010
00764 166070
00765 005540
00766 132414
00767 000751
00770 006101
00771 117710
00772 006017
00773 006025
00774 006033
00775 006041
00776 006047
00777 006054
00800 006061
00801 006068
00802 006073
00803 054724
00804 147703
00805 147703
00806 147703
00807 147703
00808 141523
00809 141523
00810 141523
00811 045670
00812 000000
00813 .NOLDC

```

```

;TYPE OUT THE DESCRIPTION OF THE MEMORY SYSTEM
JMMENT: PSM 3,3
ELEF 2, ILCTR+1,0 ;START OF INTERLEAVE TABLES
LDA 1,0,2 ;TWO WORD FDR 256K
LDA 0,1,2 ;SKP IF ANY MEM
ADD# 0,1, SNR ;NONE OF THIS TYPE
JMP JNEXT ;SAVE TABLE ADDRS.
PSM 2,2 ;AND TABLE
PSM 0,1
LDA 1, JMENT+2 ;AD2=S TABLE #
SUBOR 1,2 ;POINTS AT TYPE TEXT TABLE
ADDJ JTEXT, 2
JSR @LCLRF ;TEXT ADDRS
LDA 0,0,2
JSR @ERRTX ;FIRST 128K
POP 1,1 ;SECOND 128K
JSR @LPOCT
MOV 1,1, SZR
JSR @LPOCT ;PRETRIEVE TABLE ADDRS.
POP 2,2 ;NEXT TABLE
ADI 2,2
ELEF 1, SVONE, 0
SUB# 1,2, SZR ;SKP=S DONE ALL
JMP JMENT+3 ;CHECK NEXT TYPE
JSR @LCLRF
POPJ
;TABLE OF TEXT ADDRESSES
JTEXT: JCOR0
JCOR2
JCOR4
JCOR6
JSC0
JSC2
JSC4
JSC6
JSC8
JSC10
JHOR2: ;TXTE ITYPE
;TXTE ICORE 0 WAY
;TXTE ICORE 2 WAY
;TXTE ICORE 4 WAY
;TXTE ICORE 8 WAY
;TXTE ISC 0 WAY
;TXTE ISC 2 WAY
;TXTE ISC 4 WAY
;TXTE ISC 8 WAY
;TXTE ISC CACHED
;TXTE 18K SC
;TXTE 128K
;TXTE 256KI
00814 000000
00815 000000
00816 000000
00817 000000
00818 000000
00819 000000
00820 000000
00821 000000
00822 000000
00823 000000
00824 000000
00825 000000
00826 000000
00827 000000
00828 000000
00829 000000
00830 000000
00831 000000
00832 000000
00833 000000
00834 000000
00835 000000
00836 000000
00837 000000
00838 000000
00839 000000
00840 000000
00841 000000
00842 000000
00843 000000
00844 000000
00845 000000
00846 000000
00847 000000
00848 000000
00849 000000
00850 000000
00851 000000
00852 000000
00853 000000
00854 000000
00855 000000
00856 000000
00857 000000
00858 000000
00859 000000
00860 000000
00861 000000
00862 000000
00863 000000
00864 000000
00865 000000
00866 000000
00867 000000
00868 000000
00869 000000
00870 000000
00871 000000
00872 000000
00873 000000
00874 000000
00875 000000
00876 000000
00877 000000
00878 000000
00879 000000
00880 000000
00881 000000
00882 000000
00883 000000
00884 000000
00885 000000
00886 000000
00887 000000
00888 000000
00889 000000
00890 000000
00891 000000
00892 000000
00893 000000
00894 000000
00895 000000
00896 000000
00897 000000
00898 000000
00899 000000
00900 000000
00901 000000
00902 000000
00903 000000
00904 000000
00905 000000
00906 000000
00907 000000
00908 000000
00909 000000
00910 000000
00911 000000
00912 000000
00913 000000
00914 000000
00915 000000
00916 000000
00917 000000
00918 000000
00919 000000
00920 000000
00921 000000
00922 000000
00923 000000
00924 000000
00925 000000
00926 000000
00927 000000
00928 000000
00929 000000
00930 000000
00931 000000
00932 000000
00933 000000
00934 000000
00935 000000
00936 000000
00937 000000
00938 000000
00939 000000
00940 000000
00941 000000
00942 000000
00943 000000
00944 000000
00945 000000
00946 000000
00947 000000
00948 000000
00949 000000
00950 000000
00951 000000
00952 000000
00953 000000
00954 000000
00955 000000
00956 000000
00957 000000
00958 000000
00959 000000
00960 000000
00961 000000
00962 000000
00963 000000
00964 000000
00965 000000
00966 000000
00967 000000
00968 000000
00969 000000
00970 000000
00971 000000
00972 000000
00973 000000
00974 000000
00975 000000
00976 000000
00977 000000
00978 000000
00979 000000
00980 000000
00981 000000
00982 000000
00983 000000
00984 000000
00985 000000
00986 000000
00987 000000
00988 000000
00989 000000
00990 000000
00991 000000
00992 000000
00993 000000
00994 000000
00995 000000
00996 000000
00997 000000
00998 000000
00999 000000

```

10091 EXNEM

.TITL EXNEM

.TXT /COPYRIGHT(C)08C,1974,1975,1976

02 AC2MD 000161 13/46 73/13 73/50  
 03 ALT8L 000067 12/10 25/27 28/08 30/04 30/05 30/09 32/03  
 04 054520 35/07 35/44 36/08 36/12 36/22 57/10 58/17  
 05 044522 10/21 29/26 29/34 65/07  
 06 044107 10/21 35/05 35/23 36/16  
 07 024124 35/11 35/15  
 08 024503 35/13 35/20 35/31 35/43 36/07  
 09 043504 35/06 35/20 35/31 35/50 35/55 35/56  
 10 026103 35/10 35/19 35/22 35/26 35/29 35/50 35/55 35/56  
 11 034461 35/18 35/22 36/23 36/24  
 12 032067 36/10 36/23 36/24  
 13 030454 23/11 27/31 35/25 85/06 91/30  
 14 033471 10/43 69/39 10/39 10/40 10/41 10/42  
 15 026065 10/36 10/37 10/38 10/46 10/47 10/48 13/15  
 16 034461 10/43 10/44 10/45 10/46 10/47 10/48 13/15  
 17 033067 13/39 69/12 69/44 72/16 72/17  
 18 06121 046101 ALL RIGHTS RESERVED/  
 19 020114 13/39 63/04  
 20 044522 13/48 61/08 63/04  
 21 044107 61/15 63/13  
 22 051524 61/22 63/22  
 23 051040 61/29 63/31  
 24 051505 13/35 46/04  
 25 051105 13/30 69/23 70/25  
 26 048526 13/33 71/29 71/44 73/49  
 27 000104 13/29 69/15 70/06 71/25  
 28 06133 00020 EXISM: -8LK 16.  
 29 06153 00020 AVALM: -8LK 16.  
 30 06173 00000 LSYSE: 0  
 31 06174 154305 DIRT: -TXTE IEX.MEMEX 041  
 32 046456 13/07 64/50 27/23 27/24 27/32  
 33 046705 11/06 64/39 65/16  
 34 154305 69/04 69/04  
 35 030240 57/56 68/17 69/02  
 36 000264 27/16 27/25 27/35  
 37 000000 -NOLDC 0 60/13 60/17 60/34 60/38  
 38 000000 60/39 64/26 67/57 66/30  
 39 000000 65/39 66/03 65/05 66/30  
 40 06202 00000 65/08 67/03 67/33  
 41 06203 00000 60/17 60/33 67/31  
 42 06204 175772 64/44 67/31 67/56  
 43 06205 00000 60/30 67/56  
 44 06206 00000 60/32 60/37  
 45 06207 00000 65/36 65/40  
 46 06210 00000 13/14 67/37 67/41 67/46 68/01  
 47 06211 00000 13/15 66/09  
 48 00000 68/11 68/39  
 49 00000 68/41 68/47  
 50 00000 68/46 68/49  
 00000 68/24 68/51  
 00000 60/23 60/24 68/16  
 00000 68/29 68/32  
 00000 13/23 66/20 68/09  
 00000 13/37 66/22 68/09

13/46 73/13 73/50  
 12/10 25/27 28/08 30/04 30/05 30/09 32/03  
 35/07 35/44 36/08 36/12 36/22 57/10 58/17  
 10/21 29/26 29/34 65/07  
 10/21 35/05 35/23 36/16  
 35/11 35/15  
 35/13 35/20 35/31 35/43 36/07  
 35/06 35/20 35/31 35/50 35/55 35/56  
 35/10 35/19 35/22 35/26 35/29 35/50 35/55 35/56  
 35/18 35/22 36/23 36/24  
 36/10 36/23 36/24  
 23/11 27/31 35/25 85/06 91/30  
 10/43 69/39 10/39 10/40 10/41 10/42  
 10/36 10/37 10/38 10/46 10/47 10/48 13/15  
 10/43 10/44 10/45 10/46 10/47 10/48 13/15  
 13/39 69/12 69/44 72/16 72/17  
 13/39 63/04  
 13/48 61/08 63/04  
 61/15 63/13  
 61/22 63/22  
 61/29 63/31  
 13/35 46/04  
 13/30 69/23 70/25  
 13/33 71/29 71/44 73/49  
 13/29 69/15 70/06 71/25  
 13/32 69/31 70/26  
 68/33 68/54  
 13/43 64/33  
 23/44 27/07  
 27/17 27/19 27/23 27/24 27/32  
 13/07 64/50 67/26  
 13/04  
 11/06 64/39 65/16  
 69/04 69/04  
 68/42 68/17 69/02  
 57/56 68/17 69/02  
 27/16 27/25 27/35  
 60/13 60/17 60/34 60/38  
 60/39 64/26 67/57 66/30  
 65/39 66/03 65/05 66/30  
 65/08 67/03 67/33  
 60/17 60/33 67/31  
 64/44 67/31 67/56  
 60/30 67/56  
 60/32 60/37  
 65/36 65/40  
 13/14 67/37 67/41 67/46 68/01  
 13/15 66/09  
 68/11 68/39  
 68/41 68/47  
 68/46 68/49  
 68/24 68/51  
 60/23 60/24 68/16  
 68/29 68/32  
 13/23 66/20 68/09  
 13/37 66/22 68/09

| 0093 EXMEM    |       |       |       |       |       |       |       |  |  | 0094 EXMEM   |       |       |       |       |       |       |  |  |  |
|---------------|-------|-------|-------|-------|-------|-------|-------|--|--|--------------|-------|-------|-------|-------|-------|-------|--|--|--|
| CB-ES 000113  | 13/06 | 65/35 | 68/10 | 68/39 |       |       |       |  |  | DIXX 000174  | 72/16 |       |       |       |       |       |  |  |  |
| CB-EX 004357  | 68/27 | 68/48 |       |       |       |       |       |  |  | DIYY 000176  | 72/17 |       |       |       |       |       |  |  |  |
| CB-LC 000134  | 13/24 |       |       |       |       |       |       |  |  | DCAK 003151  | 50/39 |       |       |       |       |       |  |  |  |
| CB-PX 004437  | 67/42 | 69/06 |       |       |       |       |       |  |  | ECAD 000003  | 19/05 |       |       |       |       |       |  |  |  |
| CB-SE 000137  | 13/27 |       |       |       |       |       |       |  |  | ECBE 003717  | 59/37 | 19/05 |       |       |       |       |  |  |  |
| CB-TI 000135  | 13/25 | 67/04 | 67/09 | 67/09 |       |       |       |  |  | ECME 004537  | 70/28 | 59/17 |       |       |       |       |  |  |  |
| CB-TK 000112  | 13/05 | 57/47 | 64/27 | 66/05 | 67/03 | 67/09 | 68/21 |  |  | ECM6 003657  | 59/11 | 59/16 |       |       |       |       |  |  |  |
|               | 69/39 |       |       |       |       |       |       |  |  | ECSE 000064  | 59/06 | 59/16 | 64/40 | 67/23 |       |       |  |  |  |
| CB-TS 000223  | 10/35 | 66/06 |       |       |       |       |       |  |  | ECST 003617  | 12/07 | 16/11 | 56/37 |       |       |       |  |  |  |
| CB-WK 000150  | 13/16 |       |       |       |       |       |       |  |  | ECTX 003610  | 57/37 | 57/42 | 59/32 |       |       |       |  |  |  |
| CB-XI 004234  | 66/14 | 66/23 | 66/27 | 67/07 | 67/16 | 67/21 | 67/32 |  |  | EC.00 003403 | 57/33 | 59/25 | 58/44 | 59/32 |       |       |  |  |  |
|               | 67/38 | 67/55 | 68/02 |       |       |       |       |  |  | EC.01 003421 | 50/33 | 55/08 | 56/05 | 56/20 | 56/35 | 57/02 |  |  |  |
|               | 19/25 | 29/16 |       |       |       |       |       |  |  | EC.02 003453 | 57/32 | 58/04 | 58/26 |       |       |       |  |  |  |
| CD19P 001502  | 29/34 |       |       |       |       |       |       |  |  | EC.03 003486 | 55/12 | 56/04 | 56/04 | 56/31 |       |       |  |  |  |
| CD-EX 001524  | 29/34 |       |       |       |       |       |       |  |  | EC.04 003486 | 55/13 | 55/18 | 55/19 | 56/04 | 56/31 |       |  |  |  |
| CD-T1 001542  | 29/49 |       |       |       |       |       |       |  |  | EC.05 003486 | 56/34 | 57/01 |       |       |       |       |  |  |  |
| CD-T2 001543  | 29/18 | 29/38 | 29/50 | 29/42 | 29/51 |       |       |  |  | EC.06 003486 | 56/06 | 56/29 |       |       |       |       |  |  |  |
| CD-T3 001544  | 29/20 | 29/36 | 29/37 | 29/42 | 29/51 |       |       |  |  | EC.07 003486 | 57/09 | 58/18 | 58/45 |       |       |       |  |  |  |
| CH4T3 002405  | 41/24 | 41/37 | 41/41 | 41/28 | 42/01 | 42/35 |       |  |  | EC.08 003486 | 57/15 | 58/18 | 58/29 | 58/36 |       |       |  |  |  |
| CH4T2 002375  | 40/42 | 40/45 | 41/21 | 41/28 | 42/01 | 42/35 |       |  |  | EC.09 003486 | 57/15 | 58/09 | 58/29 | 58/36 |       |       |  |  |  |
| CHEC2 004517  | 70/07 | 70/12 | 70/27 |       |       |       |       |  |  | EC.10 003486 | 57/14 | 57/25 | 58/10 | 58/37 | 59/07 |       |  |  |  |
| CHEC1 004517  | 70/07 | 70/12 | 70/27 |       |       |       |       |  |  | EC.11 003486 | 57/13 | 58/35 | 59/29 |       |       |       |  |  |  |
| CHR?? 002426  | 41/58 | 41/55 | 41/44 | 42/08 | 43/60 |       |       |  |  | EC.12 003486 | 57/11 | 58/16 | 58/38 |       |       |       |  |  |  |
| CHR?? 002424  | 41/58 | 41/55 | 41/44 | 42/08 | 43/60 |       |       |  |  | EC.13 003486 | 57/07 | 58/24 | 58/43 |       |       |       |  |  |  |
| CLF? 002414   | 12/23 | 41/46 |       |       |       |       |       |  |  | EC.14 003486 | 37/16 | 38/28 | 39/39 |       |       |       |  |  |  |
| CHAPB 001475  | 21/42 | 27/30 | 28/24 | 35/57 | 35/21 | 35/47 | 35/51 |  |  | EC.15 003486 | 37/16 | 38/28 | 39/39 |       |       |       |  |  |  |
| CNDUA 000107  | 13/02 | 16/09 | 35/10 | 35/14 | 35/21 | 35/47 | 35/51 |  |  | EC.16 003486 | 36/23 | 38/28 | 39/39 |       |       |       |  |  |  |
|               | 65/10 |       |       |       |       |       |       |  |  | EC.17 003486 | 56/25 | 58/28 | 56/32 |       |       |       |  |  |  |
| CUR1K 000265  | 11/25 | 16/16 | 81/40 | 84/34 | 84/47 | 86/08 | 88/10 |  |  | EC.18 003486 | 56/25 | 58/28 | 56/32 |       |       |       |  |  |  |
| CURPR 000170  | 88/22 | 14/21 | 25/02 | 25/06 | 25/19 | 38/29 | 57/12 |  |  | EC.19 003486 | 57/05 | 58/22 | 58/42 |       |       |       |  |  |  |
|               | 13/54 |       |       |       |       |       |       |  |  | EC.20 003486 | 58/32 | 59/06 | 59/06 |       |       |       |  |  |  |
| DEC70 002346  | 58/15 |       |       |       |       |       |       |  |  | EC.21 003486 | 58/41 | 59/15 | 59/29 |       |       |       |  |  |  |
| DEC?P 002363  | 41/04 | 41/22 | 43/09 | 43/17 |       |       |       |  |  | EC.22 003486 | 58/40 | 59/15 | 59/29 |       |       |       |  |  |  |
| DEC?T 002531  | 41/12 | 41/17 |       |       |       |       |       |  |  | EC.23 003486 | 56/40 | 57/04 | 57/39 | 68/01 | 68/04 | 68/27 |  |  |  |
| DED?T 002355  | 40/58 | 43/17 |       |       |       |       |       |  |  | EC.24 003486 | 11/19 | 11/27 | 23/29 | 67/39 | 68/04 | 68/27 |  |  |  |
| DIR?T 000125  | 41/11 | 41/16 | 64/05 | 64/24 | 70/39 | 70/50 | 73/37 |  |  | EC.25 003486 | 68/53 | 68/53 |       |       |       |       |  |  |  |
|               | 13/16 | 13/17 | 64/05 | 64/24 | 70/39 | 70/50 | 73/37 |  |  | EINTP 002773 | 10/27 | 48/05 |       |       |       |       |  |  |  |
|               | 73/48 | 74/06 | 74/43 |       |       |       |       |  |  | EINTS 002114 | 10/27 | 48/05 |       |       |       |       |  |  |  |
| DIRT 006174   | 10/08 | 91/33 |       |       |       |       |       |  |  | EIS3 003003  | 48/05 | 48/12 | 48/13 |       |       |       |  |  |  |
| DISCM 004634  | 11/41 | 71/47 | 71/50 |       |       |       |       |  |  | ENTPA 001546 | 30/03 | 36/27 |       |       |       |       |  |  |  |
| DISDL 004733  | 73/14 | 73/52 | 74/06 |       |       |       |       |  |  | EPROG 000124 | 13/16 |       |       |       |       |       |  |  |  |
| DISDO 004615  | 71/28 | 71/32 |       |       |       |       |       |  |  | ERBEL 002204 | 37/30 | 37/33 | 38/10 | 38/21 | 38/27 |       |  |  |  |
| DISEX 004647  | 71/28 | 71/48 | 72/18 | 72/20 | 73/01 |       |       |  |  | ERBIS 003547 | 56/28 | 57/43 | 58/03 |       |       |       |  |  |  |
| DISFA 004642  | 72/20 |       |       |       |       |       |       |  |  | ERCTR 003201 | 37/19 | 37/23 | 37/27 |       |       |       |  |  |  |
| DISLO 004640  | 72/18 |       |       |       |       |       |       |  |  | ERCTX 003661 | 57/21 | 59/18 |       |       |       |       |  |  |  |
| DISLP 004736  | 64/04 | 74/09 | 74/42 |       |       |       |       |  |  | EREXI 002447 | 37/32 | 37/36 | 39/03 | 39/25 | 39/40 |       |  |  |  |
| DISLP 004736  | 73/05 | 73/08 | 73/49 |       |       |       |       |  |  | ERKMQ 002202 | 37/18 | 37/28 |       |       |       |       |  |  |  |
| DISLP 004726  | 13/19 | 73/01 | 73/34 | 73/53 |       |       |       |  |  | ERKML 002233 | 38/19 | 38/19 |       |       |       |       |  |  |  |
| DISRT 000127  | 71/17 | 71/31 | 71/46 |       |       |       |       |  |  | ERKPL 002233 | 37/20 | 37/20 |       |       |       |       |  |  |  |
| DISSTL 004600 | 10/38 | 10/39 | 10/40 | 10/45 | 10/46 | 10/47 | 11/06 |  |  | ERKPP 002213 | 37/20 | 37/20 |       |       |       |       |  |  |  |
| DISTU 004565  | 72/22 |       |       |       |       |       |       |  |  | ERR1 000151  | 13/37 | 70/14 | 70/23 | 73/25 |       |       |  |  |  |
|               | 72/22 |       |       |       |       |       |       |  |  | ERRK9 002200 | 37/17 | 37/26 | 70/23 | 73/25 |       |       |  |  |  |
| DISXL 004674  | 72/21 | 73/22 |       |       |       |       |       |  |  | ERRK9 002200 | 37/17 | 37/26 | 70/23 | 73/25 |       |       |  |  |  |
| DISXX 004657  | 73/09 | 73/33 |       |       |       |       |       |  |  | ERRK9 002200 | 37/17 | 37/26 | 70/23 | 73/25 |       |       |  |  |  |
| DIXBT 000132  | 13/22 | 71/35 | 73/17 | 73/20 | 74/45 |       |       |  |  | ERRK9 002200 | 10/29 | 29/27 | 57/52 | 68/26 |       |       |  |  |  |
| DIXCM 004711  | 71/47 | 71/49 | 73/36 | 73/47 |       |       |       |  |  | ERRK9 002200 | 10/29 | 29/27 | 57/52 | 68/26 |       |       |  |  |  |
| DIXCT 000126  | 13/18 | 71/37 | 73/26 | 73/27 |       |       |       |  |  | ERRK9 002200 | 10/28 | 37/05 |       |       |       |       |  |  |  |
| DIXNX 000130  | 13/20 | 64/09 | 64/20 | 71/33 | 73/09 | 73/15 | 73/28 |  |  | ERRK9 002200 | 10/28 | 37/05 |       |       |       |       |  |  |  |
|               | 73/31 |       |       |       |       |       |       |  |  | ERRK9 002200 | 10/25 | 52/10 | 52/34 | 52/36 | 52/36 | 53/16 |  |  |  |
| DIXOR 005001  | 73/10 | 73/16 | 74/45 |       |       |       |       |  |  | ERRK9 002200 | 10/25 | 52/10 | 52/34 | 52/36 | 52/36 | 53/16 |  |  |  |
| DIXST 000131  | 13/21 | 71/32 | 71/43 | 73/11 | 73/36 | 73/51 |       |  |  | ERRK9 002200 | 77/35 | 90/19 | 67/45 | 68/20 | 68/45 |       |  |  |  |
|               | 13/21 | 71/32 | 71/43 | 73/11 | 73/36 | 73/51 |       |  |  | ERTOT 000074 | 12/15 | 16/12 | 37/09 | 77/46 |       |       |  |  |  |





| 0099 EXNEM    |       | 0100 EXNEM |             |
|---------------|-------|------------|-------------|
| LWSB 000773   | 15/06 | 18/05      |             |
| LWS_1 001002  | 18/12 | 18/26      |             |
| LWS_2 001012  | 18/20 | 18/23      |             |
| LWS_3 001025  | 18/31 | 18/34      |             |
| LW_05 001053  | 19/08 | 19/22      | 37/26       |
| LW_10 001054  | 19/10 | 19/23      |             |
| LW_K1 001046  | 18/11 | 18/25      | 19/17       |
| LW_K1 001043  | 18/05 | 19/14      |             |
| LW_K2 001044  | 19/15 |            |             |
| LW_K3 001045  | 19/16 |            |             |
| LW_K4 001047  | 18/18 | 19/18      |             |
| LW_K6 001050  | 18/28 | 19/19      |             |
| LW_K7 001051  | 18/29 | 19/20      |             |
| LW_K8 001052  | 19/21 |            |             |
| LW_K9 001352  | 25/13 | 25/32      |             |
| LW_S3 001042  | 18/08 | 19/12      |             |
| LZMAX 000173  | 13/57 | 19/43      |             |
| LZOCX 000103  | 12/28 | 12/29      | 53/12 57/30 |
| L_393 001417  | 26/04 | 26/40      |             |
| M128S 001210  | 22/16 | 22/24      | 22/31       |
| M128T 001151  | 21/19 | 21/35      |             |
| M32E2 001154  | 21/20 | 21/31      | 21/35       |
| M32TE 001154  | 21/21 | 40/30      | 40/43       |
| MES37 002304  | 12/21 | 40/30      | 40/43       |
| MES37R 002427 | 40/30 | 40/33      | 40/51       |
| MES37R 002427 | 42/21 |            | 42/12       |
| MIM15 000146  | 13/34 | 64/10      |             |
| M031Z 000110  | 13/03 | 35/48      |             |
| MP2TX 003336  | 52/18 | 54/35      |             |
| MPASG 000117  | 13/10 | 16/08      | 67/48       |
| MPF32 001215  | 22/36 |            |             |
| MP8T 000066   | 12/09 | 16/14      | 16/30       |
| MP18L 001766  | 50/06 | 52/14      | 58/30       |
| MP18P 001765  | 30/26 | 31/06      | 32/28       |
| MPXTX 003325  | 32/38 | 33/13      | 30/42       |
| MP_K1 001055  | 52/20 | 54/31      | 32/30       |
| MP_K2 001056  | 19/04 | 19/24      |             |
| MS128 001161  | 19/06 | 19/25      |             |
| MSKRG 000005  | 20/19 | 22/08      | 22/30       |
| MS732 001120  | 10/06 | 48/35      |             |
| MS_K 001216   | 20/09 | 21/10      | 21/34       |
| MS_L1 001160  | 22/09 | 22/37      |             |
| MVETA 001217  | 21/28 | 21/42      | 22/27       |
| MWID 003160   | 15/09 | 23/03      | 23/09       |
| NEWK1 003163  | 15/11 | 51/10      |             |
| NEWNP 000065  | 12/08 | 16/15      | 16/48       |
| NEWTT 000003  | 30/37 | 31/03      | 31/12       |
| NEWYJ 003331  | 52/15 |            | 32/35       |
| NBERC 000105  | 18/46 | 55/05      | 30/23       |
| NPR06 001312  | 52/22 | 75/10      | 50/10       |
| NTPA 002151   | 12/34 |            |             |
| OPAD1 000163  | 24/09 | 24/11      | 24/29       |
| OPADK 000160  | 35/27 | 36/21      | 36/27       |
| OPADL 000163  | 40/55 | 43/09      |             |
| OPADK 000160  | 13/45 | 71/18      | 73/38       |
| OTDGL 000136  | 13/26 | 13/36      | 66/19       |
| PC177 002576  | 43/52 | 44/03      | 45/57       |
| PC177 002574  | 40/35 | 41/27      |             |
| PAC21 002372  | 40/31 | 40/46      |             |
| PAC22 002373  | 42/56 | 40/47      | 41/25       |
| PATTA 000115  | 40/32 | 40/45      | 41/26       |
| PATTB 000116  | 13/08 | 64/45      | 69/16       |
| PC121 002425  | 69/43 | 70/05      | 67/13       |
| PC122 002535  | 13/09 | 64/46      | 67/12       |
| PC125 002600  | 70/10 |            |             |
| PC420 002516  | 41/31 | 42/07      |             |
| PC670 002515  | 42/02 | 43/22      | 46/05       |
| PCE7 002423   | 42/20 | 46/05      | 46/30       |
| PDECI 000211  | 41/49 | 43/58      | 43/50       |
| PDEZ 002337   | 42/36 | 43/06      | 46/36       |
| PENDA 000071  | 40/53 | 41/14      | 46/12       |
| PEXPT 002324  | 41/39 | 42/05      |             |
| PEXPT 002324  | 10/24 | 57/40      | 77/39       |
| PFAIS 000075  | 12/12 | 25/14      | 31/23       |
| PFATT 001421  | 40/46 | 42/17      | 26/33       |
| PL92T 002554  | 12/17 | 14/17      | 50/05       |
| PL92T 002554  | 26/37 | 26/48      |             |
| PL92T 002554  | 43/38 | 43/42      |             |
| PL92T 002554  | 40/45 | 41/08      |             |
| PL92T 002554  | 40/45 | 41/08      |             |
| PL92T 002554  | 12/26 | 40/53      |             |
| PONES 000152  | 13/38 |            |             |
| PROBK 001073  | 18/09 | 19/39      | 19/45       |
| PRSTR 000070  | 12/11 | 25/10      | 26/05       |
| PTB7 002540   | 41/06 | 43/27      | 46/34       |
| PWR0N 000076  | 12/18 | 51/71      | 46/60       |
| PRWK 003124   | 50/13 | 50/14      | 50/16       |
| PWRUP 000077  | 12/19 | 49/30      |             |
| RETRN 000213  | 10/26 | 56/42      | 57/55       |
| RETUR 000140  | 13/28 | 69/12      | 69/38       |
| RISCX 002124  | 71/22 |            |             |
| RSCRA 000210  | 10/23 | 36/06      |             |
| RICTR 005140  | 77/13 | 77/14      | 77/20       |
| RTFIV 005055  | 78/24 | 78/25      | 78/27       |
| RTMIN 005137  | 76/26 | 77/11      |             |
| RTSEC 005136  | 77/10 | 77/41      | 78/15       |
| RTSR 005167   | 77/08 | 78/03      | 78/19       |
| RTLEX 005175  | 78/12 | 78/16      | 78/14       |
| RTT4M 000672  | 77/32 | 78/30      | 79/10       |
| RT_01 005007  | 12/15 | 26/06      | 77/17       |
| RT_01 005007  | 50/32 | 75/13      | 77/36       |
| RT_02 005057  | 75/17 | 76/03      | 76/14       |
| RT_03 005103  | 75/18 | 76/16      | 76/17       |
| RT_04 005106  | 77/26 | 77/27      | 77/02       |
| RT_05 005144  | 76/25 | 78/09      | 77/29       |
| RT_K1 005052  | 76/18 | 76/23      | 77/48       |
| RT_K2 005053  | 76/19 | 76/24      |             |
| RT_K3 005054  | 76/20 | 76/25      |             |
| RT_K4 005105  | 77/03 | 77/26      |             |
| RT_K5 005135  | 77/07 | 78/02      |             |
| RT_K6 005134  | 77/09 | 77/40      |             |
| SC32K 005526  | 85/32 | 85/35      |             |
| SC8KC 005536  | 85/05 | 85/36      |             |











LISTING

096-000335-03

PROGRAM

ECLIPSE SPECIAL EXERCISER

TAPE

095-000335-03

ABSTRACT

'ESPCLEX' IS A SPECIAL EXERCISER PROGRAM DEVELOPED FOR EXERCISING ALL THE CENTRAL PROCESSOR INSTRUCTIONS OF ECLIPSE IN BOTH UNMAPPED AND MAPPED MODE WITH ERCC OPTION ENABLED IN MODE 03. IT IS ALSO DESIGNED TO HANDLE POWER FAIL/AUTO RESTART.



0001 ESPCL MACRO REV 03.00  
.TITLE EPACK

13:41:26 12/31/76

10002 ESPCL

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

```
*****  
: *NAME: ESPCLX.SR          PART NUMBER: 094-000742  
: *  
: *DESCRIPTION: ECLIPSE SPECIAL EXERCISER  
: *  
: *REVISION HISTORY:  
: *   REV.      DATE  
: *   00      11/07/75  
: *   01      02/20/76  
: *   02      08/06/76  
: *   03      12/31/76  
: *  
: * COPYRIGHT (C) DATA GENERAL CORPORATION, 1975, 1976  
: * ALL RIGHTS RESERVED.  
: *****
```

0003 ESPCL

01  
02  
03  
04  
05  
06  
07

ESPCLX

SPECIAL EXERCISER FOR ECLIPSE

10004 ESPCL

01

02

03

04

05

06

07

EXERCISER FOR ECLIPSE: SPECIAL

PROGRAM NAME

-----

ESPCLX

GENERAL DESCRIPTION

-----

'ESPCLX' IS A SPECIAL EXERCISER PROGRAM DEVELOPED FOR EXECISING ALL THE CENTRAL PROCESSOR INSTRUCTIONS OF ECLIPSE IN BOTH UNMAPED AND MAPED MODE WITH ERCC OPTION ENABLED IN MODE 03. IT IS ALSO DESIGNED TO HANDLE POWER FAIL/AUTO RESTART. IN CASE OF POWER FAIL, THE PROG WILL PRINT 'POWER FAIL' AND RESTART FOR AUTO RESTART. FOR NO AUTO RESTART, THE PROG WILL HALT. PRESS CONTINUE TO PROCEED. IT WILL PRINT 'POWER FAIL' AND THE PROGRAM WILL RESTART. IT MUST BE RUN WITH 'CAT' AND 'KITEN' PROGRAM IN THE BACKGROUND TO ASSURE OF THE PROPER OPERATION OF THESE INSTRUCTIONS IN CASE OF OCH REQUESTS AND/OR INTERRUPTS.

THE INSTRUCTIONS EXERCISED ARE AS FOLLOWS:

ALL ECLIPSE INSTRUCTIONS

LOCATIONS 200 TO 216 IN PAGE 0 ARE FIXED FOR 'ESPCLX' PROGRAM AND THE USE OF THESE LOCATIONS ARE AS FOLLOWS  
LOCATION 200 IS THE STARTING ADDRESS OF THIS PROGRAM.  
LOCATION 201 KEEPS TRACK OF RELOCATED ADDR OF THE TEST CURRENTLY RUNNING AND IS USEFUL FOR DEBUG WHEN LOOPING OCCURS IN THE PROGRAM.  
LOCATION 202 CONTAINS THE STARTING ADDR OF THE PROGRAM.  
LOCATION 203 SHOWS NUMBER OF PASSES RUN THROUGH THIS PROGRAM.

LOCATION 204 SHOWS INTERNAL PASS COUNT WHICH IS FIXED BY LOCATION 205.  
LOCATION 207 IS THE CURRENT PASS COUNT FOR INDIVIDUAL TEST AND SHOWS THE PASSES REMAINING THRU THIS TEST AT A PARTICULAR TIME.  
LOCATION 214 IS THE BASE OFFSET USED TO CALCULATE THE CURRENT RELOCATION OF THE PROGRAM.  
LOCATION 215 KEEPS TRACK OF THE LISTING ADDR OF THE TEST CURRENTLY RUNNING AND IS USEFUL FOR DEBUG WHEN LOOPING OCCURS IN THE PROGRAM.

LOCATION 216 KEEPS TRACK OF THE CURRENT TEST# (TALLY) RUNNING AND IS USEFUL FOR DEBUG WHEN RUNNING UNDER A NORMAL PROGRAM EXECUTION.

\*2.3.1 NOTE:

LOCATION 214 (TEST#) IS ADVANCED EACH TIME THAT THE "SETUP" MACRO IS EXECUTED. FOR STAND ALONE SUBTEST EXECUTION, THE SIGNIFICANCE OF THIS ENTRY IS ONLY THAT OF A TALLY OF SUBTESTS ENTERED.

60





10007 ESPCL

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55

SWITCH SETTINGS  
-----

THIS PROGRAM USES THE DATA SWITCHES AS FOLLOWS:

4.1

SW"0" - USE CONTENTS OF "SWREG" IF 0  
 USE DATA SWITCHES IF 1  
 SW"1" - LOOP ON FAILING TEST IF 0  
 PROCEED TO NEXT TEST IF 1  
 SW"2" - PRINT TO DISPLAY (TTY) IF 0  
 DO NOT DISPLAY (TTY) IF 1  
 SW"3" - DO NOT PRINT % ERRORS IF 0  
 PRINT FAILURE RATE IF 1  
 SW"4" - PRINT 'PASS XX' IF 0  
 DO NOT PRINT 'PASS XX' IF 1  
 SW"5" - DO NOT OUTPUT TO 'LPT' IF 0  
 PRINT OUTPUT TO 'LPT' IF 1

SW"8" - ENABLE MPMU/MMPUI MAP DUMP IF = 1  
 DISABLE MPMU/MMPUI MAP DUMP IF = 0

SW"15" - ENABLE QUICK VERIFY (QV) MODE EXECUTION  
 IF BIT (15) OF "SWREG" IS SET = 1.

NOTE: SW"15" IS ONLY SELECTABLE IF SET USING THE DTOS  
 "SWREG" COMMAND AND RESPONDING WITH "I(CR)".  
 SEE ITEM 9.1 BELOW !!!

EDTOS PRESUMES THAT ALL SWITCHES ARE 0. RETURN IS  
 MADE TO THE OPER SYS UPON AN ERROR AFTER FIRST PRINTING  
 THE FAILURE REPORT. IN STAND ALONE MODE, THE OPERATOR  
 IS ASKED TO PRESET THESE SWITCHES (THE MACHINE HALTS  
 FOR HIM TO DO SO) AND PRESS CONTINUE. THIS PRESET VALUE  
 CAN BE CHANGED AT ANY TIME BY SETTING DATA SWITCH "0"  
 TO A "1". THE PROGRAM WILL THEN FOLLOW ACTUAL SETTINGS.

STARTING ADDRESS = 200 IN STAND ALONE MODE.  
 IF 'CAT' OR 'KITEN' WAS LOADED FROM DTOS AND RESTART  
 WAS NEEDED, THEN USE AS FOLLOWS:  
 STARTING ADDR = 170 (FOR START WITH NO 'CAT')  
 STARTING ADDR = 171 (FOR START WITH 'CAT')

MONITOR LOCATION 203 TO CHECK THE CURRENT PASS COUNT.

MONITOR LOCATION X6000 TO MAKE SURE THAT 'CAT' OR  
 'KITEN' IS RUNNING, IN CASES WHERE THE PROGRAM IS  
 STARTED WITH 'CAT' OR 'KITEN'. ( X = THE NUMBER  
 OF THE HIGHEST MEMORY MODULE IN THE SYSTEM AND IS  
 1 THR' 7) MODULO 8.K.

LOCATION X6000 MUST HAVE PATTERN CHANGING FROM ZEROES  
 TO ALL ONES TO INC/SWAP PATTERN.

1000A ESPCL

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17

OPERATING PROCEDURE/OPERATOR INPUT  
-----

LOAD THE PROGRAM VIA THE BINARY LOADER OR INSERT A  
 PRELOADED MEMORY MODULE.  
 SET SWITCHES TO 200.  
 PROGRAM WILL HALT AFTER PRINTING THE MESSAGE  
 'SET DATA SWITCHES AND PRESS CONTINUE'.  
 SET REQUIRED SWITCH SETTINGS AND PRESS CONTINUE.  
 THE PROGRAM WILL RUN UNTIL MANUALLY STOPPED. IN CASE  
 OF MALFUNCTIONING, THE PROGRAM WILL PRINT ERROR  
 MESSAGE AND TAKE APPROPRIATE ACTION AS PER THE SW  
 SETTINGS.



10011 FSPCL

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

PROGRAM DESCRIPTION/THEORY OF OPERATION  
-----  
MOST TESTS ARE MODULAR, SO THE PROGRAM CAN  
BE STARTED FROM ANY TEST WITHOUT CAUSING ANY  
INITIALIZATION ERRORS. SEE NOTE 2.3.1 ABOVE !!!  
7.1 WHEN 'ESPLEX' IS STARTED FROM CONSOLE OR VIA 'OTOS', IT  
WILL SCAN THE SYSTEM AND WILL PRINT THE SIZE OF THE  
MEMORY. THEN AFTER SETTING UP THE SWITCHES AND PRESSING  
CONTINUE, THE 1ST PASS WILL RUN VERY FAST AS EACH TEST  
IS RUN ONLY ONCE IN THE FIRST PASS. ALL OTHER PASSES  
WILL TAKE MORE TIME AS EACH TEST IS RUN ACCORDING TO THE  
TEST ITERATION COUNT SPECIFIED IN EACH SUBTEST.  
7.2 AFTER THE 1ST PASS, 'ESPCLEY' IS RELOCATED IN AVAILABLE  
LOGICAL MEMORY AND THE AREAS BELOW (CALLED 'LOBUFF') AND  
ABOVE (CALLED 'HIBUFF') OF THE RELOCATED PROGRAM ARE USED  
AS SCRATCH BUFFER AREA. 2 RELOCATED CYCLES ARE RUN  
FOR EACH LOGICAL 32K MODULE.  
ON MAPPED ECLIPSE, 3 CYCLES ARE RUN UNMAPPED AS DESCRIBED  
ABOVE. THEN THE FIRST 32K ARE MAPPED TO ITSELF AND 3 MORE  
CYCLES ARE RUN OUT OF WHICH THE 1ST ONE IS NON-RELOCATED.  
THEN PROGRAM IN THE 1ST 8K IS MOVED TO NEXT 8K AND LOGICAL  
32K ARE MAPPED TO 32K FROM THERE ONWARDS AND 3 CYCLES  
ARE RUN. THIS CONTINUES UNTILL THERE IS AT LEAST 24K LEFT  
ABOVE THE PROGRAM. THEN THE PROGRAM WILL PRINT 'PASS XX'.  
THE ORIGINAL COPY OF THE PROG IS ALWAYS LEFT UNTOUCHED  
IN THE 1ST 8K.  
WHEN THE PROGRAM IS LOADED FROM 'OTOS', 1K OCCUPIED BY  
'OTOS' MONITOR, CAT OR KITTEN IS ALWAYS LEFT UNTOUCHED.  
THE NUMBER OF PASSES EACH TEST IS RUN IN MAPPED MODE  
IS ADJUSTED ACCORDING TO THE SIZE OF THE TOTAL MEMORY SO  
AS TO EQUALISE THE RUN TIME FOR DIFFERENT SIZE SYSTEMS  
8.1 DIAGNOSTIC SUPPORT FEATURES  
-----  
DIAGNOSTIC SUPPORT FEATURES HAVE BEEN ADDED  
TO ASSIST THE USER IN IDENTIFICATION OF THE IMPACT  
OF PROGRAM RELOCATION OR THE EXECUTION IN MAP MODE.  
THE USER MUST MODIFY THE ASSOCIATED CONTROL ENTRIES  
TO ENABLE THEM, BE ADVISED, THE USER MUST RESTORE  
THE PROGRAM TO THE ORIGINAL STATE AND VERIFY NORMAL  
EXECUTION BEFORE ASSUMING THAT THE SYSTEMS CONFIGURA-  
TION IS FUNCTIONALLY CORRECT".  
8.1.1 PROGRAM RELOCATION CHECKSUM  
-----  
PRIOR TO RELOCATION IN NONMAPPED MODE A NEW "COB" CHECK  
WORD IS GENERATED, WHICH, IS VERIFIED, FOLLOWING THE BAM  
XFER EXECUTION. IF THE CHECK WORDS DO NOT COMPARE THE  
PROGRAM HALTS. DUE TO THE NATURE OF THE PROGRAM OVERLAP-  
PING ON RELOCATION AND MODIFYING THE SOURCE BUFFER FROM  
WHICH IT HAS TRANSFERRED THIS TYPE OF ERROR IS UNCOVER-  
ABLE AND THE USER IS ADVISED TO RUN THE BASIC ECLIPSE  
DIAGNOSTICS.  
8.1.2 PROGRAM RELOCATION VERIFICATION

0012 ESPCL

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

URING MAPPED MODE EXECUTION THE SOURCE BUFFER AREA  
IS VERIFIED WORD FOR WORD (EXCEPT LOC. 0 THRU 17 OCTAL)  
AND IF AN ERROR IS DETECTED THE PROGRAM HALTS. THIS  
IS A FATAL CONDITION IN THAT THE PROGRAM SEGMENT THAT IS  
TO BE EXECUTED NEXT MAY BE IN ERROR.  
WITH SLIGHT MODIFICATION (I.E. THE ADDITION OF A HALT AT  
LOCATION "MAPLIT": THE USER MAY RESTART THE FAILING  
PROGRAM FOLLOWING A "XFERRI:" HALT IN BAM ABOVE AT LOC.  
"RETRY". THE OMISSION OF THE HALT ENTRY WILL RESULT IN  
MAP MODE EXECUTION FOLLOWING THE VERIFICATION AND COULD  
MISLEAD THE USER IF FURTHER ERRORS RESULT.  
NOTE:  
ADDRESSES SPECIFIED ABOVE ARE IN RELOCATED MEMORY AREA  
I.E. THE PROGRAM LISTING ADDRESS PLUS THE CONTENTS OF  
"RELOC:" FOR "MAPLIT: AND RETRY:".  
ALSO NOTE THAT THE ABOVE PROCEDURE WILL VERIFY THE  
ABILITY OF THE RLM TO MOVE THE SOURCE CODE CURRENTLY  
RESIDENT TO THE DESTINATION BUFFER SPECIFIED. IF THE  
ADDRESS RANGE SPECIFIED ALLOWED THE ORIGINAL SOURCE  
BUFFER TO OVERLAY THE DESTINATION BUFFER, THE PROGRAM  
WILL HAVE BEEN WIRED OUT ON THE ORIGINAL TRANSFER.  
CAUTION:  
ALWAYS RUN THE BASIC ECLIPSE DIAGNOSTICS FOLLOWING  
PROGRAM CHECKSUM OR VERIFICATION ERRORS.  
INHIBIT MAP EXECUTION  
-----  
LOCATION "DMAP:" MAY BE ALTERED TO ANY NON-ZERO ENTRY  
AND THIS WILL INHIBIT MAP MODE PROGRAM EXECUTION FOR  
THE PURPOSE OF EVALUATING THE OPERATIONAL CAPABILITY OF  
THE PROGRAM WITHOUT THE MAP (MMPU/MMPUI) ENABLED.  
"DMAP" IS LOCATION "376" OCTAL AND MUST BE SET IN NON-  
MAP MODE.  
CAUTION:  
IT IS THE USERS RESPONSABILITY TO RESTORE THE PROGRAM  
TO IT'S ORIGINAL STATE AND VERIFY THE PROPER EXECUTION .  
8.3 LOCK ON FIXED RELOCATION BASE ADDRESS  
-----  
LOCATION "RELWD:" MAY BE ALTERED TO ANY VALUE IN THE  
RANGE OF GREATER THAN 8K (I.E. 20000 OCTAL) AND 8K LESS  
THAN THE CONTENTS OF "MAXLOC:". THIS WILL FIX THE LOGICAL  
ADDRESS OFFSET USED DURING RELOCATION AND EXECUTION OF  
THE PROGRAM, FOR THE PURPOSE OF EVALUATING THE OPERATION  
CAPABILITY OF THE PROGRAM WITHOUT RANDOM RELOCATION, NOTE  
HOWEVER THAT DURING MAP MODE EXECUTION THAT THE PHYSICAL  
ADDRESSES WILL THEN VARY ACCORDING TO AVAILABLE PHYSICAL  
STORAGE.

0013 ESPCL

```

01 CAUTION:
02 -----
03 DO NOT SELECT A VALUE THAT WILL OVERLAY THE "CAT"
04 "KITTEN" DTOS 1K.
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

```

"SELWMD" IS LOCATION "377" OCTAL AND MUST BE SET IN NON-MAP MODE.

FIXED RELOCATION ADDRESS = 0

LOCATION "HELWD:" MAY BE SET EQUAL TO "100000" OCTAL, I.E. BIT <6> = 1. THIS ENABLES RELOCATION XFER EXECUTION TO TAKE PLACE AS ALWAYS, BUT THE PROGRAM IS ALWAYS TRANSFERRED TO LOGICAL LOCATION "0". THIS IS ESPECIALLY USEFUL IN SYSTEMS WHERE IN MAPPED MODE THE PROGRAM FAILS IN RELOCATION AND A SPECIFIC AREA OF PHYSICAL MEMORY IS SUSPECT OF BEING INSTRUCTION EXECUTION OR DATA XFER SENSITIVE. IN MAPPED MODE THE BASIC 8K PROGRAM IS REPOSITIONED UP 8K (PHYSICALLY) AFTER EVERY THIRD EXECUTION CYCLE AND EVENTUALLY RESIDES IN THE SUSPECTED PHYSICAL AREA WHILE THE PROGRAM CODE BASICALLY REFLECTS THE PROGRAM LISTING.

CAUTION

IT IS THE USERS RESPONSIBILITY TO RESTORE THE PROGRAM TO ITS ORIGINAL STATE AND VERIFY PROPER EXECUTION.

INHIBIT ITERATION(S) CONTROL

WHEN PROGRAM EXECUTION IS STARTED AT LOC. 176 OCTAL THE ITERATION CONTROL FLAG IS COMPLETED. I.E. NORMALLY THE PROGRAM WILL EXECUTE WITH ITERATIONS FOLLOWING FIRST PASS EXECUTION (WITHOUT ERRORS). WHEN STARTED AT LOC. 176 THE CONTROL ENTRY IS COMPLETED AND THE FIRST TIME THAT THE PROGRAM IS STARTED AT THAT LOCATION ITERATIONS WILL BE SUPPRESSED IN ANY SUCCESSIVE PASSES AS WELL. NOTE THAT IF THE USER WISHES TO RETURN TO THE NORMAL MODE OF OPERATION HE JUST STARTS AT LOC. 176 OCTAL AGAIN.

RESTRICTION

THE PASS COUNT ENTRY IS NOT ADVANCED IF EITHER ITERATIONS, MAPPED EXECUTION OR RELOCATION CONTROL ARE INVOKED. I.E. END OF PASS WILL BE SIGNIFIED BY THE FOLLOWING OUTPUT:

PASS = 0  
PASS = 0  
ETC.

THIS IS TO ASSURE THAT THE USER WILL KNOW THAT NORMAL PROGRAM EXECUTION HAS BEEN SUSPENDED.

10014 ESPCL

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

```

NEW FEATURES

QUICK VERIFY EXECUTION

FOR LARGE 8/230 OR C/330 (256.K MEMORY) SYSTEMS A METHOD FOR QUICK VERIFICATION OF SYSTEMS INTEGRITY HAS BEEN ADDED. IT'S PRIMARY INTENDED USE IS FOR THE REDUCTION OF EXECUTION TIME FOLLOWING CORRECTIVE MAINTENANCE. IT MAY ALSO BE USED AS A QUICK METHOD OF USER VERIFICATION OF SYSTEMS CAPABILITY PRIOR TO LONG TERM RELIABILITY TESTING. (I.E. OVER NIGHT RUNALL OR CRUNALL EXECUTION UNDER DTOS).

BE SURE TO RETURN THE SMREG SETTING TO NON-QUICK VERIFY MODE USING THE DTOS "SMREG" COMMAND.

RESTRICTION

THIS METHOD OF OPERATION IS "NOT RECOMMENDED" FOR FINAL SYSTEMS ACCEPTANCE, OR IN CASES WHERE FAILURES OCCUR EITHER RANDOMLY OR INFREQUENTLY.

SELECTION OF QV

QUICK VERIFICATION MODE OF OPERATION IS SELECTED UNDER DTOS BY USING THE DTOS "SMREG" COMMAND AND RESPONDING WITH "1(CR)". IT WILL NOT BE SELECTED FOLLOWING THE DTOS "LOAD" COMMAND AND NORMAL SWITCH REGISTER SELECTION.

NOTE:

WHEN SELECTED THE QV OPTION IS ENABLED FOR ANY FUTURE DTOS MONITOR MODE DIAGNOSTIC PROGRAM EXECUTION AND CAN ONLY BE CHANGED WITH A NEW DTOS "SMREG" COMMAND AND THE ABSENCE OF BIT (15).





10019 ESPCL

```

01 .MACRO ELEF1
02 SETUP
03 RAND
04 ANDI
05 MOV
06 ADC
07 XCH
08 STA
09 ELEM
10 SUB#
11 ERROR
12 LOOP
13
14
15

```

```

100.
7777,0
0,1
1+183,1+183
1,1+183
1,1+183
1,0,0
1,1+183,SZR

```

```

;
; INITIALLY,
; AC(*1)=17777 AND
; AC(*1+1)=RANDOM ADDR (=OFFSET)
; AFTER ELEF,AC(*1) MUST BE =
; RANDM ADDR (=OFFSET)

```

\*

10020 ESPCL

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19

```

```

.MACRO ELEF2
SETUP
RAND
ANDI
JSR
ADDI
ADDL#
JMP
AOD
STA
MOV
ADC
ELEM
SUB#
ERROR
LOOP

```

```

100.
7777,0
1,1
9,1+183
0,1,SZR
LFC*2
0,3
0,1+4
3,1+183
1,1
1,0,1
1,1+183,SZR

```

```

;
; AC(*1)=17777
; AFTER EXECUTING ELEF,
; AC(*1) MUST BE EQ. TO
; (RANDM ADDR +PC OF INSTRUCTION
; FOLLOWING ELEF)

```

\*



10021 ESPCL

```

01 .MACRO
02 ELEF3
03 SETUP
04 RNDADR
05 ONE, TSTTOP
06 *1+183, *1+183 ; ALL AC'S ARE SET TO 177777
07 *1+183, *1+283 ;
08 *1+283, *1+383 ; AND CARRY IS INITIALISED
09 *1, *3 ; BY ADC*2
10 *1, *1 ;
11 *1, 0, 0 ; *4 MUST NOT CHANGE ANY AC'S
12 *1, *1, *3 ; FOR STATE OF CARRY EXCEPT AC*1
13
14 *1+183, *1+183, SZR ;
15 ERROR
16 *1+283, *1+283, SZR ;
17 ERROR
18 *1+383, *1+383, SZR ;
19 ERROR
20 LOOP
21

```

z

10022 ESPCL

```

01 .MACRO
02 ELEF4
03 SETUP
04 RND
05 100.
06 77777, 0
07 *1, *1 ;
08 *26, *3 ;**
09 *23, *5 ;**
10 IORI 100000, 3
11 *1, *3 ;
12 *1, *3 ;
13 STA 3, LEFLC
14 *1, *2 ;
15 ANDI 177, 0 ;
16 MOV 0, *2 ; THIS SETS UP FOR INDEX
17 *1, LEFLC, 0 ; MODE *2 FOR MULTI-INDIRECT
18 *1, *1 ; TEST
19 IORI 100000, 1 ; ADD THE INDIRECT BIT
20 *1, *2 ;
21 *1, *2 ;
22 LDA 1, LFG*1+1 ;
23 *1, *1, SZR ;
24 LOOP
25 *1, *3 ;
26 *1, *3 ;
27 *1, *3 ;
28 *1, *3 ;

```

z

10023 ESPCL

```

01 .MACRO ELD1
02 ELEF5 ^1,^2,^3,^4
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

```

10024 ESPCL

```

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

```

10024 ESPCL

```

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

```

10024 ESPCL

```

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

```

```

10025 ESPCL
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
.MACRO ESTA2
SETUP 100.
STA 0,STLCO
RAND 0,1
MOV 0,STLCO
LDA 0,STLCO
MOVZ 0,1,SZC
HLV 0,1
STA 0,1
EST 1,0,1
LDA 0,STLCO
SUB# 0,1,SZR
ERROR
LOOP
X
;C(STLCO)=RNDM ADDR
;ACI= RNDM DATA
;AC1=(RNDM ADDR/2)+BIT 15
;AC0=RNDM ADDR/2
;RNDM ADDR MUST CONTAIN
;RANDOM DATA(=CIAC1))

10026 ESPCL
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
.MACRO ESTA3
SETUP 100.
STA 0,STLCO
LDA 1,STLCO
SUB# 1,0,SZR
JMP -3
STA 0,STLCO
RAND 0,1
LDA 0,STLCO
X
.MACRO
ESTA4
ESTA3
MOVZ 0,4,SZC
ADI 1,4
HLV 0
IORI 100000,0
STA 0,3
STA 0,4
EST 1,80,4
ELDA 0,80,4
SUM# 1,0,SZR
ERROR
LOOP
X
;C(RNDM ADDR 1)=(RNDM ADDR 2)
;MAKE SURE THAT RNDM NOS.
;FROM ^2 ARE DIFFERENT
;
;
;ACI=RNDM DATA
;AC0=RNDM ADDR 1
;AC^4 IS SET EQ TO
;(RNDM ADDR 1)/2+BIT 15
;AND (RNDM ADDR 1)/2 IS
;STORED IN ESTA AND
;ELDA INSTRUCTIONS
;
;AC0 MUST BE EQ. TO
;(RNDM ADDR 2)=RNDM DATA

```

```

1:0027 ESPCL
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
- MACRO ESTAS
  SETUP 100.
  RAND
  STA
  LOBFAD
  STA
  LOBFAD
  LDA
  SUB#
  JMP
  ADDOR
  STA
  HIBFAD
  MOV
  LDA
  STA
  ADDRZ
  LDA
  STA
  MOVZR
  ADI
  HCV
  ADDOR
  STA
  LDA
  ESTA
  LOA
  SUB#
  ERROR
  LOOP
  ~2:
    :SAVE RANDOM DATA
    :C(STLC0)=(RNDM ADDR 1)
    :MAKE SURE RNDM ADDRESSES
    :FROM LOBUFF ARE DIFFERENT
    :
    :C(STLC1)=@(RNDM ADDR 2)
    ;
    :AC0=AC2=(RNDM ADDR 3)
    :STORE (@RNDM ADDR 2) IN
    : (RNDM ADDR 3) AND STORE
    : (@RNDM ADDR 1) IN
    : (RNDM ADDR 2)
    ;
    :ALSO STORE (RNDM ADDR 3)/2+
    :BIT 15 IN AC*1 AND
    :STORE (RNDM ADDR 3)/2 IN
    :ESTA INSTRUCTION
    ;
    :AC0=AC1=RANDOM DATA
    :ESTA MUST STORE
    :RANDOM DATA IN
    :RNDM ADDR 1
    ;
  ~3:
    XERR
    SETUP
    FILL
    ~2
    MOV
    LDA
    STA
    ELEF
    STA
    ~1
    ERROR
    LOOP
  X
1:0028 ESPCL
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
- MACRO EJMRF1
  FILL
  SETUP
  ~2
  MOV
  LDA
  STA
  ELEF
  STA
  ~1
  ERROR
  LOOP
  ~3:
    :AC0=AC2=RANDOM ADDR
    : (JMP @JMLC1) IS STORED IN
    :RANDOM ADDRESS.
    :ADDR OF (~1+3) IS
    :STORED IN JMLC1 FOR
    :PROPER RETURN
    ;
  X

```

10029 ESPCL

.MACRO EIMP2

```

01 01
02 02
03 03
04 04
05 05
06 06
07 07
08 08
09 09
10 10
11 11
12 12
13 13
14 14
15 15
16 16
17 17
18 18
19 19

```

```

FILL          XERR
SETUP        100.
MOV          0.3
LDA          1.0,3
STA          1.2, +2.1
ELEF        1.1, JMLC1
STA          1.2, .1
ELEF        1.0
SUB          0.1, +2
STA          -1
ERROR
LOOP

```

```

;AC0=AC3=RANDOM ADDRESS
;STORE (JMP @JMLC1) IN
;RNDM ADDRESS
;ADDR OF (*1+3) IS STORED
;IN JMLC1 FOR PROPER RETURN
;(RNDM ADDR=PC OF *-1+1)
;IS STORED IN *-1
;INSTRUCTION
;

```

-2:

x

10030 ESPCL

.MACRO EJMP3

```

01 01
02 02
03 03
04 04
05 05
06 06
07 07
08 08
09 09
10 10
11 11
12 12
13 13
14 14
15 15
16 16
17 17
18 18
19 19

```

```

FILL          XERR
SETUP        100.
MOV          0.2
LDA          1.0,2
STA          1.3, +2.1
ELEF        1.1, JMLC1
STA          0.4, SEC
MOVZX       1.4
ADI         0
HLV         0.1, +2
STA         0.4
ERROR
LOOP

```

```

;AC0=AC2=RNDM ADDRESS
;(JMP @JMLC1) IS STORED
;IN RANDOM ADDRESS
;ADDR OF (*1+3) IS STORED
;IN JMLC1 FOR GOOD RETURN
;(RNDM ADDR/2)+8BIT IS IS
;STORED IN AC+4 AND
;(RNDM ADDR/2) IS STORED
;IN *-1 INSTRUCTION
;

```

-3:

x

10031 ESPCL

```

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

.MACRO EJMP4
FILL
SETUP
*2
STA 0,JMLC2
*4
LDA 1,JMLC2
SUB# 1,0,SNR
*6
JMP 0,@JMLC2
MOV 0,2JMLC2
LDA 1,JMLC0
STA 1,0,2
ELEF 1,*3,+2,1
STA 1,JMLC1
LDA 0,JMLC2
*NOLOC
*IFE (JMSWC-1)
*ELEF 1,*3,+1
*SUB# 1,0
*ENDC
*IFE (JMSWC-2)&(JMSWC-3)
*MOVZH 0,*5,SZC
*ADI 1,*5
*HLV 0
*NOLOC
*IORI 100000,0
*STA 0,*+2
*ERROR
*LOOP
*

```

10032 ESPCL

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

.MACRO EJMP5
FILL
SETUP
*2
STA 0,JMLC2
*4
LDA 1,JMLC2
SUB# 1,0,SNR
*6
JMP 0,@JMLC2
MOV 0,2JMLC2
LDA 1,JMLC0
STA 1,0,2
ELEF 1,*3,+2,1
STA 1,JMLC1
LDA 0,JMLC2
*NOLOC
*IFE (JMSWC-1)
*ELEF 1,*3,+1
*SUB# 1,0
*ENDC
*IFE (JMSWC-2)&(JMSWC-3)
*MOVZR 0,*5,SZC
*ADI 1,*5
*HLV 0
*NOLOC
*IORI 100000,0
*STA 0,*+2
*ERROR
*LOOP
*

```

10033 ESPCL

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

.MACRO EJMP6
FILL
SETUP
*2
STA 0,JMLC2
*4
LDA 1,JMLC2
SUB# 1,0,SNR
*6
JMP 0,@JMLC2
MOV 0,2JMLC2
LDA 1,JMLC0
STA 1,0,2
ELEF 1,*3,+2,1
STA 1,JMLC1
LDA 0,JMLC2
*NOLOC
*IFE (JMSWC-1)
*ELEF 1,*3,+1
*SUB# 1,0
*ENDC
*IFE (JMSWC-2)&(JMSWC-3)
*MOVZR 0,*5,SZC
*ADI 1,*5
*HLV 0
*NOLOC
*IORI 100000,0
*STA 0,*+2
*ERROR
*LOOP
*

```

10033 ESPCL

```
01 .MACRO EISZ1
02 SETUP 100.
03 STA 0,ISLCO
04 STA 0,+B.
05 SUB 0,0
06 STA 0,@ISLCO
07 MOV 0,3
08 NEG 0,1
09 MOV*2 1,2
10 MOV*1 0,0
11 SUB# 0,3,SZR
12 ERROR 1,2,SZR
13 ERROR
14 MOV 0,0,*3
15 LOOP
16
17
18
19
20
21
22
23
```

10034 ESPCL

```
01 .MACRO EISZ2
02 SETUP 100.
03 RAND 0,ISLCO
04 STA 0,ISLCO
05 LDA 1,ISLCO
06 STA 1,@ISLCO
07 *MULOC
08 *IFE (ISZSW-1)
09 *ELEF 1,*3-*,1
10 *SUB 1,0
11 *ENDC
12 *IFE (ISZSW-2)&(ISZSW-3)
13 *MOVZR 0,*4,SZC
14 *ADI 1,*4
15 *HLV 0
16 *ENDC
17 *MULOC
18 *STA 0,+*2
19 *MOV 0,0
20 *LDA 0,ISLCO
21 *LDA 1,@ISLCO
22 *SUB#
23 *ERROR
24 *LOOP
25
26
27
28
29
30
31
32
```

```

01 :C(ISLCO)=RANDOM DATA
02 :C(ISLCO)=RANDOM ADDRESS
03 :STORE RANDOM DATA IN
04 :RANDOM ADDRESS
05
06 :C(ISLCO)=RANDOM DATA
07 :STORE RANDOM DATA IN
08 :RANDOM ADDRESS
09
10 :C(ISLCO)=RANDOM DATA
11 :STORE RANDOM DATA IN
12 :RANDOM ADDRESS
13
14 :RANDOM ADDR=PC OF ^1+1
15 :IS STORED IN ^1
16
17 (ISZSW-2)&(ISZSW-3)
18 :C(RNDM ADDR/2)+RIT 15
19 :IS STORED IN AC*4 AND
20 :C(RNDM ADDR/2) IN ^1
21
22 *EXECUTE ^1 INSTRUCTION
23 :C(RNDM ADDR) MUST BE
24 :C(RNDM DATA+1) FOR EISZ
25 :AND =(RNDM DATA-1) FOR
26 :EDSZ INSTRUCTION
27
28
29
30
31
32
```

10035 ESPCL

```
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

:MACRO EISZ3
    SETUP EISZ3
    RAND
    STA
    LOBFAD
    STA
    LDA
    STA
    HIBFAD
    MOV
    STA
    .NOLC
    .IFE
    ELEF
    SUB
    .ENDC
    .IFE
    MOVZR
    ADI
    HLV
    .NOLC
    .NOLC
    ADDR
    STA
    MOV
    LDA
    LDA
    SUB#
    ERROR
    LOOP
    *
;C(ISLC0)=RANDOM DATA
;STORE RANDOM DATA INTO
;RANDOM ADDRESS 1
;STORE RANDOM ADDRESS 1 INTO
;RANDOM ADDRESS 2
;(RNDM ADDR 2-PC OF ^1+1)
;IS IN AC0
;(ISZSW-2)&(ISZSW-3)
;IS STORED IN AC^3 AND
;ACO=(RNDM ADDR 2)/2
;ADD INDIRECT BIT TO ACO
;AND STORE IN ^1
;EXECUTE ^1 INSTRUCTION
;IN CASE IF ^1 SKIPS
;C(RNDM ADDR 1) MUST BE
;C(RNDM DATA+1) FOR EISZ AND
;C(RNDM DATA-1) FOR EDSZ
;C(RNDM ADDR 1) MUST BE
;C(RNDM DATA+1) FOR EISZ
;AND =(RNDM DATA-1) FOR EDSZ
;
```

```
10036 ESPCL
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

:MACRO EISZ4
    SETUP EISZ4
    RAND
    STA
    LOBFAD
    STA
    LDA
    STA
    HIBFAD
    MOV
    STA
    .NOLC
    .IFE
    ELEF
    SUB
    .ENDC
    .IFE
    MOVZR
    ADI
    HLV
    .NOLC
    .NOLC
    ADDR
    STA
    MOV
    LDA
    LDA
    SUB#
    ERROR
    LOOP
    *
;C(ISLC0)=RANDOM DATA
;C(ISLC1)=RNDM ADDR 1
;STORE RNDM DATA INTO
;RNDM ADDR 1
;C(ISLC2)=RNDM ADDR 2
;STORE RNDM ADDR 1 IN
;RNDM ADDR 2
;MAKE SURE THAT RNDM ADDR
;2 AND 3 ARE DIFFERENT
;AND SAVE RNDM ADDR 2
;IN RNDM ADDR 3
;ACO=(RNDM ADDR 3-PC OF
;^1+1)
;(ISZSW-2)&(ISZSW-3)
;(RNDM ADDR 3/2)*BIT 15 IS
;STORED IN AC^3 AND
;ACO=(RNDM ADDR 3/2)
;ADD INDIRECT BIT TO ACO
;AND STORE IN ^1
;EXECUTE ^1 INSTRUCTION
;C(RNDM ADDR 1) MUST BE
;C(RNDM DATA+1) FOR EISZ
;AND =(RNDM DATA-1) FOR EDSZ
;
```





10039 ESPCL

```
01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
*MACRO EDSP3  
DSSWI=0  
EDSP1 1,2,3  
EDSP2 4,5,6  
*  
*MACRO EDSP4  
DSSWI=1  
EDSP1 1,2,3  
EDSP2 4,5,6,7  
*
```

10040 ESPCL

```
01  
02  
03  
04  
05  
*MACRO FILL  
JSR 1  
*  
*FILL UPPER AND LOWER  
*SCRATCH BUFFER AREA  
*WITH (*)
```

01  
02  
03  
04  
05  
\*FILL

10041 ESPCL

```

01 .MACRO LOTBAD
02 LDA 0,LOWRF
03 ADI 2,0
04 STA 0,LDSTB
05 RNDADR LDSTB,LOBFU
06
07
08
09 .MACRO HITBAD
10 LDA 0,HIGBF
11 ADI 2,0
12 STA 0,HSTB
13 RNDADR HSTB,HIBFU
14

```

10042 ESPCL

```

01 .MACRO INTKU
02 ;
03 ; INTRP, INTERRUPT HANDLER
04 ;
05
06 CPU CPU
07 PFAIL *(YES), ?IS IT POWER FAIL?
08 MOVH 0,0 ?SAVE THE STATE OF THE MACHINE
09 PSH 0,0 ?
10 PSH 0,0 ?GET PROG INTERRUPT LOCATION
11 STA 0,WHERE ?*RECORD
12 INTA 0 ?GET THE DEVICE INTERRUPTING
13 LDA 0,ERADDR*1 ?**((2))INTRP-1 IS IT ERCC (2)?
14 SUB# 0,1,SZR ?YES, GO TO REPORT
15 JMP INTRBAD ?NO, ILLEGAL INTERRUPT
16 DIA 1,ERCC ?GET FAULT ADDR(LOW 16 BITS)
17 STA 1,MFAD ?AND SAVE IT
18 DIG 1,ERCC ?GET THE FAULT CODE
19 STA 1,MFAD1 ?AND HIGH ORDER ADDR BIT
20 MOVZR 1,1 ?
21 LDA 0,C366 ?
22 LSH 0,1 ?
23 STA 1,MFCD ?SAVE THE FAULT CODE
24 CRLFPR
25 PRINT MSI
26 CRLFPR
27 LDA 0,MFAD1
28 ANDI 1,0 ?GET HIGH ORDER ADDR BIT
29 LDA 1,MFAD ?GET LOW ORDER 16 ADDR BITS
30 LDA 2,C2000 ?CONVERT TO MODULE AND ADDR
31 DIV
32 STA 0,ERADDR ?SAVE ADDR
33 STA 1,MODULE ?SAVE MODULE
34 PRINT MS8
35 DECPR MODULE
36 PRINT MS3
37 OCTPR ERADDR
38 CRLFPR
39 PRINT MS4
40 OCTPR MFCD
41 PRINT MS7
42 OCTPR 0
43 CRLFPR
44 EJSR MAPOTA ?PRINT MAP DATA
45 LDA 0,03 ?TURN ON ERR CORR
46 DGAS 0,ERCC ?
47 JMP *+1 ?
48 LDA 0,DSPER ?*LINK FATAL
49 STA 0,0 ?*LOC. 0 ERROR CALL
50 DIA 0,MMPU ?
51 MOVH# 0,0,SZC ?
52 OUA 0,MMPU ?
53 POP 0,0 ?
54 MOVL 0,0 ?
55 POP 3,0 ?
56 INTEN
57 JMP @0 ?RESTORE THE STATE OF MACHINE
58 ***WHERE: 0 ?*RETURN TO PROGRAM
59 MODULE: 0 ?*RECORDED INTERRUPT LOC.
60

```

```

01 ERADDR: 0
02 2.
03 **ERCC
04 0,+3 ?*SETUP RESTART LOC. 0
05 0.0 ?*HERE
06 HALT ?*WAIT FOR POWER
07 SPFLPR ?*I.E. FLRTN BELOW
08 MSPFL ?*PRINT POWER FAILED
09 @BGNADR ?*ATTEMPT PROG RESTART
10 JMP
11
12 %
13 *MACRO PAGE 0
14 **WOLOC 0
15 :**ESPCLEX PAGE ZERO ENTRIES
16 ***
17 CALL: 0
18 CAL2: 0
19 LEFLOC: 0
20 LFV0: 0
21 LFSV0: 0
22 LDALC: 0
23 LDSV0: 0
24 LDSV1: 0
25 STLCO: 0
26 STLC1: 0
27 STSV0: 0
28 D100: 100.
29 JMLCO: JMP
30 JMLC1: 0
31 JMLC2: 0
32 JMLC3: 0
33 JMLC4: XOP 0,0,0
34
35 ISLC0: 0
36 ISLC1: 0
37 ISLC2: 0
38 DSLCO: 0
39 DSLC1: 0
40 DSSV0: 0
41 DSSV1: 0
42 DSSV2: 0
43 LDS1B: 0
44 HDST1B: 0
45 ONE: 1
46 60: TEM*20
47 TEM: 0
48 *KTBL: 0
49 T*LSIZ: 2.
50 02: 2.
51 03: 3.
52 ?
53 ?
54 ? PAGE 0 LOCATIONS FOR ERCC
55 ?
56 MFAD: 0
57 MFAD1: 0
58 MFCD: 0
59 C366: 366
60 D*SPER: XOP 0,0,0 ?*XERR ERROR CALL XOP WHICH REPLACES

```

```

?CALL ROUTINE TEMPORARY
?CALL ROUTINE TEMPORARY
?USED IN ELEM TESTS
?SAVE LOC FOR ELEF TESTS
?TEMP LOC FOR ELDA TESTS
?SAVE LOC FOR ELDA TESTS
?SAVE LOC FOR ELDA TESTS
?USED IN ESTA TESTS
?USED IN ESTA TESTS
?CONSTANT
?USED IN EJMP AND EJSR TESTS
?USED IN EJMP AND EJSR TESTS
?USED IN EJMP AND EJSR TESTS
?USED IN EJMP AND EJSR TESTS
**ERROR CALL
**USED IN EJMP AND EJSR TESTS
?USED IN EISZ AND EDSZ TESTS
?USED IN EISZ AND EDSZ TESTS
?USED IN EISZ AND EDSZ TESTS
?USED IN DSPA TESTS
?USED IN DSPA TESTS
?USED IN DSPA TESTS
?USED IN DSPA TESTS
?USED IN DSPA TESTS
?USED IN DSPA TEST
?CONSTANT
?CONSTANT
?TEMP LOCATION
?USED IN DSPA TESTS
?USED IN DSPA TESTS
?CONSTANT
?CONSTANT

```

```

?DIA ERCC
?DIB ERCC
?ERCC FAULT CODE
?CONSTANT
?CONSTANT
?CONSTANT

```

```

10045 ESPCL
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

: DEFINITIONS BLOCK
:
: SPECIAL INSTRUCTION DEFINITIONS
.DERA ELOB=102170
.DERA ESTB=122170
.OIAC LDI=103650
.OIAC STI=123650
:
: XOP ERROR CALL (XERR)
:
.DUSR XERR=100030
.DUSR LDI=143650
.DUSR STI=147650
.DUSR CMV=154650
.DUSR CMP=157650
.DUSR CTR=163650
.DUSR CMI=167650
.DUSR EDI=173650
.DUSR LSN=177650
.DUSR SCL=127510
.OIAC FINT=143150

: EDIT CODES
000000 .DUSR DEMD=0
000001 .DUSR DMDF=1
000002 .DUSR DSTK=2
000003 .DUSR DOTK=3
000004 .DUSR DSSZ=4
000005 .DUSR DSSO=5
000006 .DUSR DSTZ=6
000007 .DUSR DMVQ=7
000010 .DUSR DMVN=10
000011 .DUSR DSTO=11
000012 .DUSR DINT=12
000013 .DUSR DAPT=13
000014 .DUSR DMVC=14
000015 .DUSR DMVA=15
000016 .DUSR DINS=16
000017 .DUSR DAPS=17
000020 .DUSR DINC=20
000021 .DUSR DICI=21
000022 .DUSR DADI=22
000023 .DUSR DASI=23
000024 .DUSR DMVF=24
000025 .DUSR DMVC=25
000026 .DUSR DMVS=26
000027 .DUSR DAPU=27

: ASSORTED INSTRUCTION DEFINITIONS
.DUSR EHMT=MALT
.DUSR RHOP=1000
.DUSR HOP=400
.DUSR NOP=401
.DUSR SKIP=402

: DEVICE CODE DEFINITIONS
000002 .DUSR ERCC=2
000003 .DUSR MNPUS=3

: HARDWARE RESERVED LOCATIONS
000000 IR=0 **I/O RETURN
000001 IME1 **INTERUPT HANDLER
000002 SC=2 **SCL HANDLER
000003 PF=3 **PROTECTION FAULT HANDLER
000004 VS=4 **VCT STACK
000005 CM=5 **CURRENT PRIORITY MASK
000006 VL=6 **VCT LIMIT
000007 VF=7 **VCT FAULT HANDLER

000040 SP=40 **STACK POINTER
000041 FP=41 **FRAME POINTER
000042 SL=42 **STACK LIMIT
000043 SF=43 **STACK FAULT
000044 LX=44 **XOP ORIGIN
000045 FF=45 **FLT POINT FAULT
000046 CF=46 **COMMERCIAL FAULT

: MISCELLANEOUS
:DSPER=XUP 0,0,0
XORC=LX
COMP=CF
000044
000046

10046 ESPCL
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

: ASSORTED INSTRUCTION DEFINITIONS
.DUSR EHMT=MALT
.DUSR RHOP=1000
.DUSR HOP=400
.DUSR NOP=401
.DUSR SKIP=402

: DEVICE CODE DEFINITIONS
000002 .DUSR ERCC=2
000003 .DUSR MNPUS=3

: HARDWARE RESERVED LOCATIONS
000000 IR=0 **I/O RETURN
000001 IME1 **INTERUPT HANDLER
000002 SC=2 **SCL HANDLER
000003 PF=3 **PROTECTION FAULT HANDLER
000004 VS=4 **VCT STACK
000005 CM=5 **CURRENT PRIORITY MASK
000006 VL=6 **VCT LIMIT
000007 VF=7 **VCT FAULT HANDLER

000040 SP=40 **STACK POINTER
000041 FP=41 **FRAME POINTER
000042 SL=42 **STACK LIMIT
000043 SF=43 **STACK FAULT
000044 LX=44 **XOP ORIGIN
000045 FF=45 **FLT POINT FAULT
000046 CF=46 **COMMERCIAL FAULT

: MISCELLANEOUS
:DSPER=XUP 0,0,0
XORC=LX
COMP=CF
000044
000046

:**ERROR LINKAGE FILLER WORD
:**(XERR) REPLACES JSR CENTER CALL

```

```

0047 ESPCL
01 00042 001132
02 00043 000163 SFLIP: FATAL
03 00044 014263 LXOP: DMIGIN
04 00045 000045 .LOC 45
05 00046 015326 EGSS :POINTER TO EGGS BLOCK
06 00047 015326 .ZHEL
07 00048 000000-000010
08 00049 000060
09 00050 000060
10 00051 000060
11 00052 000060
12 00053 000060
13 00054 000060
14 00055 000060
15 00056 000060
16 00057 000060
17 00058 000060
18 00059 000060
19 00060 013425 ISTKI: STKI
20 00061 012344 ICAL: CAL
21 00062 000000 NEMTOP: 0
22 00063 000000 ISTIOP: 0
23 00064 012524 IMAPIT: MAPIT
24 00065 123456 RAM: 123456
25 00066 012317 RNDTA: RNDOT
26 00067 000500 BUFP: LORUFF

```

```

0048 ESPCL
01 00042 001132
02 00043 000163 SFLIP: FATAL
03 00044 014263 LXOP: DMIGIN
04 00045 000045 .LOC 45
05 00046 015326 EGSS :POINTER TO EGGS BLOCK
06 00047 015326 .ZHEL
07 00048 000000-000010
08 00049 000060
09 00050 000060
10 00051 000060
11 00052 000060
12 00053 000060
13 00054 000060
14 00055 000060
15 00056 000060
16 00057 000060
17 00058 000060
18 00059 000060
19 00060 013425 ISTKI: STKI
20 00061 012344 ICAL: CAL
21 00062 000000 NEMTOP: 0
22 00063 000000 ISTIOP: 0
23 00064 012524 IMAPIT: MAPIT
24 00065 123456 RAM: 123456
25 00066 012317 RNDTA: RNDOT
26 00067 000500 BUFP: LORUFF

```

```

0049 ESPCL
01 00042 001132
02 00043 000163 SFLIP: FATAL
03 00044 014263 LXOP: DMIGIN
04 00045 000045 .LOC 45
05 00046 015326 EGSS :POINTER TO EGGS BLOCK
06 00047 015326 .ZHEL
07 00048 000000-000010
08 00049 000060
09 00050 000060
10 00051 000060
11 00052 000060
12 00053 000060
13 00054 000060
14 00055 000060
15 00056 000060
16 00057 000060
17 00058 000060
18 00059 000060
19 00060 013425 ISTKI: STKI
20 00061 012344 ICAL: CAL
21 00062 000000 NEMTOP: 0
22 00063 000000 ISTIOP: 0
23 00064 012524 IMAPIT: MAPIT
24 00065 123456 RAM: 123456
25 00066 012317 RNDTA: RNDOT
26 00067 000500 BUFP: LORUFF

```

```

:NOTE:
:THE FOLLOWING "VCT" INSTRUCTION ASSOCIATED ENTRIES
:ARE USED FOR SCRATCH AREA DURING RELOCATION AND
:MAPPED MODE EXECUTION. IT IS THE USERS RESPONSIBILITY
:TO SETUP THESE LOCATIONS PRIOR TO HIS EXECUTION OF
:THE "VCT" INSTRUCTION.
:
23 00004 077077 CKMRD: HALTA 3 ;
24 00005 077077 XAC1: HALTA 3 ;
25 00006 077077 XAC2: HALTA 3 ;
26 00007 077077 XAC3: HALTA 3 ;
27 00010 077077 COUNT: HALTA 3 ;VERIFICATION SUCCESSFUL TALLY
29 00011 077077 HAF001: HALTA 3 ;***** FATAL ERROR HALT *****
30 00012 077077 HAF002: HALTA 3 ;***** FATAL ERROR HALT *****
31 00013 077077 HAF003: HALTA 3 ;***** FATAL ERROR HALT *****
32 00014 077077 HAF004: HALTA 3 ;***** FATAL ERROR HALT *****
33 00015 077077 HAF005: HALTA 3 ;***** FATAL ERROR HALT *****
34 00016 077077 HAF006: HALTA 3 ;***** FATAL ERROR HALT *****
35 00017 077077 HAF007: HALTA 3 ;***** FATAL ERROR HALT *****
36
37 :NOTE
38 :AUTO INC/DEC LOCATIONS BELOW ARE INITIALIZED
39 :TO HALT ON ILLEGAL ENTRY.
40
41 00020 077077 HAF010: HALTA 3 ;***** FATAL ERROR HALT *****
42 00021 077077 HAF011: HALTA 3 ;***** FATAL ERROR HALT *****
43 00022 077077 HAF012: HALTA 3 ;***** FATAL ERROR HALT *****
44 00023 077077 HAF013: HALTA 3 ;***** FATAL ERROR HALT *****
45 00024 077077 HAF014: HALTA 3 ;***** FATAL ERROR HALT *****
46 00025 077077 HAF015: HALTA 3 ;***** FATAL ERROR HALT *****
47 00026 077077 HAF016: HALTA 3 ;***** FATAL ERROR HALT *****
48 00027 077077 HAF017: HALTA 3 ;***** FATAL ERROR HALT *****
49 00030 077077 HAF020: HALTA 3 ;***** FATAL ERROR HALT *****
50 00031 077077 HAF021: HALTA 3 ;***** FATAL ERROR HALT *****
51 00032 077077 HAF022: HALTA 3 ;***** FATAL ERROR HALT *****
52 00033 077077 HAF023: HALTA 3 ;***** FATAL ERROR HALT *****
53 00034 077077 HAF024: HALTA 3 ;***** FATAL ERROR HALT *****
54 00035 077077 HAF025: HALTA 3 ;***** FATAL ERROR HALT *****
55 00036 077077 HAF026: HALTA 3 ;***** FATAL ERROR HALT *****
56 00037 077077 HAF027: HALTA 3 ;***** FATAL ERROR HALT *****
57
58 00040 000040
59 00041 001010
60 00042 001010

```

```

:LOC 40 :40->44 RESERVED FOR STACK CONTROL
:STACK
:
58 00040 000040
59 00041 001010
60 00042 001010

```

```

:LOC 40 :40->44 RESERVED FOR STACK CONTROL
:STACK
:
58 00040 000040
59 00041 001010
60 00042 001010

```

```

0049 ESPCL
01 00070 113710 BAMBY:      BAM      ;MOVE PROGRAM...
02 00071 102470 EJM      ;VERIFY ;CHECK REFERED OK
03      012403      ;
04      ;
05      ;
06      ;
07      ;
08      ;
09 00073 020214 LDA 0,RELOC ;INITIALIZE POINTERS
10 00074 024134 LDA 1,BLKZ
11 00075 172470 ELEM 2,BAMBLK
12 00077 176470 ELEM 3,BAMBLK+(CENTLO-BAMBLK)
13 00107 176470 CALL ;ADJUST CALL PARAMETERS
14 00101 113710 CALL ;EXIT TO PROGRAM
15 00102 006061 JSR @ICAL ;CALL SUBROUTINE
16      ;
17      ;
18 00103 001471 BEGIN ;"LOOP" ENTRY
19 00104 011724 BAMBLK: LOP
20 00105 012014 INIT ;"SETUP"
21      ;
22 00106 012255 .RAND ;"ERROR"
23 00107 012272 RNDMAD ;GETS RNDM ADDR
24 00110 001010 LOBUFF*200.
25 00111 015353 JIBUFF ;BOTTOM OF UPPER BUFFER
26 00112 013175 JMFIL ;FILLS SCRATCH BUFFER WITH
27      ;TRAILING ARGUMENT SPECIFIED
28      ;
29 00113 013212 JMFIL
30      ;
31 00114 001010 STAKB
32 00115 013442 STAKF
33 00116 000000 0
34      ;
35 00117 011724 EMILO: LOP
36 00120 012014 ENTIN: INIT
37      ;
38 00121 012255 ENTER: ERR
39      ;
40 00122 012272 ENTRA: RAND
41 00123 012272 RNDMAD ;RANDOM
42 00124 015353 LOBFU: LOBUFF*200.
43 00125 013175 HIGB: HIBUFF ;SCRATCH BUFFERS FILL SUBR.
44 00126 013212 JFILL: JMFIL
45 00127 013212 HFILL: JMFIL
46 00128 010100 STKB: STAKB
47 00129 013442 STKF: STAKF
48 00130 013442 STKF: 0
49 00131 000000 COMRF: CMCLF
50      ;
51 00132 000500 LOWBF: LOBUFF ;INITIALLY BOTTOM OF LOWER BUFFER
52 00133 015663 HIBFU: HIBUFF*200. ;INITIALLY TOP OF UPPER BUFFER
53      ;
54 00134 000000 .NOLOC 0
55 00135 000013 BLKZ: BLKX
56 00136 177777 XOPFLG: -1

```

```

0050 ESPCL
01 00136 000011 C11: 11
02 00137 000060 C60: 60
03 00140 000144 C144: 144
04 00141 000201 C201: 201
05 00142 000240 C240: 240
06 00143 000377 C377: 377
07 00144 000400 C400: 400
08 00145 002000 C2000: 2000
09 00146 004000 E4000: 11*400
10 00147 106400 E10640: 215*400
11      ;LOCATIONS FOR MAP .....
12 00150 000000 MPNSW: 0 ;CURRENT RUN MODE SWITCH
13 00151 000000 MAPSW: 0 ;MAP YES/NO SWITCH
14 00152 000000 HIG1K: 0 ;HIGHEST PHYSICAL (MOD1K)
15 00153 000000 DTOS1K: 0 ;DTOS 1K BLOCK
16 00154 177770 MPBGN: -8. ;BEGINNING OF MAP ((MOD1K)
17 00155 000000 MPEND: 0 ;END OF MAP (MOD1K)
18 00156 177770 MNSB: -4. ;CONSTANT
19 00157 000000 USRMAP: 0 ;CURRENT USER MAP
20      ;
21      ; PAGE 0 LOCATION FOR POWER FAIL
22      ;
23 00160 001464 PFLPR: PFLRTN ;RETURN AFTER POWER FAIL
24      ;
25 00161 000000 MAPF: 0 ;MMPU/MMPUI MAP FLAG 0/1
26 00162 000000 MMPUI: 0 ;MAP STATUS WORD
27      ;
28      ;
29      ;
30      ;
31 00163 077077 HAF030: HALTA 3 ;STACK FAULT (FATAL ERROR)
32      ;
33      ;
34 00164 077077 HF031: HALTA 3 ;***** FATAL ERROR HALT *****
35      ;
36      ;
37      ;
38      ;
39      ;
40 00165 015326 .EGGS: EGGS ;XOP ORIGIN FAULT (FATAL ERROR)
;***** FATAL ERROR HALT *****
;FATAL SUGGESTS STACK OVER/UNDER FLOW HAS OCCURRED
;FALSE SUGGESTS "XOP" ILLEGAL LINKAGE VIA ORIGIN TABLE HAS OCCURRED

```

0051 ESPCL  
01 : ENTRY POINTS FOR CAT/NO-CAT RESTARTS  
02 LDC 170 :START W/D CAT  
03 SUBO 0,0,SKP :START WITH CAT  
04 00171 102000 MERIT: ADC 0,0  
05 00172 142470 ESTA 0,CATSW  
06 015135  
07 00174 002175 ROTOS: JMP 2,\*1  
08 00175 001203 : SPACE RESERVED FOR SPECIAL PURPOSE ENTRIES  
09 LDC 176 :  
10 EJSR FIXIT ;COMPLEMENT ITERATION CONTROL  
11 000176  
12 106670  
13 013741  
14 :THIS FEATURE MAY BE USED WITH CAT/NO-CAT EXECUTION  
15 :BY INSERTING A "NOP" = 401 IN LOCATION "CFXIT":  
16 :AND STARTING AT LOCATION 176 OCTAL ABOVE.  
17 :I.E. DISABLE/ENABLE ITERATIONS EACH SUCCESSIVE  
18 :START AT LOC. 176  
19 : LOCATIONS 200 - 215 RESERVED FOR ESPLEX CONTROL ARGUMENTS  
20 LDC 200  
21 000200 DIOSH: JMP  
22 00201 000000 ITRET: 0  
23 00202 001132 BGADR: NSTR  
24 00203 000000 PASS: 0  
25 00204 000003 PASSIN: 3  
26 00205 000003 PASSV1: 3  
27 00206 000000 ITR: 0  
28 00207 000000 ITRCT: 0  
29 00210 000000 ITRER: 0  
30 00211 000000 ITRCC: 0  
31 00212 000000 ERGT: 0  
32 00213 000000 LOPRET: 0  
33 00214 000000 RELOC: 0  
34 00215 000000 LISTNG: 0  
35 00216 000000 TESTN: 0  
36 00217 015663 MINLOC: PRGEND\*200,\*1  
37 000220 :LOC 220 :MANUAL MPMU/MMPUI DUMP ENTRY  
38 00220 106470 DMCC: EJSR .DNW7  
39 014271  
40 :NOT A  
41 00222 077077 M\F052: HALTA 3 :\*\*\*\*\* FATAL ERROR HALT \*\*\*\*\*  
42 00223 000220 JMP \*-3 :AGAIN ON CONTINUE  
43  
44  
45 00224 100030 DSPXX: XOP 0,0,0 :ERROR CALL "XOP"  
46 00225 000000 PRNSW: 0  
47 00226 000000 MAXLOC: 0 :MAX LOC WHERE PROG CAN GO  
48 00227 000000 ACO: 0  
49 00230 000000 AC1: 0  
50 00231 000000 AC2: 0  
51 00232 000000 AC3: 0  
52 00233 000000 CRY: 0  
53 00234 015330 ICATSW: CATSW :  
54 00235 015333 ISWREG: SWREG :\* POINTER TO SWITCH REG.  
55 00236 000000 ICAT: 0 :CAT STARTING ADDRESS  
56 00237 114630 LWERR: XOP 0,3,\*6 :FATAL ERROR CALL  
57 00240 020000 KOMST: 20000 :8K  
58 00241 000000 M0FLG: 0

0052 ESPCL  
01 :NOTE: USER PAGE ZERO ENTRIES ARE INSERTED BELOW  
02 PAGE0: (ENTRX-M0FLG)-1 :OCTAL ENTRIES ARE AVAILABLE CURRENTLY  
03 00242 000127 :CAUTION:  
04 :-----  
05 : DO NOT EXCEED NUMBER OF ENTRIES AVAILABLE  
06 :  
07 : PAGE  
08 :  
09 :\*\*  
10 :\*\*ESPCLX PAGE ZERO ENTRIES  
11 :  
12 00243 000000 CALL1: 0 :CALL ROUTINE TEMPORARY  
13 00244 000000 CALR2: 0 :CALL ROUTINE TEMPORARY  
14 00245 000000 LEFLC: 0 :USED IN ELFF TESTS  
15 00246 000000 LEFLC: 0 :USED IN ELFF MULTI-INDIRECT TESTS  
16 00247 000000 LPSV0: 0 :SAVE LOC FOR ELFF TESTS  
17 00250 000000 LDALC: 0 :TEMP LOC FOR ELDA TESTS  
18 00251 000000 LDSV0: 0 :SAVE LOC. FOR ELDA TESTS  
19 00252 000000 LDSV1: 0 :SAVE LOC. FOR ELDA TESTS  
20 00253 000000 STLC1: 0 :SAVE LOC. USED IN ESTA TESTS  
21 00254 000000 STLC1: 0 :USED IN ESTA TESTS  
22 00255 000000 STSV0: 0 :CONSTANT  
23 00256 000144 D109: 100. :USED IN EJMP AND EJSR TESTS  
24 00257 002260 JMLC0: JMP :USED IN EJMP AND EJSR TESTS  
25 00260 000000 JMLC1: 0 :USED IN EJMP AND EJSR TESTS  
26 00261 000000 JMLC2: 0 :USED IN EJMP AND EJSR TESTS  
27 00262 000000 JMLC3: 0 :USED IN EJMP AND EJSR TESTS  
28 00263 100030 JMLC4: XOP 0,0,0 :\*\*ERROR CALL  
29 :\*\*USED IN EJMP AND EJSR TESTS  
30 00264 000000 ISLC0: 0 :USED IN EISZ AND EDSZ TESTS  
31 00265 000000 ISLC1: 0 :USED IN EISZ AND EDSZ TESTS  
32 00266 000000 ISLC2: 0 :USED IN EISZ AND EDSZ TESTS  
33 00267 000000 DSIC0: 0 :USED IN DSPA TESTS  
34 00270 000000 DSIC1: 0 :USED IN DSPA TESTS  
35 00271 000000 DSSV0: 0 :USED IN DSPA TESTS  
36 00272 000000 DSSV1: 0 :USED IN DSPA TESTS  
37 00273 000000 DSSV2: 0 :USED IN DSPA TESTS  
38 00274 000000 LDSTB: 0 :USED IN DSPA TEST  
39 00275 000000 HDSTB: 0 :CONSTANT  
40 00276 000001 ONE: 1 :CONSTANT  
41 00277 006000 B0: TEM\*20 :TEMP LOCATION  
42 00300 000000 TEM: 0 :USED IN DSPA TESTS  
43 00301 000000 MXTRL: 0 :USED IN DSPA TESTS  
44 00302 000000 YLSIZ: 0 :CONSTANT  
45 00303 000002 D2: 2.  
46 00304 000003 D3: 3.  
47 :  
48 : PAGE 0 LOCATIONS FOR ERCC  
49 :  
50 :  
51 00305 000000 MFAD: 0 :DIA ERCC  
52 00306 000000 MFAD1: 0 :DIB ERCC  
53 00307 000000 MFCD: 0 :ERCC FAULT CODE  
54 00310 000366 C366: 366 :CONSTANT  
55 00311 100030 DSPER: XOP 0,0,0 :\*\*ERR ERROR CALL XOP WHICH REPLACES  
56 :\*\*JSR CENTER.  
57 00371 012016 ENTRX: INITZ :SETUP LINKAGE  
58 00372 000000 LAST: 0 :\*DOA MPMU/MMPUI WORD  
59 00373 000000 ERRR: 0 :CONTROL ERR RETURN  
60 00374 000000 NSCAN: 0 :TRACE SCAN COUNTER



0053 ESPCL  
01 00375 000000 ID: 0

PROGRAM ERROR ID WORD

0054 ESPCL

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

MAP/RELOCATION CONTROL ENTRIES BELOW  
.LOC 376 :CONTROL ENTRIES  
:MAP/RELOCATION  
:CONTROL ENTRIES  
:INHIBIT MAP MODE EXECUTION NOT = 0  
:ENABLE MAP EXECUTION IF = 0  
:RELOCATION BASE (FIXED) IF NOT = 0  
:NOTE:  
:MUST BE IN THE RANGE 20000 (OCTAL) OR GREATER  
:AND 8K LESS THAN THE CONTENTS OF MAXLOC:  
CAUTION  
:-----  
:"DMAP:" AND "RELWD:" ENTRIES ARE ONLY VALID WHEN  
MADE (SET) IN NON-MAPPED MODE.  
:INITIALIZE  
:MMPU/MMPUI  
:FLAG  
:MAPP: 0 ;MMPU = 0 ;MMPUI = 1  
: 1600  
26 00441 020777 CKMP: LDA 0,CKMP-1  
27 00442 063003 DDC 0,MMPU  
28 00443 064403 DIA 1,MMPU  
29 00444 107400 AND 0,1  
30 00445 106414 SUB# 0,1,SZR  
31 00446 000403 JMP --+3  
32 00447 126520 SUBZL 1,1  
33 00450 000402 JMP --+2  
34 00451 124510 XOR 1,1  
35 00452 044161 STA 1,MAPP ;MMPU = 0  
36 00453 125005 MOV 1,1,SNR ;MMPU = 1  
37 00454 001400 JMP 0,3 ;MMPUI = 1  
38 00455 102070 EJMPC ENZ,0 ;SETUP PHY MEMORY MASK CONTROL  
39 013134

: RELOCATABLE LOWER BUFFER AREA  
:\*  
:\* ONCE ONLY MONITOR SCAN TABLE  
:\* (TYPICALLY)  
:LOBUFF: 000011 ;H001  
:000500: THRU  
:000552: 000477 ;H053  
: ETC. THRU SCAN DELIMITER  
:000xxx: -1 ;SCAN TABLE DELIMITER



```

0057 ESPCL                               :
01 02 00731 077077 XHLT7: HALTA 3        ;XOP* INSTRUCTION FAILURE
02                                     ;SUGGEST YOU RUN BASIC ECLIPSE
03                                     ;DIAGNOSTICS.(NO GO)
04 00732 000777  ;RETURN BELOW ON SUCCESSFUL "POPB" ABOVE   CURRENT BIT(S) <1-4>
05 XOPR: JMP *-1 ;AGNOSTICS.(NO GO)
06 00733 020440  LDA 0,XOP.6 ;ACS,ACD
07 00734 163770  ADDI 4000,0 ;ADVANCE
08 00735 004000  ANDI 74000,0 ;
09 00736 143770  STA 0,XOP.6 ;
10 00740 040833  LDA 1,XOP.1 ;XOP* COMMAND CURRENT
11 00741 024425  ANDI 103777,1 ;LESS ACS,ACD
12 00742 147770  ;
13 00743 103777  ;
14 15 00744 104410  ;
16 00745 044421  ;
17 18 00746 101034  ;
19 00747 000647  ;
20 21 00750 020416  ;
22 00751 163770  ;
23 24 00753 143770  ;
25 26 00755 040411  ;
27 00756 010413  ;
28 00757 034412  ;
29 00760 010410  ;
30 00761 000635  ;
31 00762 010135  ;
32 00763 000401  ;
33 00764 002616  ;
34 00765 100030  XOP.0: XOP
35 00766 100030  XOP.1: XOP
36 00767 177740  XOP.2: -32.
37 00770 177740  XOP.3: -32.
38 00771 000000  XOP.4: 0
39 00772 100647  XOP.5: 8XHLT0
40 00773 000000  XOP.6: 0
41                                     ;EJEC

```

```

0056 ESPCL
01
02
03
04
05
06
07
08 01010 000000
09
10
11
12
13 01131 000000
14
15
16
17

```

PROGRAM STACK AREA BELOW

```

; STAKB:
; FLOCATION:
; ESTAK: 0
; THR:
; FLOCATION:
; ESTAK: 0
; NOTES:
; THE USER STACK IS RELOCATED DURING RELOCATION
; THE PROG STACK IS PRESET TO THE ABOVE AREA

```

```

0059 ESPCL
01 0060 ESPCL
02 01132 062677 NSTRT: IORST
03 01133 006120 JSR
04 01134 000001 1
05
06
07
08
09 01135 014216 PSEUDO: DSZ
10 01136 000401 NOP
11
12 01137 010135 ISZ
13 01140 000431 JMP
14 01141 162070 ELEF
15
16 01143 040040 STA
17 01144 040041 STA
18 01145 163770 ADDI
19
20 01147 040042 STA
21 01150 162070 ELEF
22
23 01152 040043 STA
24 01153 106470 EJSR
25
26 01155 032165 LDA
27 01156 151035 MOVZ#
28 01157 000412 JMP
29 01160 020237 LDA
30 01161 176070 ELEF
31
32 01163 031400 MOST:
33 01164 151415 INC#
34 01165 000404 JMP
35 01166 041000 STA
36 01167 175400 IAC
37 01167 000773 JMP
38 01171 106470 EJSR
39 012003
40 01173 100030 XOP
41 01174 162070 ELEF
42
43 01176 040127 STA
44
45 01177 006060 IN$TK
46 01200 162470 JSR
47
48 01202 040044 ELEF
49
50 01204 101005 XSTRT:
51 01205 000463 JMP NOCAT
52 01206 106470 EJSR SIZE
53
54 01210 126040 ADCO 1,1
55 01211 147000 ADD 2,1
56 01212 044062 STA 1,MENTOP
57 01213 020145 :RETURN WITH MENTOP IN AC2
58
59 01214 166470 :FLOOD AC1 WITH LENGTH OF PROG.
60 014136 ELEF

SUB 0,2
ADCZ# 1,2,SNC
JMP NOCAT
STA 2,MAXLOC
LDA 1,C400
ADD 1,2
STA 2,ICAT
JFM 8015M
LDA 2,MAXLOC
LDA 1,MINLOC
SUB 1,2
STA 2,MAXLOC
.ENDC

:IFE 80ASW
PRINT PROG
:PRINT MESSAGE PROG
PROG 3,3,4
LDA 1,MAPSW
MOV 1,1,SNR
JMP SWMS0
EJSR PRMAP

JMP SWMS1
PRINT MESIZ
XOP 3,3,4
MESIZ
LDA 1,MENTOP
OCIP
XOP 1,1,2
:PRINT (AC1) OCTAL
CRLFPR
XOP 1,1,5
:OUTPUT CR/LF
LDA 3,EGGS
LDA 1,0,3
MOV 1,1,SZR
JMP SWMS2
PRINT SETSW
XOP 3,3,4
SETSW
CRLFPR
XOP 1,1,5
:OUTPUT CR/LF
HALT
READS 0
ANDJ 177776,0
STA 0,815WREG
.ENDC
PRINT SWTS
XOP 3,3,4
SWTS
LDA 1,815WREG
OCIP
XOP 1,1,2
:PRINT (AC1) OCTAL
CRLFPR
XOP 1,1,5
:OUTPUT CR/LF
JMP START
EJSR SIZE
LDA 1,C201

:NOTE PC ERROR BELOW INDICATES THAT THE ERROR HAS
:TAKEN PLACE PRIOR TO NORMAL TEST CODE EXECUTION
:
:PSEUDO: DSZ
NOP
ISZ
JMP
ELEF
STA
STA
ADDI
STA
ELEF
STA
EJSR
LDA
MOVZ#
JMP
LDA
ELEF
MOST:
INC#
JMP
STA
IAC
JMP
EJSR
XOP
ELEF
STA
JSR
ELEF
XSTRT:
JMP NOCAT
EJSR SIZE
ADCO 1,1
ADD 2,1
STA 1,MENTOP
LDA 0,C2000
:RETURN WITH MENTOP IN AC2
:FLOOD AC1 WITH LENGTH OF PROG.

```

```

0061 ESPCL
01 01273 022234
02 01274 101004
03 01275 024145
04
05 01276 132400
06 000001
07 01277 054217
08 01300 132400
09
10 01301 050226
11 01302 126440
12 01303 046234
13 01304 000725

```

```

LDA 0,0ICATSW
MOV 0,0,SRZ
LDA 1,C2000

SUB 1,2
IFN BDISW
LDA 1,MINLOC
SUB 1,2
-ENDC
STA 2,MAXLOC
SURO 1,1
STA 1,0ICATSW
JMP SWMES

```

```

0062 FSPCL

```

```

01 01305 000003
02
03
04
05 01306 176520
06 01307 054374
07 01310 174400
08 01311 054203
09 01312 054214
10 01313 054150
11 01314 054155
12 01315 054375
13 01316 054215
14 01317 054225
15 01320 156470
16 01322 012741
17
18 01323 040204
19 01324 020156
20 01325 040154
21 01326 020151
22 01327 101005
23 01330 000404
24 01331 162470
25
26 01333 040002
27 01334 024217
28 01335 030226
29
30 01336 133000
31 01337 110110
32
33 01341 050133
34 01342 024067
35 01343 044132
36 01344 102440
37 01345 163770
38 123456
39 01347 040065
40 01350 024735
41 01351 065102
42 01352 126000
43 01353 066177
44 01354 000073
45
46
47

```

```

;IF BDTOS SET CATSW, DO NOT
;WIPE IT OUT WITH ANOTHER PRG

START:
REG:
SUBZL 3,3
STA 3,3
SUB 3,3
STA 3,PASS
STA 3,RELOC
STA 3,MPSM
STA 3,MPEM
STA 3,I0
STA 3,TESTN
STA 3,PRNSK
ESTA 3,FRID

LDA 0,PASSYL
STA 0,PASSIN
LDA 0,MNSB
STA 0,MP8GN
LDA 0,MAPSM
MOV 0,0,SNR
JMP .+4
ELEF 0,SCLC

STA 0,2
LDA 1,MINLOC
LDA 2,MAXLOC
ADD 1,2
SBI 1,2
STA 2,STSTOP
STA 2,MIBFU
LDA 1,8BFF
STA 1,L0MBF
SURO 0,0
ADDI 123456,0

STA 0,RAN
LDA 1,MEG-1
DDAS 1,ERCC
ADC 1,1
DOHS 1,CPU
JMP HAMZ
EXPAND USER INTERRUPT SERVICE BELOW

;*
INTRP:

```

```

:TRACE SCAN COUNTER
:CLEAR PASS COUNT
:SET RELOCATION TO 0
:CLEAR CURRENT RUN MODE SW
:CLEAR END OF MAP LOC.
:PROGRAM ERROR ID WORD.
:**
:PROCESS FLAG
:RESET INTERNAL PASS COUNT.
:INITIALIZE FOR MAP.
:IS MAP PRESENT ?
:YES,SET UP SCL HANDLER ADDR
:NO,DO NOT USE LOC. 2
:
:MINLOC:STORE TOPMOST USABLE
:MAXLOC:ADDRESS IN TSTOP
:
:STSTOP
:INITIALISE TOP OF UPPER SUFF
:INITIALISE BOTTOM OF LOWER
:BUFFER FOR INITIAL PASS
:CLR RAND # SAVE LOC.
:RESET RAND # GEN.
:
:MEG-1 :3.
:ERCC
:TURN ON THE INTERRUPTS
:AND START THE PROGRAM

```

0063 ESPCL

```

01 INTRU
02
03
04
05
06 SKPOZ CPU
07 JMP PFAIL *(YES), GO AWAIT RESTART & PRINT
08 PSH 0.3
09 MOVR 0.0
10 PSH 1.0
11 INTA 0
12 LDA 1,ERADDR+1
13 SUB# 0.1,SZM
14 INTBAD
15 JMB
16 DIA 1,ERCC
17 STA 1,MFAD
18 DIB 1,ERCC
19 MOVR 1.1
20 STA 1,MFADI
21 LSH 0.1
22 LDA 0,C366
23 STA 1,MFCD
24 CRLFPR
25 XOP 1,1,5
26 PRINT MS1
27 XOP 3,3,4
28 MSI
29 CRLFPR
30 XOP 1,1,5
31 ANDI 1,0
32 LDA 1,MFAD
33 DIV 2,C2000
34 STA 0,ERADDR
35 PRINT MS8
36 XOP 3,3,4
37 MSI
38 DECPR MODULE
39 ELDA 1,MODULE
40 XOP 1,1,3
41 PRINT MS3
42 XOP 3,3,4
43 MS3
44 ELDA 1,ERADDR
45 XOP 1,1,2
46 CRLFPR
47 XOP 1,1,5
48 PRINT MS4
49 XOP 3,3,4
50 MS4
51 OCTPR MFCD
52 ELDA 1,MFCD
53 XOP 1,1,2
54
55
56
57
58
59
60

```

0064 ESPCL

```

01
02 01432 174430
03 01433 015061
04
05 01434 126470
06 076343
07 01436 124230
08
09 01437 124530
10 01440 106470
11 011611
12 01442 020304
13 01443 061102
14 01444 000401
15 01445 060403
16 01446 101212
17 01407 061003
18 01450 103210
19 01451 101100
20 01452 163210
21 01453 060177
22 01454 002000
23 01455 000000 MODULE: 0
24 01456 000000 ERADDR: 0
25 01457 000002
26 01440 020403 PFAIL: 0
27 01461 040000 HALTA
28 01462 063077
29 01463 002160 PFLRTN: PRINT
30 XOP 3,3,4 MSPPFL
31 01464 174430 MSPPFL
32 01465 015233
33 01466 002202
34
35
36
37
38
39
40 01467 063077 INTBAD: HALTA
41 01470 002202 H\F033: JMP

```

```

PRINT MS7
XOP 3,3,4
MS7
OCTPR 0
ELDA 1,0
PRINT CONTENTS OF
: *0 IN OCTAL
: *OUTPUT CR/LF
: PRINT MAP DATA
LDA 0,03
DOAS 0,ERCC
JMP +1
DIA 0,MMPU
MOVH# 0,0,SZC
DGA 0,0
POP 0,0
MOVL 0,0
POP 3,0
JMP @0
: RESTORE THE STATE OF MACHINE
: **ERCC
: *SETUP RESTART LOC. 0
: *HERE
: *WAIT FOR POWER
: *E.PFLRTN BELOW
: *PRINT POWER FAILED
: *PRINT MESSAGE MSPPFL
: *ATTEMPT PROG RESTART
: ** ILLEGAL INTERRUPT
INTBAD:
H\F033: HALTA
@BGNADR : **DEV # CODE IN (C0)

```



```

10067 ESPCL
01
02
03 01607 006120 :TEST "LOB"
04 01607 006120 :INITIALIZE TEST
05 01610 000100 :ITERATION VALUE
06 01611 006121 :INCREASE THE NUMBER
07 01612 105300 :OF ZEROS IN THE RANDOM
08 01613 123400 :C(AC0)=RANDOM #
09 01614 111000 :C(AC2)=RAND
10 01615 174510 :C(AC3)=0
11 01616 156410 :COUNT LEADING ZEROS
12 01617 156410 :IN C(AC3) FROM THE
13 01620 156410 :WORD IN C(AC2)
14 01621 156410 :BECAUSE ZERO COUNT HAS
15 01622 165220 :BEEN MADE X4 SHIFT RIGHT
16 01623 125220 :SHIFTING SHOULD LOSE NO BITS
17 01624 131210 :LSH 1,2
18 01625 124400 :NEG 1,1
19 01626 131210 :LSH 1,2
20 01627 142414 :SUB# 2,0,SZR
21 01630 100030 :ERROR
22 01631 006117 :XOP 0,0,0
23 01632 100030 :LOOP
24 01633 006117 :JSR @ENTLO
25 01634 006117 :ITERATE TEST ROUTINE
26
27 01632 006120 :TEST "COB"
28 01633 000200 :INITIALIZE TEST
29 01634 174610 :ITERATION VALUE
30 01635 182001 :SET C(AC3) TO 0
31 01636 000400 :SET C(AC0) TO 177777
32 01637 030777 :116 BITS X 16 COB INSTRUCTIONS
33 01640 116610 :LDA 2,-1
34 01641 116610 :COB 0,3
35 01642 116610 :COB 0,3
36 01643 116610 :COB 0,3
37 01644 116610 :COB 0,3
38 01645 116610 :COB 0,3
39 01646 116610 :COB 0,3
40 01647 116610 :COB 0,3
41 01650 116610 :COB 0,3
42 01651 116610 :COB 0,3
43 01652 116610 :COB 0,3
44 01653 116610 :COB 0,3
45 01654 116610 :COB 0,3
46 01655 116610 :COB 0,3
47 01656 116610 :COB 0,3
48 01657 116610 :SUB# 2,3,SZR
49 01660 156414 :ERROR
50 01661 100030 :XOP 0,0,0
51 01662 006117 :LOOP
52 01663 006117 :JSR @ENTLO
53 01664 006117 :ITERATE TEST ROUTINE

```

```

10067 ESPCL
01
02
03 01607 006120 :TEST "LOB"
04 01607 006120 :INITIALIZE TEST
05 01610 000100 :ITERATION VALUE
06 01611 006121 :INCREASE THE NUMBER
07 01612 105300 :OF ZEROS IN THE RANDOM
08 01613 123400 :C(AC0)=RANDOM #
09 01614 111000 :C(AC2)=RAND
10 01615 174510 :C(AC3)=0
11 01616 156410 :COUNT LEADING ZEROS
12 01617 156410 :IN C(AC3) FROM THE
13 01620 156410 :WORD IN C(AC2)
14 01621 156410 :BECAUSE ZERO COUNT HAS
15 01622 165220 :BEEN MADE X4 SHIFT RIGHT
16 01623 125220 :SHIFTING SHOULD LOSE NO BITS
17 01624 131210 :LSH 1,2
18 01625 124400 :NEG 1,1
19 01626 131210 :LSH 1,2
20 01627 142414 :SUB# 2,0,SZR
21 01630 100030 :ERROR
22 01631 006117 :XOP 0,0,0
23 01632 100030 :LOOP
24 01633 006117 :JSR @ENTLO
25 01634 006117 :ITERATE TEST ROUTINE
26

```





```

10071 ESPCL
01
02
03
04 01743 006120 T\N006: JSR @BENTIN ;TEST "HXR"
05 01744 000100 RAND ;INITIALIZE TEST
06 ;ITERATION VALUE
07 01745 006121 JSR @BENTRA ;C(ACO)=RANDOM #
08 01746 034402 LDA 3,*+2 ;REMOVE 4 LOW ORDER BITS
09 01747 117401 AND 0,3,SKP ;FROM C(RANDOM) AND PUT IN
10 01750 177760 MOV 0,1 ;C(ACS).
11 01751 105000 MOV 0,1
12 01752 105000 MOV 0,1
13 01753 105100 HXR 1,0 ;C(ACO), RIGHT 4 PLACES
14 01754 105110 HXR 1,1 ;C(ACI), RIGHT 4 PLACES
15 01755 105120 HXR 1,0 ;C(ACO), LEFT 4 PLACES
16 01756 105130 HXR 1,1 ;C(ACI), LEFT 4 PLACES
17 01757 105140 HXR 1,0 ;C(ACO), RIGHT 4 PLACES
18 01760 105150 HXR 1,1 ;C(ACI), RIGHT 4 PLACES
19 01761 101410 HXR 1,0 ;C(ACO), LEFT 4 PLACES
20 01762 105410 HXR 1,1 ;C(ACI), LEFT 4 PLACES
21 01763 101510 HXR 1,0 ;C(ACO), RIGHT 4 PLACES
22 01764 105510 HXR 1,1 ;C(ACI), RIGHT 4 PLACES
23 01765 101410 HXR 1,0 ;C(ACO), LEFT 4 PLACES
24 01766 105410 HXR 1,1 ;C(ACI), LEFT 4 PLACES
25 01767 101510 HXR 1,0 ;C(ACO), RIGHT 4 PLACES
26 01770 105510 HXR 1,1 ;C(ACI), RIGHT 4 PLACES
27 01771 101410 HXR 1,0 ;C(ACO), LEFT 4 PLACES
28 01772 105410 HXR 1,1 ;C(ACI), LEFT 4 PLACES
29 01773 101510 HXR 1,0 ;C(ACO), RIGHT 4 PLACES
30 01774 105510 HXR 1,1 ;C(ACI), RIGHT 4 PLACES
31 01775 101410 HXR 1,0 ;C(ACO), LEFT 4 PLACES
32 01776 105410 HXR 1,1 ;C(ACI), LEFT 4 PLACES
33 01777 101510 HXR 1,0 ;C(ACO), RIGHT 4 PLACES
34 02000 105510 HXR 1,1 ;C(ACI), RIGHT 4 PLACES
35 02001 101410 HXR 1,0 ;C(ACO), LEFT 4 PLACES
36 02002 105410 HXR 1,1 ;C(ACI), LEFT 4 PLACES
37 02003 101510 HXR 1,0 ;C(ACO), RIGHT 4 PLACES
38 02004 105510 HXR 1,1 ;C(ACI), RIGHT 4 PLACES
39 02005 101410 HXR 1,0 ;C(ACO), LEFT 4 PLACES
40 02006 105410 HXR 1,1 ;C(ACI), LEFT 4 PLACES
41 02007 101510 HXR 1,0 ;C(ACO), RIGHT 4 PLACES
42 02010 105510 HXR 1,1 ;C(ACI), RIGHT 4 PLACES
43 02011 101410 HXR 1,0 ;C(ACO), LEFT 4 PLACES
44 02012 105410 HXR 1,1 ;C(ACI), LEFT 4 PLACES
45 02013 162415 SUB# 3,0,SNR ;C(ACS)=CORRECT
46 02014 166414 SUB# 3,1,SZR ;C(AC2)=ORIGINAL
47 ERROR ;C(ACO-1)=HSR/HSL RESULT
48 02015 100030 E\N006: XOP 0,0,0 ;ERROR CALL
49 LOOP
50 02016 006117 L\N006: JSR @BENTLO ;ITERATE TEST ROUTINE
51
10072 ESPCL
01
02
03
04 02017 006120 T\N007: JSR @BENTIN ;INITIALIZE TEST
05 02020 000020 RAND ;ITERATION VALUE
06 ;C(ACO)=RANDOM #
07 02021 006121 JSR @BENTRA
08 02022 101000 MOV 0,0
09 02023 105000 MOV 0,0+183
10 02024 131000 MOV 0+183,0+283
11 02025 141610 DHXL 3,0
12 02026 141710 DHXR 3,0
13 02027 034402 LDA 0+383,*+2
14 02030 157401 AND 0+283,0+383,SKP
15 02031 000017 L7
16 02032 115415 SUB# 0,0+383,SNR ;CHECK HIGH PART
17 02033 132414 SUB# 0+183,0+283,SZR ;CHECK LOW PART
18 ERROR ;SHIFTING 12 FAILED
19 02034 100030 E\N007: XOP 0,0,0 ;ERROR CALL
20 LOOP
21 02035 006117 L\N007: JSR @BENTLO ;ITERATE TEST ROUTINE
22
23
24
25 02036 006120 T\N010: JSR @BENTIN ;INITIALIZE TEST
26 02037 000020 RAND ;ITERATION VALUE
27 ;C(ACO)=RANDOM #
28 02040 006121 JSR @BENTRA
29 02041 105000 MOV 0,1
30 02042 131000 MOV 1,1+183
31 02043 155000 MOV 1+183,1+283
32 02044 145610 DHXL 3,1
33 02045 145710 DHXR 3,1
34 02046 020402 LDA 1+383,*+2
35 02047 163401 AND 1+283,1+383,SKP
36 02050 000017 L7
37 02051 122415 SUB# 1,1+383,SNR ;CHECK HIGH PART
38 02052 156414 SUB# 1+183,1+283,SZR ;CHECK LOW PART
39 ERROR ;SHIFTING 12 FAILED
40 02053 100030 E\N010: XOP 0,0,0 ;ERROR CALL
41 LOOP
42 02054 006117 L\N010: JSR @BENTLO ;ITERATE TEST ROUTINE

```

10073 ESPCL

```

01 02055 006120  \N011: JSR  @ENTLO  ;ITERATE TEST ROUTINE
02 02056 000020  RAND
03 02057 006121  JSR  @ENTRA
04 02058 111000  MOV  2+2+183
05 02059 155800  MOV  2+183+2+283
06 02060 161000  DXHL 3+2
07 02061 151610  LDA  2+283+*2
08 02062 024402  AND  2+283+2+383,SKP
09 02063 107401  SUB# 2+2+383,SNR
10 02064 146415  SUB# 2+183+2+283,SZR
11 02065 162414  ERROR
12 02066 107401  XOP  0,0,0
13 02067 000017  LOOP
14 02068 100030  \N011: JSR  @ENTLO  ;ITERATE TEST ROUTINE
15 02069 006117  \N011: JSR  @ENTLO  ;ITERATE TEST ROUTINE
16 02070 006120  \N012: JSR  @ENTLO  ;ITERATE TEST ROUTINE
17 02071 000020  RAND
18 02072 006121  JSR  @ENTRA
19 02073 115000  MOV  3+3+183
20 02074 105000  MOV  3+183+3+283
21 02075 155710  DXHL 3+3
22 02076 155710  LDA  3+383+*2
23 02077 133401  AND  3+283+3+383,SKP
24 02078 004017  SUB# 3+3+383,SNR
25 02079 172415  SUB# 3+183+3+283,SZR
26 02080 106414  ERROR
27 02081 100030  \N012: XOP  0,0,0
28 02082 006117  \N012: JSR  @ENTLO  ;ITERATE TEST ROUTINE
29 02083 006117  \N012: JSR  @ENTLO  ;ITERATE TEST ROUTINE

```

10074 ESPCL

```

01 02113 006120  \N013: JSR  @ENTIN  ;INITIALIZE TEST
02 02114 000005  ITERATION VALUE
03 02115 030132  LDA  2+L0MBF
04 02116 020433  LDA  0,LIT0
05 02117 041000  STA  0,0,2
06 02118 041000  JSR  0,3
07 02119 020433  LDA  0,LIT2
08 02120 041001  STA  0,1,2
09 02121 041001  INC  2,3
10 02122 155400  INC  3,3
11 02123 024123  LDA  1,L0BFF
12 02124 024123  SUB  2,1
13 02125 146400  ADCZL 0,0
14 02126 102120  STA  0,TEM
15 02127 040300  SUB  0,0
16 02128 102400  RAM
17 02129 050123  SZBA1: LDA  2,L0BFF
18 02130 020417  LDA  0,LIT1;JSR  0,2 I.E.=FAILS TO SKIP A "JSR" TO
19 02131 041377  STA  0,-1,2 ;ERROR IS EXECUTED
20 02132 041376  STA  0,-2,2
21 02133 004402  JSR  *2
22 02134 000410  JMP  SZBA4
23 02135 171000  MOV  3,2
24 02136 004402  JSR  *2
25 02137 000404  JMP  SZBA3
26 02138 020411  LDA  0,LIT3
27 02139 024411  LDA  1,LIT4
28 02140 002132  SZBA2: JMP  @L0MBF
29 02141 100030  SZBA3: ERROR
30 02142 000000  \N013: XOP  0,0,0
31 02143 000000  SZBA4: LOOP
32 02144 006117  \N013: JSR  @ENTLO
33 02145 000406  JMP  LIT4+1
34 02146 106210  LIT0: SZB  0,1
35 02147 005000  LIT1: JSR  0,2
36 02148 005400  LIT2: JSR  0,3
37 02149 000300  LIT3: TEM
38 02150 000017  LIT4:
39 02151 000017  LIT4:
40 02152 000017  LIT4:

```

```

10076 ESPCL
01
02
03
04 02177 006120 T\N015: JSR @ENTIN          ;TEST "BTZ"
05 02200 000040 BTZ3:                       ;INITIALIZE TEST
06                                     ;ITERATION VALUE
07 02201 006121          ;C(AC0)=RANDOM #
08 02202 105045          ;IN C(TEM). FOR EACH (1) BIT
09 02203 000776          ;IN THE WORD A "BTZ" POINTER
10 02204 040300          ;IS GENERATED AND A "BTZ"
11 02205 034277          ;EXECUTED. C(TEM) SHOULD BE
12 02206 116510          ;CHANGED TO ALL ZEROS BY
13 02207 176110          ;THE SEVERAL "BTZ" INSTRUCTIONS
14 02210 101004          ;WILL ITERATE FOR EACH (1) BIT
15 02211 000774          JMP #-4
16 02212 030300          LDA 2,TEM
17 02213 101002          MOV 0,0,SZC
18 02214 151014          MOV# 2,2,SZR
19                                     ERROR
20 02215 100030 E\N015: XOP 0,0,0
21                                     LOOP
22 02216 006117 L\N015: JSR @ENTLO          ;ITERATE TEST ROUTINE
23

```

```

10075 ESPCL
01
02
03 02156 006120 T\N014: JSR @ENTIN          ;TEST "BTO"
04 02157 000040 BT4.1:                       ;INITIALIZE TEST
05                                     ;ITERATION VALUE
06                                     ;A RANDOM NUMBER IS STORED
07 02160 006121          ;C(AC0)=RANDOM #
08 02161 104000          ;IN C(TEM). FOR EACH (0) BIT
09 02162 125045          ;IN THE NUMBER, A POINTER IS
10 02163 000775          ;GENERATED AND A "BTO" INST-
11 02164 040300          ;RUCTION EXECUTED. THUS
12 02165 034277          ;AFTER THE SEVERAL BTO
13 02166 136510          ;INSTRUCTIONS THE WORD IN MEMORY
14 02167 176010          ;SHOULD BE ALL ONES.
15 02170 125004          ;ITERATE FOR EACH (0) BIT.
16 02171 000774          JMP #-4
17 02172 024300          LDA 1,TEM
18 02173 101002          MOV 0,0,SZC
19 02174 124014          COM# 1,1,SZR
20 02175 100030 E\N014: XOP 0,0,0
21                                     ERROR
22 02176 006117 L\N014: JSR @ENTLO          ;ITERATE TEST ROUTINE
23

```

```

10077 ESPCL
01 02217 000403 JMP SZ5
02 02220 000020 LIT5: ;**
03 02221 102244 LIT6: 38752
04
05
06 02222 006120 SZ5: SETUP 40
07 02223 000040 TAN016: JSR @ENTIN
08 RAND
09 02224 006121 JSR @ENTRA
10 02225 042124 STA 0,@HIGHF
11 02226 024772 LDA 1,LIT5
12 02227 030772 LDA 2,LIT6
13 02230 176620 SUBZR 3,3
14 02231 104110 SZ51: SBI 1,1
15 02232 146310 SZB0 2,1
16 02233 175060 MOVC 3,3
17 02234 175203 MOVR 3,3,SNC
18 02235 000774 JMP #4
19 02236 026124 LDA 1,@HIGHBF
20 02237 162415 SUB# 3,0,SNR
21 02240 124014 COM# 1,1,SZR
22 ERROR
23 02241 100030 ENR016: XOP 0,0,0
24 LOOP
25 02242 006117 LAN016: JSR @ENTLO
26 02243 000405 JMP #5
27 02244 102245 SZ52: @+1
28 02245 102246 @+1
29 02246 102247 @HIGHF
30 02247 100124
31

10078 ESPCL
01
02
03
04 02250 020124 LDA 0,@HIGHF ;TEST "LDR"
05 02251 040300 STA 0,TEM
06 LD#5:
07 000020 I#N#2
08 02252 176470 ELEF 3,@I020 ;SPECIAL
09 000027 ;ERR LINK
10 02254 054373 STA 3,ERRR ;ERR LINK
11 02255 006120 TAN017: JSR @ENTIN ;INITIALIZE TEST
12 02256 000005 ;ITERATION VALUE
13 RAND
14 02257 006121 JSR @ENTRA ;C(AC0)=RANDOM #
15 02260 043300 STA 0,@TEM ;IN EACH BUFFER LOCATION:
16 02261 030300 LDA 2,TEM ;STORE RANDOM NUMBER,
17 02262 151120 MOVZL 2,2 ;LOAD IT WITH "LDB"
18 02263 155400 INC 2,3 ;AND LOAD IT VIA "LDA"
19 02264 146710 LDB 2,1 ;CHECK ALL LOADS FOR
20 02265 176710 LDB 3,3 ;ACCURACY.
21 02266 125300 MOV# 1,1
22 02267 137000 ADD 1,3
23 02270 026300 LDA 1,@TEM ;C(AC0)=ORIGINAL/CORRECT
24 02271 122415 SUB# 1,0,SNR ;C(AC1)=WORD FROM BUFFER
25 02272 116414 SUB# 0,3,SZR ;C(AC2)=FIRST BYTE POINTER
26 ERROR ;C(AC3)=LDB RESULTS
27 02273 100030 ENR017: XOP 0,0,0 ;ERROR CALL
28 LOOP
29 02274 006117 LAN017: JSR @ENTLO ;ITERATE TEST ROUTINE
30 02275 010300 ISZ TEM
31 02276 020300 LDA 0,TEM ;NEXT BUFFER LOCATION
32 02277 034133 LDA 3,@HIGHF ;TEST FOR END OF
33 02300 116414 SUB# 0,3,SZR ;BUFFER
34 02301 000751 JMP LDB5
35 @I020:

```

```

10079 ESPCL
01
02
03 02302 020132 LDA 0,LOWBF ;TEST "STB"
04 02303 040300 STA 0,TEM
05
06
07 02304 176470 ELIF 3,BAL021 ;SPECIAL
08 000021 IEN+2
09 02306 054373 ;*ERR LINK
10 02307 006120 T^N020: JSR @ENTIN ;INITIALIZE TEST
11 02310 000903 3. ;ITERATION VALUE
12 ;STORE RANDOM #
13 JSR @ENTRA ;C(AC0)=RANDOM #
14 LDA 2,TEM ;VIA STB
15 MOVZL 2,2
16 MOVS 0,3
17 STB 2,3 ;STORE LEFT BYTE
18 INC 2,2 ;STORE RIGHT BYTE
19 MOVZK 2,2 ;C(AC3)=STB RESULT
20 02320 151220 LDA 3,0,2 ;C(AC0)=CORRECT
21 02321 035000 SUR# 3,0,SZR ;C(AC2)=WORD IN BUFFER.
22 02322 162414 ERROR
23 02323 100030 E^R020: XOP 0,0,0 ;ERROR CALL
24 ;LOOP
25 02324 006117 L^N020: JSR @ENTLO ;ITERATE TEST ROUTINE
26 02325 010300 ISZ TEM
27 02326 020300 LDA 0,TEM ;NEXT BUFFER WORD
28 02327 024123 LDA 1,LOBFU ;TEST FFOR ENO OF
29 02330 106414 SUR# 0,1,SZR ;BUFFER.
30
31 02331 000753 BAL021:
32

10080 ESPCL
01
02
03 ; TESTS FOR 'XCT', 'POP', 'PUSH' AND 'MSP'
04
05
06 ;**SETUP STACK DETECTION FOR "ESP01" TESTING BELOW
07 ;**
08 02332 162070 ELIF 0,400,0 ;**
09 000400 STA 0,SP ;**
10 02334 040040 STA 0,FP ;**
11 02335 040041 ADDI 32,**0
12 02336 163770
13 ;C(AC0)=RANDOM #
14 02340 040040 STA 0,SL
15 02341 020130 LDA 0,ST^F
16 02342 040045 STA 0,8F ;**
17
18 ;**
19 ESP01: SETUP 100. ;INITIALIZE TEST
20 02343 006120 T^N021: JSR @ENTIN ;ITERATION VALUE
21 02344 000144 RAND
22 JSR @ENTRA ;C(AC0)=RANDOM #
23 02345 006121 MOV 0,2 ;C(AC2)=RANDOM NUMBER
24 02346 111000 MOVS 2,3 ;C(AC3)=RANDOM NUMBER
25 02347 153300 ELEM 0,400,0 ;**
26 02350 162070
27 000400 STA 0,SP
28 02352 040040 LDA 0,PSH23
29 02353 020450 LDA 1,POP10
30 02354 024450 XCT 0
31 02355 123370 XCT 0 ;PUSH AC2-AC3 TO STACK
32 02356 123370 XCT 0 ;PUSH AC2-AC3 TO STACK
33 02357 123370 XCT 0 ;PUSH AC2-AC3 TO STACK
34 02360 123370 XCT 0 ;PUSH AC2-AC3 TO STACK
35 02361 123370 XCT 0 ;PUSH AC2-AC3 TO STACK
36 02362 123370 XCT 0 ;PUSH AC2-AC3 TO STACK
37 02363 123370 XCT 0 ;PUSH AC2-AC3 TO STACK
38 02364 123370 XCT 0 ;PUSH AC2-AC3 TO STACK
39 02365 123370 XCT 0 ;PUSH AC2-AC3 TO STACK
40 02366 123370 XCT 0 ;PUSH AC2-AC3 TO STACK
41 02367 123370 XCT 0 ;PUSH AC2-AC3 TO STACK
42 02370 123370 XCT 0 ;PUSH AC2-AC3 TO STACK
43 02371 123370 XCT 0 ;PUSH AC2-AC3 TO STACK
44 02372 123370 XCT 0 ;PUSH AC2-AC3 TO STACK
45 02373 123370 XCT 0 ;PUSH AC2-AC3 TO STACK
46 02374 123370 XCT 0 ;PUSH AC2-AC3 TO STACK
47 02375 123370 XCT 1 ;POP INTO AC1-AC0 FROM STACK
48 02376 112415 SUR# ;AC0 MUST BE =AC2
49 02377 136414 SUR# 1,3,SZR ;AC1 MUST RE =AC3
50 ERROR
51 02400 100030 E^R021: XOP 0,0,0 ;ERROR CALL
52 02401 030421 LDA 2,M^NS30 ;RESET THE SP BY 'MSP'
53 02402 113370 MSP 2
54 02403 030040 LDA 2,SP
55 02404 176070 ELEM 3,400,0 ;**
56 000400 SUB# 2,3,SZM ;'MSP' FAILED
57 02406 156414 ERRORH
58
59 02407 100030 E^R022: XOP 0,0,0 ;ERROR CALL
60 02410 104710 XCH 0,1 ;C(AC0)=RAND NUMBER

```

```

0081 ESPCL
01 02411 030414 LDA
02 02412 030414 LDA
03 02413 133370 XCI
04 02414 137370 XCI
05 02415 112415 SUB#
06 02416 136414 SUB#
07 ERROR
08 02417 100030 EAX023: XOP
09 LOOP
10 02420 006117 LAN021: JSR
11 02421 004006 JMP
12 02422 177742 MMS30: -30.
13 02423 157110 PSH23: PSH
14 02424 129210 POP10: POP
15 02425 107110 PSH01: PSH
16 02426 173210 POP32: POP
17
18 INSTK
19 02427 006060 JSR
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

;C(AC1)=RAND NUMBER
2,PSH01
3,POP32
2
3
0,2,SNR
1,3,SRZ
0,0,0
;ERROR CALL
;ITERATE TEST ROUTINE
;INITIALIZE SYS STACK
;C(AC1)=AC1 TO STACK
;POP INTO AC3-AC2 FROM STACK
;AC0 SB =AC2
;AC1 SB =AC3
;AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;POPJ' MUST JUMP TO 'ESP2X'
;ERROR CALL
;SHOULD NEVER HALT HERE
;ITERATE TEST ROUTINE
;FILL UPPER AND LOWER
;SCRATCH BUFFER AREA
;WITH (XERR)
;INITIALIZE TEST
;ITERATION VALUE
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;ERROR CALL
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;ERROR CALL
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;POPJ' MUST JUMP TO 'ESP2X'
;ERROR CALL
;SHOULD NEVER HALT HERE
;ITERATE TEST ROUTINE
;TESTS FOR 'PSHR' AND 'POPJ'
;ESP02: FILL XERR
;JSR
;XERR
;JFILL
;ESP02: FILL XERR
;JSR
;XERR
;SETUP
;INITIALIZE TEST
;ITERATION VALUE
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;ERROR CALL
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;ERROR CALL
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;POPJ' MUST JUMP TO 'ESP2X'
;ERROR CALL
;SHOULD NEVER HALT HERE
;ITERATE TEST ROUTINE
;0082 ESPCL
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
;TESTS FOR 'PSHR' AND 'POPJ'
;ESP02: FILL XERR
;JSR
;XERR
;JFILL
;ESP02: FILL XERR
;JSR
;XERR
;SETUP
;INITIALIZE TEST
;ITERATION VALUE
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;ERROR CALL
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;ERROR CALL
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;POPJ' MUST JUMP TO 'ESP2X'
;ERROR CALL
;SHOULD NEVER HALT HERE
;ITERATE TEST ROUTINE
;0081 ESPCL
01 02411 030414 LDA
02 02412 030414 LDA
03 02413 133370 XCI
04 02414 137370 XCI
05 02415 112415 SUB#
06 02416 136414 SUB#
07 ERROR
08 02417 100030 EAX023: XOP
09 LOOP
10 02420 006117 LAN021: JSR
11 02421 004006 JMP
12 02422 177742 MMS30: -30.
13 02423 157110 PSH23: PSH
14 02424 129210 POP10: POP
15 02425 107110 PSH01: PSH
16 02426 173210 POP32: POP
17
18 INSTK
19 02427 006060 JSR
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
;C(AC1)=RAND NUMBER
2,PSH01
3,POP32
2
3
0,2,SNR
1,3,SRZ
0,0,0
;ERROR CALL
;ITERATE TEST ROUTINE
;INITIALIZE SYS STACK
;C(AC1)=AC1 TO STACK
;POP INTO AC3-AC2 FROM STACK
;AC0 SB =AC2
;AC1 SB =AC3
;AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;POPJ' MUST JUMP TO 'ESP2X'
;ERROR CALL
;SHOULD NEVER HALT HERE
;ITERATE TEST ROUTINE
;FILL UPPER AND LOWER
;SCRATCH BUFFER AREA
;WITH (XERR)
;INITIALIZE TEST
;ITERATION VALUE
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;ERROR CALL
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;ERROR CALL
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;POPJ' MUST JUMP TO 'ESP2X'
;ERROR CALL
;SHOULD NEVER HALT HERE
;ITERATE TEST ROUTINE
;0082 ESPCL
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
;TESTS FOR 'PSHR' AND 'POPJ'
;ESP02: FILL XERR
;JSR
;XERR
;JFILL
;ESP02: FILL XERR
;JSR
;XERR
;SETUP
;INITIALIZE TEST
;ITERATION VALUE
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;ERROR CALL
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;ERROR CALL
;C(AC3)= SB RETURN ADDR
;C(AC1))= ACTUAL ADDR ON
;THE STACK
;POPJ' MUST JUMP TO 'ESP2X'
;ERROR CALL
;SHOULD NEVER HALT HERE
;ITERATE TEST ROUTINE

```

```

10084 ESPCL
01
02
03
04
05
06
07 02332 006120 ESP04: SETUP 100.
08 02533 000144 TAN024: JSR @ENTIN ;INITIALIZE TEST
09 02534 106470 EJSR ;ITERATION VALUE
10 02535 007562 RNDOT ;GENERATE RAND NUMBER
11 02536 000001 ;BETWEEN 1 AND 17
12 02537 000017 ;
13 02540 178400 SUB 3,3 ;
14 02541 058300 STA 3,TEM ;CLEAR LOCATION TO 0
15 02542 028277 LDA 1,80 ;
16 02543 124000 ADD 1,0 ;
17 02544 102010 BTO 0,0 ;BTO SETS ONE BIT IN
18 02545 105000 MOV 0,1 ;TEMP LOC 'TEM'
19 02546 126770 SNB 1,1 ;SNB MUST SKIP ON
20 ;NON-ZERO BIT IN 'TEM'
21 02547 100030 ENR036: XOP 0,0,0 ;ERROR CALL
22 02550 126110 BTZ 1,1 ;NON-ZERO BIT TO 0
23 ;
24 02551 006117 LAN024: JSR @ENTLO ;ITERATE TEST ROUTINE
25

```

```

10083 ESPCL
01
02
03
04
05
06
07
08 02462 006120 ESP03: SETUP 100.
09 02463 000144 TAN023: JSR @ENTIN ;INITIALIZE TEST
10 02464 105220 SUBZL 100. ;ITERATION VALUE
11 02465 105800 INC ;SET C(AC0)=1,C(AC1)=2 AND
12 02466 151400 INC ;C(AC2)=3 AND THEN 'SAVE'.
13 02467 004402 JSR *+2 ;
14 02470 000421 JMP ESP3 ;
15 02471 163710 SAVE 0 ;
16
17 02473 143710 MUL ;
18 02474 153710 DIV ;
19 02475 147710 MULS ;
20 02476 157710 DIVS ;
21 02477 147710 MULS ;
22 02500 137710 DIVX ;
23 02501 101020 MOVZ 0,0 ;RESET CARRY
24 02502 104210 UAD 0,1 ;
25 02503 130310 DSB 1,2 ;C(AC2) MUST BE 11
26 02504 050300 STA 2,TEM ;CHECK C(AC2) AT THE END
27 02505 102000 ADC 0,0 ;CHANGE C(AC0)
28 ;RTN' MUST RESTORE AC0,AC1,
29 ;AC2 AND CARRY
30 02507 100030 ENR030: XOP 0,0,0 ;ERROR CALL
31 02510 063077 HALT ;ERROR CALL
32 02511 176000 ESP3: ADC 3,3 ;
33 02512 163014 ADD# 3,0,SZR ;SHOULD NEVER COME HERE
34 ;
35 02513 100030 ENR031: XOP 0,0,0 ;ERROR CALL
36 02514 154010 ADI 3,3 ;
37 02515 166414 SUB# 3,1,SZR ;C(AC1) BAD
38 ;
39 02516 100030 ENR032: XOP 0,0,0 ;ERROR CALL
40 02517 114010 ADI 1,3 ;
41 02520 172414 SUB# 3,2,SZR ;C(AC2) BAD
42 ;
43 02521 100030 ENR033: XOP 0,0,0 ;ERROR CALL
44 02522 175002 MOV 3,5,SZC ;CARRY BAD
45 ;
46 02523 100030 ENR034: XOP 0,0,0 ;ERROR CALL
47 02524 030300 LDA 2,TEM ;
48 02525 173710 ADDI -11,2 ;C(AC2) BEFORE 'RTN' MUST BE =11
49 ;
50 02527 151004 MOV 2,2,SZR ;
51 ;
52 02530 100030 ENR035: XOP 0,0,0 ;ERROR CALL
53 ;
54 02531 006117 LAN023: JSR @ENTLO ;ITERATE TEST ROUTINE
55

```



```

10085 ESPCL
01
02 XERR ;FILL UPPER AND LOWER
03 JSR ;SCRATCH BUFFER AREA
04 XERR ;WITH (XERR)
05
06 SETUP ;INITIALIZE TEST
07 JSR ;ITERATION VALUE
08 PSBJ ;
09
10 AOC 1,1
11 MSP 1
12 ESP5A: AOC 0,0,0
13 ERROR
14 E\R037: XOP
15 HALT
16 POP
17 ELEM
18 SUR#
19 ERROR
20 E\R040: XOP
21 LOOP
22 E\R025: JSR
23
24

```

```

10086 ESPCL
01
02 ; TESTING 'XOP' INSTRUCTION
03
04
05
06 FILL XERR ;FILL UPPER AND LOWER
07 JSK XERR ;SCRATCH BUFFER AREA
08 XERR ;WITH (XERR)
09
10 SETUP 100. ;INITIALIZE TEST
11 JSR ;INITIALIZE TEST
12 LDA 0,STKB ;ITERATION VALUE
13 STA 0,SP ;RESET THE SP
14 ADDI 5,0 ;
15
16 STA 0,SL ;**
17 LDA 2,HIGRF ;SETUP 'XOP' ORIGIN
18 STA 2,XORG ;TABLE ADDR IN LOC. 44
19 ELEM 0,ESP6 ;SET UP 'ESP6' AS RETURN
20
21 STA 0,0,2 ;ADDR AFTER EXECUTING 'XOP'
22 XOP 0,1,0
23 HALTA 0 ;***XOP* ABOVE FAILED TO EXECUTE PROPERLY
24 JMP *-1 ;**FATAL ERROR HALT NO RECOVERY
25 ;**NO GO
26 ;**NOTE:
27 ;** SUCCESSFUL OPERATION OF THE "XOP" INSTRUCTION IS
28 ;** REQUIRED DUE TO THE USE OF THE "XOP" INSTRUCTION
29 ;** IN LINKING THE ERROR REPORT UTILITIES. IT IS SUGGESTED
30 ;** THAT THE USER RUN THE BASIC ECLIPSE DIAGNOSTICS.
31 ORIGIN
32
33 LOOP
34 ESP6: JSR ;**
35 LAM026: JSR ;**
36 LDA 0,ESP6-1
37 STA 0,XORG
38 INSTK
39 JSR ;**INITIALIZE SYS STACK

```



```

10089 ESPCL      LEFC1:  ELEF2  1,1
01              SETUP  100.
02              JSR    BENTIN ;INITIALIZE TEST
03 02654 006120 T\N031: JSR    ;ITERATION VALUE
04 02655 000144 LFC1:   RAND
05              JSR BENTRA      ;C(ACO)=RANDOM #
06 02656 006121      ANDI 77777,0
07 02657 143770      JSR  +1
08 02658 077777      ADDI 9,,3
09 02661 004801      ADUL# 0,3,SZC
10 02662 177770      JMP  LFC1
11 02664 117112      ADD  0,3
12 02665 000771      STA  0,,4
13 02666 117000      MOV  3,1*1&3
14 02667 040404      ADC  1,1
15 02670 171000      ELEF 1,0,1
16 02671 126000      SUB# 1,1*1&3,SZR
17 02672 166470      ERROR
18 02673 000000      XOP  0,0,0
19 02674 132414      LOOP
20 02675 100030 E\N031: JSR    ;ITERATE TEST ROUTINE
21 02676 006117 L\N031: JSR

```

```

10090 ESPCL      LEFC3:  ELEF2  3,3
01              SETUP  100.
02              JSR    BENTIN ;INITIALIZE TEST
03 02677 006120 T\N032: JSR    ;ITERATION VALUE
04 02700 000144 LFC3:   RAND
05              JSR BENTRA      ;C(ACO)=RANDOM #
06 02701 006121      ANDI 77777,0
07 02702 143770      JSR  +1
08 02704 077777      ADDI 9,,3
09 02705 177770      ADUL# 0,3,SZC
10 02707 117112      JMP  LFC3
11 02710 000771      ADD  0,3
12 02711 117000      STA  0,,4
13 02712 040404      MOV  3,3*1&3
14 02713 161000      ADC  3,3
15 02714 176000      ELEF 3,0,1
16 02715 176470      SUB# 3,3*1&3,SZR
17 02717 162414      ERROR
18 02720 100030 E\N044: XOP  0,0,0
19 02721 006117 L\N032: JSR

```

```

10092 ESPCL
01
02 LEF00: SETUP 100.
03 T\N033: JSR @ENTIN ;INITIALIZE TEST
04 02743 006120 ;ITERATION VALUE
05 LFE0: RAND
06 JSR @ENTRA ;C(AC0)=RANDOM #
07 MOV 0,3 ;AC3=RNOM #
08 RAND ;C(AC0)=RANDOM #
09 JSR @ENTRA ;C(AC0)=RANDOM #
10 02750 143770 ANDI 77777,0
11 ADDL# 0,3,SZC
12 02752 117112 JMP LFE0 ;AC0=RNOM # (OFFSET)
13 02753 00772 STA 0,-*4 ;AC1=OFFSET+C(AC3)
14 02754 040404 MOV 3,1 ;AFTER ELEF,
15 02755 165000 ADD 0,1 ;AC3 MUST BE EO. TO
16 02756 107000 ELEF 2,0,3 ;OFFSET+C(AC3)
17 02757 173270 SUR# 2,1,SZR
18 02761 146414 ERROR
19 02762 100030 E\N046: XOP 0,0,0 ;ERROR CALL
20 02763 006117 L\N033: JSR @ENTLO ;ITERATE TEST ROUTINE
21
22 LOOP
23

```

```

10091 ESPCL
01
02 LEF00: SETUP 100.
03 T\N033: JSR @ENTIN ;INITIALIZE TEST
04 02723 000144 ;ITERATION VALUE
05 LFE0: RAND
06 JSR @ENTRA ;C(AC0)=RANDOM #
07 MOV 0,2 ;AC2=RNOM #
08 RAND ;C(AC0)=RANDOM #
09 JSR @ENTRA ;C(AC0)=RANDOM #
10 02727 143770 ANDI 77777,0
11 ADDL# 0,2,SZC
12 02731 113112 JMP LFE0 ;AC0=RNOM # (OFFSET)
13 02732 00772 STA 0,-*4 ;AC1=OFFSET+C(AC2)
14 02733 040404 MOV 2,1 ;AFTER ELEF,
15 02734 145000 ADD 0,1 ;AC3 MUST BE EO. TO
16 02735 107000 ELEF 3,0,2 ;OFFSET+C(AC2)
17 02736 177070 SUR# 3,1,SZR
18 02740 166414 ERROR
19 02741 100030 E\N045: XOP 0,0,0 ;ERROR CALL
20 02742 006117 L\N033: JSR @ENTLO ;ITERATE TEST ROUTINE
21
22 LOOP
23

```

10093 ESPCL

```
01
02
03 ; NOW, CHECK ELEF WITH SINGLE INDIRECT ADDRESSING
04 ;
05
06 LEF0: SETUP 100.
07 02764 006120 JSR @ENTIN ;INITIALIZE TEST
08 02765 006144 LFF0: 100.
09 RNDADR ONE,TSSTOP
10 RAND
11 02766 006121 JSR @ENTRA ;C(AC0)=RANDOM #
12 02767 006122 JSR @RNADR ;GET ADDRESS IN THE RANGE
13 02770 00276 ONE ;C(ONE) AND C(TSTOP)
14 02771 00063 TSSTOP
15 02772 111000 MOV 0,2
16 02773 025000 LDA 1,0,2
17 02774 125112 MOV# 1,1,SZC
18 02775 000771 JMP LFF0
19 02776 040245 STA 0,LEFLOC
20 02777 026245 LDA 1,LEFLOC
21 03000 163770 ADDI 100000,0
22 100000
23 03002 040403 STA 0,+.3
24 03003 102000 AOC 0,0
25 03004 162070 ELEF 0,0,0
26 100000
27 03006 106414 SUB# 0,1,SZR
28 ERROR
29 03007 100030 E\R047: XOP ;ACO MUST BE = C(OFFSET)
30 03008 102000 ERROR CALL
31 03010 006117 L\N035: JSR ;ERROR CALL
32 ;ITERATE TEST ROUTINE
```

10094 ESPCL

```
01
02
03 ; NOW IS THE TIME TO CHECK MULTI-INDIRECT ELEF
04 ;
05
06 LEF0: SETUP 100.
07 03011 006120 JSR @ENTIN ;INITIALIZE TEST
08 03012 000144 LFF0: 100.
09 RAND
10 03013 006121 JSR @ENTRA ;C(AC0)=RANDOM #
11 03014 143770 ANDI 77777,0 ;RNDM ADDR IS SAVED IN
12 03016 040420 STA 0,LEF0+1 ;ACO AND (LEF0+1)
13 03017 004401 JSR +1 ;AC3=ADDR LEF0+1
14 03020 177770 ADDI +13,+.3 ;**
15 100000,3 ;ADD INDIRECT BIT
16 03022 117770 IORI
17 03024 054245 STA 3,LEFLOC ;SAVE @LEF0 IN LEFLOC
18 100000 ADI 1,3 ;SAVE @(LEF0+1) IN LEFLC
19 03025 114010 STA 3,LEFLC
20 03026 054246 ELEF 2,@LEFLC,0 ;MULTI-INDIRECT ELEF
21 03027 172070 SUB# 2,0,SZR ;SHOULD LOAD C(LEF0+1)
22 100245 ERROR ;IN AC3 WHICH IS =RNDM #
23 03031 142414 XOP 0,0,0 ;ERROR CALL
24 03032 100030 E\R050: LOP
25 03033 006117 L\N036: JSR @ENTLO ;ITERATE TEST ROUTINE
26 03034 000403 JMP +.3
27 03035 100246 LFF0: @LEFLC
28 03036 000080 0
29
30
31
32
```

```

10095 ESPCL
01
02 LEFG1: SETUP 100.
03 03037 006120 T\N037: JSR @ENTIN ;INITIALIZE TEST
04 03040 000144 RANO ;ITERATION VALUE
05
06 03041 006121 JSR @ENTRA ;C(AC0)=RANDOM.#
07 03042 143770 ANDI 77777,0 ;AC0=RANDOM ADDR
08 077777
09 03044 040417 STA 0,LF61+1 ;SAVE PNDM ADDR IN LFG1+1
10 03045 004401 JSR .+1 ;
11 03046 177770 ADDI +13.,3 ;**
12
13 03050 117770 LORI 100000,3 ;ADD INDIRECT BIT
14
15 03052 054246 STA 3,LEFLC ;SAVE @(LFG1+1) IN LEFLC
16 03053 162470 ELEF 0,06.,1 ;**DO MULTI INDIRECT
17
18 03055 024406 LDA 1,LF61+1 ;
19 03056 106414 SUB# 0,1,8ZR ;
20 ERROR
21 03057 100030 E\N051: YOP 0,0,0 ;ERROR CALL
22 LOOP
23 03060 006117 L\N037: JSR @ENTLO ;ITERATE TEST ROUTINE
24 03061 000403 JMP .+3 ;
25 03062 100246 LF61: @LEFLC
26 03063 000000
27

```

```

10096 ESPCL
01
02 LEFG2: ELEF4 2,2
03 SETUP 100.
04 03064 006120 T\N040: JSR @ENTIN ;INITIALIZE TEST
05 03065 000144 RANO ;ITERATION VALUE
06
07 03066 006121 JSR @ENTRA ;C(AC0)=RANDOM.#
08 03067 143770 ANDI 77777,0 ;AC0=RANDOM ADDR
09
10 03071 040432 STA 0,LF62+1 ;SAVE RNDM ADDR IN LFG2+1
11 03072 004401 JSR .+1 ;
12 03073 177770 ADDI +23.,3 ;**
13
14 03075 117770 LORI 100000,3 ;ADD INDIRECT BIT
15
16 03077 054245 STA 3,LEFLC ;SAVE @LF62 IN LEFLC
17 03100 114010 ADI 1,3 ;
18 03101 054246 STA 3,LEFLC ;SAVE @(LF62+1) IN LEFLC
19 03102 143770 ANDI 1777,0 ;
20
21 03104 111000 MOV 0,2 ;THIS SETS UP FOR INDEX
22 03105 166070 ELEF 1,LEFLC,0 ;MODE 2 FOR MULTI-INDIRECT
23
24 03107 106400 SUB 0,1 ;TEST
25 03110 107770 LORI 100000,1 ;ADD THE INDIRECT BIT
26
27 03112 044402 STA 1.,+2 ;
28 03113 163070 ELEF 0,80,2 ;
29
30 03115 024406 LDA 1,LF62+1 ;
31 03116 106414 SUB# 0,1,8ZR ;
32 ERROR
33 03117 100030 E\N052: YOP 0,0,0 ;ERROR CALL
34 LOOP
35 03120 006117 L\N040: JSR @ENTLO ;ITERATE TEST ROUTINE
36 03121 000403 JMP .+3 ;
37 03122 100246 LF62: @LEFLC
38 03123 000000

```



```

10099 ESPCL
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
; TESTING "ELDA" - LONG LDA INSTRUCTION
;
LDA1: ELDA1 1.0,SNC,ELDA
ELEF3 1.0,SNC,ELDA
SETUP 100.
04 03217 004120 T\N043: JSR @ENTIN ;INITIALIZE TEST
05 03218 000276 ONE ;ITERATION VALUE
06 03219 000063 RNDADR ONE,TSTTOP
07 RAND
08 03212 006121 JSR @ENTRA ;C(AC0)=RANDOM #
09 03213 006122 JSR @ENTRA ;GET ADDRESS IN THE RANGE
10 03214 000276 ONE ;C(ONE) AND C(TSTTOP)
11 03215 000063 TSTTOP
12 03216 105000 MOV 0.1 ;
13 03217 152040 ADCO 1+183.1+183 ; ALL AC'S ARE SET TO 17777
14 03218 006122 JSR @ENTRA 1+183.1+283 ;
15 03219 000276 ONE 1+283.1+383 ;
16 03215 000063 TSTTOP 1,+3 ;
17 03216 105000 MOV 0.1 ;
18 03217 152040 ADCO 1+183.1+183 ;
19 03220 155000 MOV 1+283.1+283 ;
20 03221 161000 MOV 1+283.1+383 ;
21 03222 044403 STA 1,+3 ;
22 03223 126000 ADC 1.1 ;
23 03224 126070 ELDA 1.0,0 ;ELDA MUST NOT CHANGE ANY AC'S
24 03226 125003 MOV 1.1,SNC ;
25 03227 100030 E\N055: XOP 0,0,0 ;ERROR CALL
26 03230 150014 COM# 1+183.1+183,SZR ;
27 03231 100030 E\N056: XOP 0,0,0 ;ERROR CALL
28 03232 174014 COM# 1+283.1+283,SZR ;
29 03233 100030 E\N057: XOP 0,0,0 ;ERROR CALL
30 03234 100014 COM# 1+383.1+383,SZR ;
31 03235 100030 E\N060: XOP 0,0,0 ;ERROR CALL
32 03236 006117 L\N043: JSR @ENTLO ;ITERATE TEST ROUTINE
33
34

```

```

10100 ESPCL
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
LDA1: ELDA1 2.2,SZC,ELDA
ELEF3 2.2,SZC,ELDA
SETUP 100.
04 03237 004120 T\N044: JSR @ENTIN ;INITIALIZE TEST
05 03240 000144 ONE ;ITERATION VALUE
06 RNDADR ONE,TSTTOP
07 RAND
08 03241 006121 JSR @ENTRA ;C(AC0)=RANDOM #
09 03242 006122 JSR @ENTRA ;GET ADDRESS IN THE RANGE
10 03243 000276 ONE ;C(ONE) AND C(TSTTOP)
11 03244 000063 TSTTOP
12 03245 111000 MOV 0.2 ;
13 03246 176020 ADCZ 2+183.2+183 ; ALL AC'S ARE SET TO 17777
14 03247 161000 MOV 2+183.2+283 ;
15 03250 105000 MOV 2+283.2+383 ;
16 03251 050403 STA 2,+5 ;
17 03252 152000 ADC 2.2 ;
18 03253 132070 ELDA 2.0,0 ;ELDA MUST NOT CHANGE ANY AC'S
19 03255 151002 MOV 2.2,SZC ;
20 03256 100030 E\N061: XOP 0,0,0 ;ERROR CALL
21 03257 174014 COM# 2+183.2+183,SZR ;
22 03260 100030 E\N062: XOP 0,0,0 ;ERROR CALL
23 03261 100014 COM# 2+283.2+283,SZR ;
24 03262 100030 E\N063: XOP 0,0,0 ;ERROR CALL
25 03263 124014 COM# 2+383.2+383,SZR ;
26 03264 100030 E\N064: XOP 0,0,0 ;ERROR CALL
27 03265 006117 L\N044: JSR @ENTLO ;ITERATE TEST ROUTINE
28
29
30
31
32
33
34

```



```

10101 ESPCL
01
02 LOAD0: SETUP 100.
03 03266 006120 T\N045: JSR @ENTIN ;INITIALIZE TEST
04 03267 000144 100. ;ITERATION VALUE
05 RNDADR ONE,TSITOP
06 RAND
07 03270 006121 ;C(AC0)=RANDOM #
08 03271 006122 JSR @ENTRA ;SET ADDRESS IN THE RANGE
09 03272 000276 JSR @RNADR ;C(ONE) AND C(TSITOP)
10 03273 000063 ONE
11 03274 040404 STA 0.+.4 ;SAVE THIS ADDR IN AC2
12 03275 111000 MOV 0.2 ;AC1 = C(RNDM ADDR)
13 03276 025000 LDA 1.0.2 ;EXECUTE ELDA
14 03277 136070 ELDA 3.0.0
15 03301 000000 SUB# 1.3.SZR
16 03302 100030 E\N065: XOP ERROR
17 03303 006117 L\N045: JSR @ENTLO ;ITERATE TEST ROUTINE
18 03304 000000 LOOP
19
20
21
22
23
24
25
26
27
28

```

```

10102 ESPCL
01
02 LOAD1: SETUP 100.
03 03304 006120 T\N046: JSR @ENTIN ;INITIALIZE TEST
04 03305 000144 100. ;ITERATION VALUE
05 RNDADR ONE,TSITOP
06 RAND
07 03306 006121 ;C(AC0)=RANDOM #
08 03307 006122 JSR @ENTRA ;SET ADDRESS IN THE RANGE
09 03310 000276 JSR @RNADR ;C(ONE) AND C(TSITOP)
10 03311 000063 ONE
11 03312 040413 STA 0.+.11. ;SAVE IN ELDA INSTRUCTION
12 03313 004401 JSR -.1 ;
13 03314 177770 ADDI 9.+.3 ;
14 03314 163000 ADD 3.0 ;ADD PC OF ELDA+1 TO
15 03317 024063 LDA 1,TSITOP ;RNDM ADDR
16 03320 106433 SUBZ# 0.1.SMC ;
17 03321 000765 JMP LAR1 ;ADDR> TSITOP, GET NEW ADDR
18 03322 115000 MOV 0.3 ;SET UP THIS ADDR IN AC3
19 03323 025400 LDA 1.0.3 ;AC1= C(RNDM ADDR+ PC OF (ELDA+1))
20 03324 132470 ELDA 2.0.1 ;AFTER ELDA,AC2 MUST BE=
21 03326 132414 SUB# 1.2.SZR ;C(RNDM ADDR+PC OF (ELDA+1))
22 03327 100030 E\N066: XOP ERROR
23 03327 100030 E\N066: XOP ERROR CALL
24 03330 006117 L\N046: JSR @ENTLO ;ITERATE TEST ROUTINE
25
26
27
28

```

10103 ESPCL

```
01
02
03 03331 006120 LOAD2: SETUP 100.
04 03332 000144 L\N047: JSR @ENTIN ;INITIALIZE TEST
05 RNDADR ONE,TSTTOP ;ITERATION VALUE
06 RAND
07 03333 006121 JSR @ENTRA ;C(AC0)=RANDOM #
08 03334 006122 JSR @RNDADR ;GET ADDRESS IN THE RANGE
09 03335 000276 ONE ;C(ONE) AND C(TSTTOP)
10 03336 000063 TSTTOP
11 03337 143370 HLV ;SAVE (RNDM ADDR/2) IN ELDA
12 03340 111000 MOV 0 ;RANO INDEX REGISTER AC2
13 03341 040405 STA 0.2 ;
14 03342 103000 ADD 0.0 ;AC0 =OFFSET+C(AC2)
15 03343 115000 MOV 0.3 ;SET AC3=OFFSET+C(AC2)
16 03344 025400 LDA 1.0,3 ;AC1 IS C(OFFSET+C(AC2))
17 03345 123070 ELDA 0.0,2 ;AC0 MUST BE EQ. TO AC1
18 000000
19 03347 106414 SUB# 0.1,SZR ;
20 ERROR
21 03350 100030 E\R067: XOP ;ERROR CALL
22 LOOP
23 03351 006117 L\N047: JSR @ENTLO ;ITERATE TEST ROUTINE
24
```

10104 ESPCL

```
01
02
03 03352 006120 LDAB3: SETUP 100.
04 03353 000144 L\N050: JSR @ENTIN ;INITIALIZE TEST
05 RNDADR ONE,TSTTOP ;ITERATION VALUE
06 RAND
07 03354 006121 JSR @ENTRA ;C(AC0)=RANDOM #
08 03355 006122 JSR @RNDADR ;GET ADDRESS IN THE RANGE
09 03356 000276 ONE ;C(ONE) AND C(TSTTOP)
10 03357 000063 TSTTOP
11 03360 143370 HLV ;SAVE (RNDM ADDR/2) IN ELDA
12 03361 115000 MOV 0.3 ;AND (RNDM ADDR/2)+1 IN
13 03362 040405 STA 0.0,+5 ;INDEX REGISTER AC3
14 03363 111140 MOVOL 0.2 ;
15 03364 025000 LDA 1.0,2 ;AC1=C(OFFSET+C(AC3))
16 03365 114010 ADI 1.3 ;AFTER ELDA,
17 03366 123070 ELDA 0.0,3 ;AC0 MUST BE EQ. TO
18 000000
19 03370 106414 SUB# 0.1,SZR ;C(OFFSET+C(AC3))
20 ERROR
21 03371 100030 E\R070: XOP ;ERROR CALL
22 LOOP
23 03372 006117 L\N050: JSR @ENTLO ;ITERATE TEST ROUTINE
```

10105 ESPCL

```

01
02
03
04
05
06
07
08 03373 006120
09 03374 000144
10 03375 024063
11 03376 162470
12 03400 000023
13 03400 106400
14 03401 044250
15
16
17 03402 006121
18 03403 006122
19 03404 000276
20 03405 000250
21 03406 176470
22 03410 000013
23 03410 117000
24 03411 035400
25 03412 024063
26 03413 166433
27 03414 000766
28 03415 103770
29
30 03417 040403
31 03420 025400
32 03421 132470
33
34 03423 132414
35
36 03424 100030
37
38 03425 006117
39

```

```

: CHECKING SINGLE LEVEL INDIRECT "ELDA"
:

```

```

LOAD1: SETUP 100.
L\N051: JSR @ENTRIN ;INITIALIZE TEST
100. ;ITERATION VALUE
LDA 1,TSTTOP
ELEF 0,LAC10-,,1 ;SAVE (TSTTOP-PC OF ELDA+1)
SUB 0,1 ;IN LDALC
STA 1,LDALC
LAC1: RNDADR ONE,LDALC
RAND
JSR @ENTRA ;C(AC0)=RANDOM #
@RNADR ;GET ADDRESS IN THE RANGE
ONE ;C(ONE) AND C(LDALC)
LDALC
ELEF 3,LAC10-,,1
ADD
LDA 0,3 ;AC3=OFFSET+(PC OF ELDA+1)
LDA 3,0,3 ;AC3=C(OFFSET+PC OF ELDA+1)
LDA 1,0,3 ;MAKE SURE THAT ADDR IN AC3
SUBZ# 3,1,SNC ;DOES NOT INDIRECT
JMP LAC1
IDRI 100000,0 ;ADD INDIRECT BIT FOR ELDA
STA 0,+,3
LDA 1,0,3
ELDA LAC10: ELDA 2,-30,1
SUB# 1,2,SZR ;C(C(OFFSET+PC OF ELDA+1))
ERROR
E\N071: XDP 0,0,0 ;ERROR CALL
LOOP
L\N051: JSR @ENTLDO ;ITERATE TEST ROUTINE

```

10106 ESPCL

```

01
02
03
04
05
06
07 03426 006120
08 03427 000144
09
10
11 03430 006121
12 03431 006122
13 03432 000132
14 03433 000123
15 03434 040251
16
17
18 03435 006121
19 03436 006122
20 03437 000124
21 03440 000133
22 03441 103240
23 03442 040252
24
25
26 03443 006121
27 03444 006122
28 03445 000132
29 03446 000123
30 03447 024251
31 03450 122415
32 03451 000772
33 03452 111000
34 03453 163270
35
36 03455 040410
37 03456 034252
38 03457 055000
39 03460 177220
40 03461 030251
41 03462 051400
42 03463 021000
43 03464 126070
44
45 03466 106414
46
47 03467 100030
48
49 03470 006117

```

```

: TESTS FOR MULTI-INDIRECT "ELDA"
:

```

```

LOAD0: SETUP 100.
L\N052: JSR @ENTIN ;INITIALIZE TEST
100. ;ITERATION VALUE
RND
JSR @ENTRA ;C(AC0)=RANDOM #
@RNADR ;GET A RANDOM ADDRESS IN
LOW# ;LOWER BUFFER
LORFU
STA 0,LDSV0 ;C(LDSV0)=ADDR FROM LOW BUFF
HIBFA
RAND
JSR @ENTRA ;C(AC0)=RANDOM #
@RNADR ;GET A RANDOM ADDRESS IN
HIG# ;THE UPPER BUFFER
HIFU
ADDR 0,0 ;C(LDSV1)=ADDR FROM HIGH BUFF
STA 0,LDSV1 ;WITH INDIRECT BIT SET
RND
JSR @ENTRA ;C(AC0)=RANDOM #
@RNADR ;GET A RANDOM ADDRESS IN
LOW# ;LOWER BUFFER
LDA 1,LDSV0 ;MAKE SURE RNDM ADDRESSES
SUB# 1,0,SNR ;FROM LOBUFF ARE DIFFERENT
LAD0 0,2 ;AC0=RNDM ADDR 3 FROM LOW
ADDI 100000,0 ;BUFF WITH INDIRECT BIT SET
STA 0,+,0 ;SAVE AC0 IN ELDA INSTRUCTION
LDA 3,LDSV1 ;STORE (@RNDM ADDR 2) IN
STA 3,0,2 ;(RNDM ADDR 3) AND
ADDZ# 3,3 ;(@RNDM ADDR 1) IN
LDA 2,LDSV0 ;(RNDM ADDR 2)
STA 2,0,3
LDA 0,0,2
ELDA 1,-30,0 ;AC1 MUST REEC(RNDM ADDR 1)
SUB# 0,1,SZR
ERROR
E\N072: XDP 0,0,0 ;ERROR CALL
LOOP
L\N052: JSR @ENTLDO ;ITERATE TEST ROUTINE

```

10107 ESPCL

```

01
02
03 03471 006120  LOAD1:  SETUP 100.  INITIALIZE TEST
04 03472 000124  T\N053: JSR 100.  @ENTIN  ;ITERATION VALUE
05
06
07 03473 006121  ;C(ACO)=RANDOM #
08 03474 006122  ;GET A RANDOM ADDRESS IN
09 03475 000124  HIGBF @RNADR  ;THE UPPER BUFFER
10 03476 000133  HIGBF
11 03477 040251  STA 0,LDSV0  ;C(LDSV0)=ADDR FROM HIGH BUFF
12
13
14 03500 006121  RAND
15 03501 006122  JSR @ENTRA  ;C(ACO)=RANDOM #
16 03502 000132  LMBF @RNADR  ;GET A RANDOM ADDRESS IN
17 03503 000123  HIGBF  ;THE UPPER BUFFER
18 03504 103240  ADDR 0,0  ;C(LDSV1)=ADDR FROM LOW BUFF WITH
19 03505 040252  STA 0,LDSV1  ;INDIRECT RIT SET
20
21
22 03506 006121  RAND
23 03507 006122  JSR @ENTRA  ;C(ACO)=RANDOM #
24 03510 000124  HIGBF @RNADR  ;GET A RANDOM ADDRESS IN
25 03511 000133  HIGBF  ;THE UPPER BUFFER
26 03512 024251  LDA 1,LDSV0  ;
27 03513 106415  SUB# 0,1,SNR  ;MAKE SURE THAT RNDM NDS.
28 03514 00772  JMP LAD10  ;FROM HIBUFF ARE DIFFERENT
29 03515 166470  ELEF 1,LAD10-,,1  ;
30
31 03517 111000  MOV 0,2  ;@ (RNDM ADDR 3-PC OF ELDA*1) IS
32 03520 122400  SUB 1,0  ;SAVED IN ELDA INSTRUCTION
33 03521 103240  ADDR 0,0  ;
34 03522 040410  STA 0,,*8.  ;
35 03523 034252  LDA 3,LDSV1  ;ALSO STORE @ (RNDM ADDR 2) IN
36 03524 055000  STA 3,0,2  ; (RNDM ADDR 3) AND
37 03525 177220  ADDR 3,3  ; (RNDM ADDR 1) IN
38 03526 030251  LDA 2,LDSV0  ;
39 03527 051400  LDA 2,0,3  ;
40 03530 021000  LDA 0,0,2  ;
41 03531 126470  ELDA 1,*0,1  ;
42
43 03533 106414  SUB# 0,1,SZR  ;C(RNDM ADDR 1)
44
45 03534 100030  ERROR  ;ERROR CALL
46
47 03535 006117  LAD10: LAD10:  ;ITERATE TEST ROUTINE
48

```

10108 ESPCL

```

01
02
03 03536 006120  LOAD2:  SETUP 2,LAR2  ;INITIALIZE TEST
04 03537 000124  T\N054: JSR 100.  @ENTIN  ;ITERATION VALUE
05
06
07 03540 006121  RAND
08 03541 006122  JSR @ENTRA  ;C(ACO)=RANDOM #
09 03542 000132  LMBF @RNADR  ;GET A RANDOM ADDRESS IN
10 03543 000123  HIGBF  ;THE UPPER BUFFER
11 03544 040251  STA 0,LDSV0  ;C(LDSV0)=ADDR FROM LOW BUFF
12
13
14 03545 006121  RAND
15 03546 006122  JSR @ENTRA  ;C(ACO)=RANDOM #
16 03547 000132  LMBF @RNADR  ;GET A RANDOM ADDRESS IN
17 03548 000123  HIGBF  ;THE UPPER BUFFER
18 03549 000123  HIGBF
19 03551 103240  ADDR 0,0  ;C(LDSV1)=ADDR FROM HIGH
20 03552 040252  STA 0,LDSV1  ;BUFF WITH INDIRECT RIT SET
21
22
23 03553 006121  RAND
24 03554 006122  JSR @ENTRA  ;C(ACO)=RANDOM #
25 03555 000124  HIGBF @RNADR  ;GET A RANDOM ADDRESS IN
26 03556 000133  HIGBF  ;THE UPPER BUFFER
27 03557 024252  LDA 1,LDSV1  ;MAKE SURE THAT RNDM NDS.
28 03560 127220  ADDR 1,1  ;FROM HIBUFF ARE DIFFERENT
29 03561 125415  SUB# 1,0,SNR  ;
30 03562 000771  JMP LAD2  ;
31 03563 111000  MOV 0,2  ;ALSO STORE @ (RNDM ADDR 2)
32 03564 034252  LDA 3,LDSV1  ;IN (RNDM ADDR 3) AND
33 03565 055000  STA 3,0,2  ; (RNDM ADDR 1) IN
34 03566 177220  ADDR 3,3  ; (RNDM ADDR 2)
35 03567 030251  LDA 2,LDSV0  ;
36 03570 051400  STA 2,0,3  ;
37 03571 025000  LDA 1,0,2  ;
38 03572 111222  MOVZR 0,2,SZC  ;ALSO STORE (RNDM ADDR 3)/2
39 03573 110010  ADI 1,2  ;*BIT 15 IN AC2
40 03574 143370  HLV 0  ;@ (RNDM ADDR 3)/2 IS SAVED
41 03575 103240  ADDR 0,0  ;IN ELDA INSTRUCTION
42 03576 040402  STA 0,,*2  ;
43 03577 123070  ELDA 0,*0,2  ;
44
45 03601 106414  SUB# 0,1,SZR  ;C(RNDM ADDR 1)
46
47 03602 100030  ERROR  ;ERROR CALL
48
49 03603 006117  LAD2: LAD2:  ;ITERATE TEST ROUTINE

```

```

10109 ESPCL
01 01 LAD3: ELD3: 3,LAD3
02 02 SETUP 100
03 03 JSR @ENTRIN ;INITIALIZE TEST
04 04 JSR @ENTRIN ;INITIALIZE TEST
05 05 JSR @ENTRIN ;INITIALIZE TEST
06 06 JSR @ENTRIN ;INITIALIZE TEST
07 07 JSR @ENTRIN ;INITIALIZE TEST
08 08 JSR @ENTRIN ;INITIALIZE TEST
09 09 JSR @ENTRIN ;INITIALIZE TEST
10 10 JSR @ENTRIN ;INITIALIZE TEST
11 11 JSR @ENTRIN ;INITIALIZE TEST
12 12 JSR @ENTRIN ;INITIALIZE TEST
13 13 JSR @ENTRIN ;INITIALIZE TEST
14 14 JSR @ENTRIN ;INITIALIZE TEST
15 15 JSR @ENTRIN ;INITIALIZE TEST
16 16 JSR @ENTRIN ;INITIALIZE TEST
17 17 JSR @ENTRIN ;INITIALIZE TEST
18 18 JSR @ENTRIN ;INITIALIZE TEST
19 19 JSR @ENTRIN ;INITIALIZE TEST
20 20 JSR @ENTRIN ;INITIALIZE TEST
21 21 JSR @ENTRIN ;INITIALIZE TEST
22 22 JSR @ENTRIN ;INITIALIZE TEST
23 23 JSR @ENTRIN ;INITIALIZE TEST
24 24 JSR @ENTRIN ;INITIALIZE TEST
25 25 JSR @ENTRIN ;INITIALIZE TEST
26 26 JSR @ENTRIN ;INITIALIZE TEST
27 27 JSR @ENTRIN ;INITIALIZE TEST
28 28 JSR @ENTRIN ;INITIALIZE TEST
29 29 JSR @ENTRIN ;INITIALIZE TEST
30 30 JSR @ENTRIN ;INITIALIZE TEST
31 31 JSR @ENTRIN ;INITIALIZE TEST
32 32 JSR @ENTRIN ;INITIALIZE TEST
33 33 JSR @ENTRIN ;INITIALIZE TEST
34 34 JSR @ENTRIN ;INITIALIZE TEST
35 35 JSR @ENTRIN ;INITIALIZE TEST
36 36 JSR @ENTRIN ;INITIALIZE TEST
37 37 JSR @ENTRIN ;INITIALIZE TEST
38 38 JSR @ENTRIN ;INITIALIZE TEST
39 39 JSR @ENTRIN ;INITIALIZE TEST
40 40 JSR @ENTRIN ;INITIALIZE TEST
41 41 JSR @ENTRIN ;INITIALIZE TEST
42 42 JSR @ENTRIN ;INITIALIZE TEST
43 43 JSR @ENTRIN ;INITIALIZE TEST
44 44 JSR @ENTRIN ;INITIALIZE TEST
45 45 JSR @ENTRIN ;INITIALIZE TEST
46 46 JSR @ENTRIN ;INITIALIZE TEST
47 47 JSR @ENTRIN ;INITIALIZE TEST
48 48 JSR @ENTRIN ;INITIALIZE TEST
49 49 JSR @ENTRIN ;INITIALIZE TEST

10110 FSPCL
01 01 STAA0: ESTA1 0,0,SNC,LOBFAD
02 02 STAA0: ESTA1 0,0,SNC,LOBFAD
03 03 STAA0: ESTA1 0,0,SNC,LOBFAD
04 04 STAA0: ESTA1 0,0,SNC,LOBFAD
05 05 STAA0: ESTA1 0,0,SNC,LOBFAD
06 06 STAA0: ESTA1 0,0,SNC,LOBFAD
07 07 STAA0: ESTA1 0,0,SNC,LOBFAD
08 08 STAA0: ESTA1 0,0,SNC,LOBFAD
09 09 STAA0: ESTA1 0,0,SNC,LOBFAD
10 10 STAA0: ESTA1 0,0,SNC,LOBFAD
11 11 STAA0: ESTA1 0,0,SNC,LOBFAD
12 12 STAA0: ESTA1 0,0,SNC,LOBFAD
13 13 STAA0: ESTA1 0,0,SNC,LOBFAD
14 14 STAA0: ESTA1 0,0,SNC,LOBFAD
15 15 STAA0: ESTA1 0,0,SNC,LOBFAD
16 16 STAA0: ESTA1 0,0,SNC,LOBFAD
17 17 STAA0: ESTA1 0,0,SNC,LOBFAD
18 18 STAA0: ESTA1 0,0,SNC,LOBFAD
19 19 STAA0: ESTA1 0,0,SNC,LOBFAD
20 20 STAA0: ESTA1 0,0,SNC,LOBFAD
21 21 STAA0: ESTA1 0,0,SNC,LOBFAD
22 22 STAA0: ESTA1 0,0,SNC,LOBFAD
23 23 STAA0: ESTA1 0,0,SNC,LOBFAD
24 24 STAA0: ESTA1 0,0,SNC,LOBFAD
25 25 STAA0: ESTA1 0,0,SNC,LOBFAD
26 26 STAA0: ESTA1 0,0,SNC,LOBFAD
27 27 STAA0: ESTA1 0,0,SNC,LOBFAD
28 28 STAA0: ESTA1 0,0,SNC,LOBFAD
29 29 STAA0: ESTA1 0,0,SNC,LOBFAD
30 30 STAA0: ESTA1 0,0,SNC,LOBFAD
31 31 STAA0: ESTA1 0,0,SNC,LOBFAD
32 32 STAA0: ESTA1 0,0,SNC,LOBFAD
33 33 STAA0: ESTA1 0,0,SNC,LOBFAD
34 34 STAA0: ESTA1 0,0,SNC,LOBFAD
35 35 STAA0: ESTA1 0,0,SNC,LOBFAD
36 36 STAA0: ESTA1 0,0,SNC,LOBFAD
37 37 STAA0: ESTA1 0,0,SNC,LOBFAD
38 38 STAA0: ESTA1 0,0,SNC,LOBFAD
39 39 STAA0: ESTA1 0,0,SNC,LOBFAD
40 40 STAA0: ESTA1 0,0,SNC,LOBFAD
41 41 STAA0: ESTA1 0,0,SNC,LOBFAD
42 42 STAA0: ESTA1 0,0,SNC,LOBFAD
43 43 STAA0: ESTA1 0,0,SNC,LOBFAD
44 44 STAA0: ESTA1 0,0,SNC,LOBFAD
45 45 STAA0: ESTA1 0,0,SNC,LOBFAD

```

```

10111 ESPCL
01
02
03
04 03705 006120 T=057: JSR          SETUP          100.
05 03706 000144 T=057: JSR          @ENTIN      #INITIALIZE TEST
06                                #ITERATION VALUE
07                                #
08 03707 006121 JSR @ENTRA      #C(AC0)=RANDOM #
09 03710 006122 JSR @RNADR      #GET A RANDOM ADDRESS IN
10 03711 000124 HIBF          #THE UPPER BUFFER
11 03712 000133 HIBFU          #
12 03713 040253 STA          0,STLCO
13 03714 040404 STA          0,.*4
14                                #
15 03715 006121 JSR @ENTRA      #C(AC0)=RANDOM #
16 03716 040254 STA          0,STLC1
17 03717 142070 ESTA          0,0,0
18                                #
19 03721 026253 LDA          1,@STLCO
20 03722 030254 LDA          2,STLC1
21 03723 132414 SUR#         1,2,SZR
22                                #
23 03724 100030 E=103: XOP      #ERROR CALL
24                                #
25 03725 006117 L=057: JSR      #ENTLO #ITERATE TEST ROUTINE
26
27
28
10112 ESPCL
01
02
03
04 03726 006120 T=060: JSR          SETUP          100.
05 03727 000144 T=060: JSR          @ENTIN      #INITIALIZE TEST
06                                #ITERATION VALUE
07                                #
08 03730 006121 JSR @ENTRA      #C(AC0)=RANDOM #
09 03731 006122 JSR @RNADR      #GET A RANDOM ADDRESS IN
10 03732 000124 HIBF          #THE UPPER BUFFER
11 03733 000133 HIBFU          #
12 03734 040253 STA          0,STLCO
13 03735 166470 STA          1,SABI-.1
14                                #
15 03737 122400 SUB          1,0
16 03740 040404 STA          0,SABI+1
17                                #
18 03741 006121 JSR @ENTRA      #C(AC0)=RANDOM #
19 03742 105000 MOY          0,1
20 03743 146470 SABI: ESTA      1,0,1
21                                #
22 03745 032253 LDA          2,@STLCO
23 03746 112414 SUR#         0,2,SZR
24                                #
25 03747 100030 E=104: XOP      #ERROR CALL
26                                #
27 03750 006117 L=060: JSR      #ENTLO #ITERATE TEST ROUTINE
28

```

10113 FSPCL

```

01
02
03
04
05 03751 006120 T\N061: JSR @ENTIN ;INITIALIZE TEST
06 03752 000144 HIGBF ;ITERATION VALUE
07 HIGFAD
08 RAND
09 03753 006121 JSR @ENTRA ;C(AC0)=RANDOM #
10 03754 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
11 03755 000124 HIGBF ;THE UPPER BUFFER
12 03756 000133 HIGFU
13 03757 040253 STA 0,STLCO ;C(STLCO)=RNDM ADDR
14 RAND
15 03760 006121 JSR @ENTRA ;C(AC0)=RANDOM #
16 03761 105000 MOV 0,1 ;AC1= RNDM DATA
17 03762 020253 LDA 0,STLCO ;
18 03763 111222 MOVZR 0,2,SZC ;AC2=(RNDM ADDR/2)+BIT 15
19 03764 110010 ADI 1,2 ;
20 03765 143370 HLV 0 ;AC0=RNDM ADDR/2
21 03766 040402 STA 0,+2 ;
22 03767 147070 ESTA 1,0,2 ;
23 LDA 0,+STLCO ;RNDM ADDR MUST CONTAIN
24 03771 022253 SUB# 0,1,SZR ;RANDOM DATA(=C(AC1))
25 03772 106414 ERROR
26 03773 100030 EVR105: XDP 0,0,0 ;ERROR CALL
27 03774 006117 L\N061: JSR @ENTLO ;ITERATE TEST ROUTINE
28
29

```

10114 FSPCL

```

01
02
03
04
05 03775 006120 T\N062: JSR @ENTIN ;INITIALIZE TEST
06 03776 000144 LOBFAD ;ITERATION VALUE
07 RAND
08 JSR @ENTRA ;C(AC0)=RANDOM #
09 03777 006121 JSR @RNADR ;GET A RANDOM ADDRESS IN
10 04000 006122 LOBFU ;LOWER BUFFER
11 04001 000132 STA 0,STLCO ;C(STLCO)=RNDM ADDR
12 04002 000123 RAND
13 04003 040253 JSR @ENTRA ;C(AC0)=RANDOM #
14 04004 006121 MOV 0,1 ;AC1= RNDM DATA
15 04005 105000 LDA 0,STLCO ;
16 04006 020253 MOVZR 0,3,SZC ;AC3=(RNDM ADDR/2)+BIT 15
17 04007 115222 ADI 1,3 ;
18 04010 114010 HLV 0 ;AC0=RNDM ADDR/2
19 04011 143370 STA 0,+2 ;
20 04012 040402 ESTA 1,0,3 ;
21 04013 147470 LDA 0,+STLCO ;RNDM ADDR MUST CONTAIN
22 04015 022253 SUB# 0,1,SZR ;RANDOM DATA(=C(AC1))
23 ERROR
24 04017 100030 EVR106: XDP 0,0,0 ;ERROR CALL
25 04020 006117 L\N062: JSR @ENTLO ;ITERATE TEST ROUTINE
26
27
28
29

```

!0115 ESPCL

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53

```
! NOW CHECK "ESTA" SINGLE LEVEL INDIRECT  
STAC2:  ESTA4  LOBFAD,HIBFAD,SAC20,2  
ESTAS3  LOBFAD,HIBFAD,SAC20  
        SETUP 100.  
        JSR @ENTIN ;INITIALIZE TEST  
        LOBFAD 100.  
        RAND @ENTIN ;ITERATION VALUE  
        JSR @ENTRA ;C(ACO)=RANDOM #  
        LOBF  @RNADR ;GET A RANDOM ADDRESS IN  
        LOBFU ;LOWER BUFFER  
        STA 0,STLCO ;C(STLCO)=RNDM ADDR 1  
        RAND  
        JSR @ENTRA ;C(ACO)=RANDOM #  
        LOBF  @RNADR ;GET A RANDOM ADDRESS IN  
        LOBFU ;LOWER BUFFER  
        STA 0,STLCO ;C(STLCO)=@RNDM ADDR 2)  
        RAND  
        JSR @ENTRA ;C(ACO)=RANDOM #  
        LOBF  @RNADR ;GET A RANDOM ADDRESS IN  
        LOBFU ;LOWER BUFFER  
        LDA 1,STLCO ;  
        JMP SAO 1,0,SNR ;  
        MOV SAO 0,2 ;ACO=AC2=RNDM ADDR 3  
        ADDR 0,0 ;@RNDM ADDR 3) IS SAVED  
        STA 0,+,9. ;IN ESTA INSTRUCTION  
        LDA 3,STLCO ;STORE (@RNDM ADDR 2) IN  
        ADDR 3,3 ;@RNDM ADDR 3) AND  
        LDA 2,STLCO ;@RNDM ADDR 1) IN  
        STA 2,0,5 ;@RNDM ADDR 2)  
        RAND  
        JSR @ENTRA ;C(ACO)=RANDOM #  
        MOV 0,2 ;ACO=AC2=RNDM DATA  
        ESTA 2,@0,0 ;STORE RANDOM DATA IN  
        LOA 1,@STLCO ;(RNDM ADDR 1)  
        SUB# 1,0,5ZR ;AC1 MUST BE# RANDOM DATA  
        ERROR  
        XOP 0,0,0 ;ERROR CALL  
        LOOP  
        @ENTLO ;ITERATE TEST ROUTINE  
        JSR @ENTLO ;ITERATE TEST ROUTINE
```

!0116 ESPCL

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53

```
! CHECKING MULTI-INDIRECT "ESTA"  
STAC2:  ESTA4  LOBFAD,HIBFAD,SAC20,2  
ESTAS3  LOBFAD,HIBFAD,SAC20  
        SETUP 100.  
        JSR @ENTIN ;INITIALIZE TEST  
        LOBFAD 100.  
        RAND @ENTIN ;ITERATION VALUE  
        JSR @ENTRA ;C(ACO)=RANDOM #  
        LOBF  @RNADR ;GET A RANDOM ADDRESS IN  
        LOBFU ;LOWER BUFFER  
        STA 0,STLCO ;C(STLCO)=RNDM ADDR 1  
        RAND  
        JSR @ENTRA ;C(ACO)=RANDOM #  
        LOBF  @RNADR ;GET A RANDOM ADDRESS IN  
        LOBFU ;LOWER BUFFER  
        LDA 1,STLCO ;  
        JMP SAO 1,0,SNR ;  
        MOV SAO 0,2 ;ACO=AC2=RNDM ADDR 3  
        ADDR 0,0 ;@RNDM ADDR 3) IS SAVED  
        STA 0,+,9. ;IN ESTA INSTRUCTION  
        LDA 3,STLCO ;STORE (@RNDM ADDR 2) IN  
        ADDR 3,3 ;@RNDM ADDR 3) AND  
        LDA 2,STLCO ;@RNDM ADDR 1) IN  
        STA 2,0,5 ;@RNDM ADDR 2)  
        RAND  
        JSR @ENTRA ;C(ACO)=RANDOM #  
        MOV 0,2 ;ACO=AC2=RNDM DATA  
        ESTA 2,@0,0 ;STORE RANDOM DATA IN  
        LOA 1,@STLCO ;(RNDM ADDR 1)  
        SUB# 1,0,5ZR ;AC1 MUST BE# RANDOM DATA  
        ERROR  
        XOP 0,0,0 ;ERROR CALL  
        LOOP  
        @ENTLO ;ITERATE TEST ROUTINE  
        JSR @ENTLO ;ITERATE TEST ROUTINE
```



```

:0117 ESPCL
01
02
03
04 04125 006120 T\N065:
05 04126 000144
06
07
08 04127 006121
09 04130 006122
10 04131 000124
11 04132 000133
12 04133 040253
13
14
15 04134 006121
16 04135 006122
17 04136 000132
18 04137 000123
19 04140 103240
20 04141 040254
21
22
23 04142 006121
24 04143 006122
25 04144 000124
26 04145 000133
27 04146 024253
28 04147 122415
29 04150 000772
30 04151 166470
31
32 04153 111000
33 04154 122400
34 04155 103240
35 04156 040411
36 04157 034254
37 04160 055000
38 04161 177220
39 04162 030253
40 04163 051400
41
42 04164 006121
43 04165 105000
44 04166 166470 S401:
45
46 04170 032253
47 04171 162414
48
49 04172 100030 E\N111:
50
51 04173 006117 L\N065:
52
53
:0118 ESPCL
01
02
03
04 04174 006120 T\N066:
05 04175 000144
06
07
08 04176 006121
09 04177 040255
10
11
12 04200 006121
13 04201 006122
14 04202 000132
15 04203 000123
16 04204 040253
17
18
19 04205 006121
20 04206 006122
21 04207 000132
22 04210 000123
23 04211 024253
24 04212 106415
25 04213 000772
26 04214 103240
27 04215 040254
28
29
30 04216 006121
31 04217 006122
32 04220 000124
33 04221 000133
34 04222 111000
35 04223 034254
36 04224 055000
37 04225 177220
38 04226 030253
39 04227 051400
40 04230 111222
41 04231 110010
42 04232 143370
43 04233 103240
44 04234 040403
45 04235 024255
46 04236 147070
47
48 04240 022253
49 04241 106414
50
51 04242 100030 E\N112:
52
53 04243 006117 L\N066:

```

```

:0117 ESPCL
01
02
03
04 04125 006120 T\N065:
05 04126 000144
06
07
08 04127 006121
09 04130 006122
10 04131 000124
11 04132 000133
12 04133 040253
13
14
15 04134 006121
16 04135 006122
17 04136 000132
18 04137 000123
19 04140 103240
20 04141 040254
21
22
23 04142 006121
24 04143 006122
25 04144 000124
26 04145 000133
27 04146 024253
28 04147 122415
29 04150 000772
30 04151 166470
31
32 04153 111000
33 04154 122400
34 04155 103240
35 04156 040411
36 04157 034254
37 04160 055000
38 04161 177220
39 04162 030253
40 04163 051400
41
42 04164 006121
43 04165 105000
44 04166 166470 S401:
45
46 04170 032253
47 04171 162414
48
49 04172 100030 E\N111:
50
51 04173 006117 L\N065:
52
53

```

```

:0118 ESPCL
01
02
03
04 04174 006120 T\N066:
05 04175 000144
06
07
08 04176 006121
09 04177 040255
10
11
12 04200 006121
13 04201 006122
14 04202 000132
15 04203 000123
16 04204 040253
17
18
19 04205 006121
20 04206 006122
21 04207 000132
22 04210 000123
23 04211 024253
24 04212 106415
25 04213 000772
26 04214 103240
27 04215 040254
28
29
30 04216 006121
31 04217 006122
32 04220 000124
33 04221 000133
34 04222 111000
35 04223 034254
36 04224 055000
37 04225 177220
38 04226 030253
39 04227 051400
40 04230 111222
41 04231 110010
42 04232 143370
43 04233 103240
44 04234 040403
45 04235 024255
46 04236 147070
47
48 04240 022253
49 04241 106414
50
51 04242 100030 E\N112:
52
53 04243 006117 L\N066:

```

```

:0117 ESPCL
01
02
03
04 04125 006120 T\N065:
05 04126 000144
06
07
08 04127 006121
09 04130 006122
10 04131 000124
11 04132 000133
12 04133 040253
13
14
15 04134 006121
16 04135 006122
17 04136 000132
18 04137 000123
19 04140 103240
20 04141 040254
21
22
23 04142 006121
24 04143 006122
25 04144 000124
26 04145 000133
27 04146 024253
28 04147 122415
29 04150 000772
30 04151 166470
31
32 04153 111000
33 04154 122400
34 04155 103240
35 04156 040411
36 04157 034254
37 04160 055000
38 04161 177220
39 04162 030253
40 04163 051400
41
42 04164 006121
43 04165 105000
44 04166 166470 S401:
45
46 04170 032253
47 04171 162414
48
49 04172 100030 E\N111:
50
51 04173 006117 L\N065:
52
53

```

```

:0119 ESPCL
01 STAD3: ESTAS 3,SAD3
02
03 SETUP 100.
04 04244 006120 TAN067: JSR @ENTRIN ;INITIALIZE TEST
05 04245 000144 100. ;ITERATION VALUE
06 RAND
07 JSR @ENTRA ;C(AC0)=RANDOM #
08 04246 006121 STA 0,STSV0 ;SAVE RANDOM DATA
09 LOBFAD ;RANDOM DATA
10 RAND
11 04250 006121 JSR @ENTRA ;C(AC0)=RANDOM #
12 04251 006122 JSR @RNADR ;C(AC0)=RANDOM #
13 04252 000132 LOWBF ;FROM A RANDOM ADDRESS IN
14 04253 000123 LOBFU ;LOWER BUFFER
15 04254 002253 STA 0,STLCO ;C(STLCO)=(RANDOM ADDR 1)
16 LOBFAD ;RANDOM DATA
17 RAND
18 04255 006121 JSR @ENTRA ;C(AC0)=RANDOM #
19 04256 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
20 04257 000132 LOWBF ;LOWER BUFFER
21 04260 000123 LOBFU
22 04261 024253 LDA 1,STLCO ;MAKE SURE RNDM ADDRESSES
23 04262 106415 SUB# 0,1,SNR ;FROM LOWRUFF ARE DIFFERENT
24 04263 00772 JMP SAD3 ;C(STLCO)=@ (RANDOM ADDR 2)
25 04264 103240 ADDR 0,0 ;
26 04265 040254 STA 0,STLCO ;
27 HIGFAD ;RANDOM DATA
28 RAND
29 04266 006121 JSR @ENTRA ;C(AC0)=RANDOM #
30 04267 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
31 04270 000124 HIGBF ;THE UPPER BUFFER
32 04271 000133 HIGFU
33 04272 111000 MOV 0,2 ;AC0=AC2=(RNDM ADDR 3)
34 04273 034254 LDA 3,STLCO ;STORE (RANDOM ADDR 2) IN
35 04274 055000 STA 3,0,2 ;(RNDM ADDR 3) AND STORE
36 04275 177220 ADDR ;(RANDOM ADDR 1) IN
37 04276 030253 LDA 2,STLCO ;C(AC0)=RANDOM #
38 04277 051400 STA 2,0,3 ;STORE (RANDOM ADDR 3)/2+
39 04300 115222 MOVZ 0,3,@ZC ;ALSO STORE (RNDM ADDR 3)/2+
40 04301 114010 ADI 1,3 ;BIT 15 IN AC3 AND
41 04302 143370 HLV 0 ;ESTA INSTRUCTION
42 04303 103240 ADDR 0,0 ;ESTA INSTRUCTION
43 04304 040603 STA 0,+3 ;
44 04305 024255 LDA 1,STSV0 ;C(AC0)=RANDOM DATA
45 04306 147470 ESTA 1,@,5 ;ESTA MUST STORE
46 100000
47 04310 022253 LDA @,STLCO ;RANDOM DATA IN
48 04311 106414 SUB# 0,1,+SZR ;RANDOM ADDR 1
49 ERROR
50 04312 100030 ENR113: XOP 0,0,0 ;ERROR CALL
51 LOOP
52 04313 006117 LAN067: JSR @ENTLO ;ITERATE TEST ROUTINE

```

```

:0120 ESPCL
01
02
03 ; MISC TEST OF "ELDA" AND "ESTA"
04 ;
05
06
07 STAE0: SETUP 100.
08 04314 006120 TAN070: JSR @ENTIN ;INITIALIZE TEST
09 04315 000144 LDA 100. ;ITERATION VALUE
10 04316 020256 LDA 0,0,100 ;
11 04317 040253 STA 0,STLCO ;
12 04320 030132 LDA 2,LOWBF ;
13 04321 034124 LDA 3,HIGBF ;
14 04322 054255 STA 3,STSV0 ;SAVE AC3
15 RAND ;RANDOM DATA
16 04323 006121 JSR @ENTRA ;C(AC0)=RANDOM #
17 04324 034255 LDA 3,STSV0 ;RESTORE AC3
18 04325 145070 ESTA 0,0,2 ;FILL LOWER BUFFER AND HIGHER
19 000000
20 04327 143470 ESTA 0,0,3 ;BUFFER WITH RANDOM DATA
21 000000
22 04331 151400 INC 2,2 ;
23 04332 175400 INC 3,3 ;
24 04333 054255 STA 3,STSV0 ;SAVE AC3
25 04334 014253 DSZ STLCO ;
26 04335 000766 JMP SAE0 ;
27 04336 024256 LDA 1,0,100 ;
28 04337 044255 STA 1,STLCO ;
29 04340 034132 LDA 3,LOWBF ;
30 04341 030124 LDA 2,HIGBF ;
31 04342 123470 SAE0: ELDA 0,0,3 ;READ BACK BOTH BUFFERS
32 000000
33 04344 127070 ELDA 1,0,2 ;AND COMPARE
34 000000
35 04346 106414 SUR# 0,1,+SZR ;
36 ERROR
37 04347 100030 ENR114: XOP 0,0,0 ;ERROR CALL
38 04350 014255 DSZ STLCO ;
39 04351 000771 JMP SAE0 ;
40 LOOP
41 04352 006117 LAN070: JSR @ENTLO ;ITERATE TEST ROUTINE
42

```

10121 ESPCL

```

01
02
03
04 04353 006120 T\W071: SETUP 100.
05 04354 000144 JSR @ENTIN ;INITIALIZE TEST
06 04355 030132 LDA 2,LDMBF ;ITERATION VALUE
07 04356 034124 LDA 3,HIGBF ;MISC TEST OF ELDA,ESTA
08 04357 163070 ELEF 0,2,2 ; AND ELEF
09 000002
10 04361 167470 ELEF 1,2,3 ;
11 04363 177470 ELEF 3,2,3 ;
12 04365 000002 ELEF 2,2,2 ;
13 04365 173070 ELEF 2,0,2 ;
14 04367 153070 ESTA 3,0,3 ;
15 000000
16 04371 157470 ESTA 2,0,2 ;
17 000000
18 04373 133070 ELDA 2,0,2 ;
19 000000
20 04375 137470 ELDA 3,0,3 ;
21 000000
22 04377 112414 SUB# 0,2,SZR
23 000000 ERROR
24 04400 100030 E\W115: XOP 0,0,0 ;ERROR CALL
25 000000 SUB# 1,3,SZR
26 04401 136414 ERROR 0,0,0 ;
27 000000
28 04402 100030 E\W116: XOP 0,0,0 ;ERROR CALL
29 000000 LOOP
30 04403 006117 L\W071: JSR @ENTLO ;ITERATE TEST ROUTINE
31
32
33

```

10122 ESPCL

```

01
02
03
04
05 04404 006120 T\W072: SETUP 100.
06 04405 000144 JSR @ENTIN ;INITIALIZE TEST
07 04405 000144 RANU 100. ;ITERATION VALUE
08 04406 006121 JSR @ENTRA ;C(ACO)RANDOM #
09 04407 040436 STA 0,SAE2 ;
10 04410 100510 XDR 0,0 ;MISC TEST OF ELEF,ELDA AND
11 04411 122470 ELDA 0,SAE2,-1,1 ;ESTA
12 000033
13 04413 142470 ESTA 0,SAE2,-1,1 ;
14 000032
15 04415 111000 MOV 0,2 ;
16 04416 150000 MOV 2,3 ;
17 04417 152470 ESTA 2,SAE2+1,-1,1 ;
18 000027
19 04421 156470 ESTA 3,SAE2+2,-1,1 ;
20 000026
21 04423 176470 ELEF 3,SAE2,-1,1 ;
22 000021
23 04425 133470 ELDA 2,0,3 ;
24 000000
25 04427 127470 ELDA 1,1,3 ;
26 000001
27 04431 146414 SUB# 2,1,SZR
28 000001 ERROR
29 04432 100030 E\W117: XOP 0,0,0 ;ERROR CALL
30 04433 123470 ELDA 0,2,3 ;
31 000002
32 04435 172470 ELEF 2,SAE2,-1,1 ;
33 000007
34 04437 137070 ELDA 3,3,2 ;
35 000003
36 04441 116414 SUB# 0,3,SZR
37 000000 ERROR
38 04442 100030 E\W120: XOP 0,0,0 ;ERROR CALL
39 000000 LOOP
40 04443 006117 L\W072: JSR @ENTLO ;ITERATE TEST ROUTINE
41 04444 000405 JMP *-5
42 04445 000004 SAE2: .RLK 4
43

```

```

10123 ESPCL
01
02
03
04
05 STAE3: SETUP
06 04451 006120 T\N073: JSR @ENTIN ; INITIALIZE TEST
07 04452 000144 100. ; ITERATION VALUE
08 04453 006121 RAND
09 04454 111000 JSR @ENTRA ; C(CAC0)=RANDOM #
10 04455 176870 MOV 0,2 ; MISC TEST OF ELDA,ESTA
11 04456 000016 ELEF 3,SAE3--1,1 ; RAND ELEF
12 04457 155870 ESTA 2,0,3 ;
13 04458 000000 XCH 2,3 ;
14 04461 154710 ESTA 3,1,2 ;
15 04462 157070 ELDA 0,0,2 ;
16 04464 123070 ELDA 1,1,2 ;
17 04466 127070 ELDA 0,1,SZR ;
18 04470 106414 SUR# 0,1,SZR
19 04471 100030 E\N121: XOP 0,0,0 ; ERROR CALL
20 04472 006117 L\N073: LOOP @ENTLO ; ITERATE TEST ROUTINE
21 04473 000403 JMP *+3
22 04474 000002 SAE3: -BLK 2
23
24
25
26
27
28

10124 ESPCL
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

; CHECKING OUT "EJMP" -LONG JMP INSTRUCTION
;
;
JMPA0:
FILL XERR
JSR @JFILL
XERR
SETUP 100.
@ENTIN ; INITIALIZE TEST
; ITERATION VALUE
RAND
JSR @ENTRA
NEG 0,2 ; C(CAC0)=RANDOM #
NEG 2,1 ; EJMP SHOULD NOT
HLV 1 ; CHANGE ANY AC'S AND
MOV 1,3 ; STATE OF CARRY.
EJMP 1,1 ;
ADD# 0,2,SZR ;
ERROR
XOP 0,0,0 ; ERROR CALL
SUB# 1,3,SZR ;
ERROR
XOP 0,0,0 ; ERROR CALL
MOV 0,0,SNC ;
ERROR
XOP 0,0,0 ; ERROR CALL
LOOP
@ENTLO ; ITERATE TEST ROUTINE
JSR

```

10125 ESPCL

```

01 JMPB0: EJMP1 EJMP,HIBFAD,JPB0
02
03
04 FILL XERR
05 JSR @JFILL ;FILL UPPER AND LOWER
06 XERR ;SCRATCH BUFFER AREA
07 ;WITH (XERR)
08 SETUP 100.
09 @ENTIN ;INITIALIZE TEST
10 T\N075: JSR ;ITERATION VALUE
11 RAND
12 HIBFAD
13 JSR @ENTRA ;C(AC0)=RANDOM #
14 JSR @RNADR ;GET A RANDOM ADDRESS IN
15 HIBGF ;THE UPPER BUFFER
16 HIBFU
17 MOV 0.2 ;AC0=AC2=RANDOM ADDR
18 LDA 1,JMLC0 ;(JMP @JMLC1) IS STORED IN
19 STA 1,0.2 ;RANDOM ADDRESS.
20 ELEF 1,JPB0-.+2.1 ;ADDR OF (EJMP*) IS
21
22 STA 1,JMLC1 ;STORED IN JMLC1 FOR
23 STA 0.+.2 ;PROPER RETURN
24 EJMP 0.0 ;
25
26 ERROR
27 0.0,0 ;ERROR CALL
28 XDP LOOP
29 04542 006117 L\N075: JSR @ENTLO ;ITERATE TEST ROUTINE

```

10126 ESPCL

```

01 EJMP,LOBFAD,JPB1
02
03
04 XERR
05 @JFILL ;FILL UPPER AND LOWER
06 ;SCRATCH BUFFER AREA
07 ;WITH (XERR)
08 SETUP 100.
09 @ENTIN ;INITIALIZE TEST
10 T\N076: JSR ;ITERATION VALUE
11 RAND
12 LOBFAD
13 JSR @ENTRA ;C(AC0)=RANDOM #
14 JSR @RNADR ;GET A RANDOM ADDRESS IN
15 LOBF ;LOWER BUFFER
16 LOBFU
17 MOV 0.2 ;AC0=AC2=RANDOM ADDR
18 LDA 1,JMLC0 ;(JMP @JMLC1) IS STORED IN
19 STA 1,0.2 ;RANDOM ADDRESS.
20 ELEF 1,JPB1-.+2.1 ;ADDR OF (EJMP*) IS
21
22 STA 1,JMLC1 ;STORED IN JMLC1 FOR
23 STA 0.+.2 ;PROPER RETURN
24 EJMP 0.0 ;
25
26 ERROR
27 0.0,0 ;ERROR CALL
28 XDP LOOP
29 04565 006117 L\N076: JSR @ENTLO ;ITERATE TEST ROUTINE

```

```

10127 ESPL
01
02 JMPB2: EJMP2 EJMP,JPB2
03
04 FILL XERR
05 JSR @JFILL ;FILL UPPER AND LOWER
06 XERR ;SCRATCH BUFFER AREA
07 ;WITH (XERR)
08 SETUP
09 T\N077: JSR @NENTIN ;INITIALIZE TEST
10 @NENTIN ;ITERATION VALUE
11 HIRFAD
12 RAND
13 04572 006121 ;C(LACO)=RANDOM #
14 04573 006122 ;SET A RANDOM ADDRESS IN
15 04574 000124 ;THE UPPER BUFFER
16 04575 000133
17 04576 115000 MOV 0,3
18 04577 024257 LOA 1,JMLCO
19 04600 045400 STA 1,0,3
20 04601 164470 ELEF 1,JPB2--,2,1
21 000011
22 04603 044260 STA 1,JMLC1
23 04604 164470 ELEF 1,JPB2--,1
24 000004
25 04606 122400 SUB 1,0
26 04607 040402 STA 0,*,2
27 04610 102470 JPB2: EJMP 0,1
28 000000
29 ERROR
30 04612 100030 EAR127: XOP
31 LOOP
32 04613 006117 LAN077: JSR @NENTLO ;ITERATE TEST ROUTINE

10128 ESPL
01
02 JMPB3:
03
04 04614 006125 FILL XERR
05 04615 100030 JSR @JFILL
06 XERR ;FILL UPPER AND LOWER
07 ;SCRATCH BUFFER AREA
08 ;WITH (XERR)
09 100,
10 @NENTIN ;INITIALIZE TEST
11 @NENTIN ;ITERATION VALUE
12 0,0,0
13 04622 100030 EAR130: XOP ;ERROR CALL
14 04623 102470 EJMP ;JUMP TO PC OF EJMP*3
15 000002
16 ERROR
17 04625 100030 EAR131: XOP ;ERROR CALL
18 04626 102470 EJMP ;JUMP TO PC OF EJMP*3
19 000002
20 ERROR
21 04630 100030 EAR132: XOP ;ERROR CALL
22 04631 102470 EJMP ;JUMP TO PC OF EJMP*3
23 000002
24 ERROR
25 04633 100030 EAR133: XOP ;ERROR CALL
26 04634 102470 EJMP ;JUMP TO PC OF EJMP*3
27 000002
28 ERROR
29 04636 100030 EAR134: XOP ;ERROR CALL
30 04637 102470 EJMP ;JUMP TO PC OF EJMP*3
31 000002
32 ERROR
33 04641 100030 EAR135: XOP ;ERROR CALL
34 04642 102470 EJMP ;JUMP TO PC OF EJMP*3
35 000002
36 ERROR
37 04644 100030 EAR136: XOP ;ERROR CALL
38 04645 102470 EJMP ;JUMP TO PC OF EJMP*3
39 000002
40 ERROR
41 04647 100030 EAR137: XOP ;ERROR CALL
42 04650 102470 EJMP ;JUMP TO PC OF EJMP*3
43 000002
44 ERROR
45 04652 100030 EAR140: XOP ;ERROR CALL
46 04653 102470 EJMP ;JUMP TO PC OF EJMP*3
47 000002
48 ERROR
49 04655 100030 EAR141: XOP ;ERROR CALL
50 04656 102470 EJMP ;JUMP TO PC OF EJMP*3
51 000002
52 ERROR
53 04660 100030 EAR142: XOP ;ERROR CALL
54 04661 102470 EJMP ;JUMP TO PC OF EJMP*3
55 000002
56 ERROR
57 04663 100030 EAR143: XOP ;ERROR CALL
58 04664 102470 EJMP ;JUMP TO PC OF EJMP*3
59 000002
60 ERROR

```

```

0129 ESPCL
01 04666 10030 E\R144: XOP      0.0,0
02 04667 102470 EJMPC          2.1
03 000002
04 ERROR
05 04671 10030 E\R145: XOP      0.0,0
06 04672 102470 EJMPC          2.1
07 000002
08 ERROR
09 04674 10030 E\R146: XOP      0.0,0
10 04675 102470 EJMPC          2.1
11 000002
12 ERROR
13 04677 10030 E\R147: XOP      0.0,0
14 LOOP
15 04700 006117 L\N100: JSR     @ENTLO ;ITERATE TEST ROUTINE
16

```

```

10130 ESPCL
01 EJMP,LOHFAD,JP84,2
02
03
04 XERR ;FILL UPPER AND LOWER
05 XERR ;SCRATCH BUFFER AREA
06 XERR ;WITH (XERR)
07
08 SETUP 100.
09 @ENTLO ;INITIALIZE TEST
10 100. ;ITERATION VALUE
11 LOHFAD
12 RAND
13 JSR @ENTRA ;C(AC0)=RANDOM #
14 JSR @RNADR ;GET A RANDOM ADDRESS IN
15 LOBF ;LOWER BUFFER
16 LOBFU
17 MOV 0,P ;AC0=AC2+RNDM ADDRESS
18 LDA 1,JMLC0 ;(JMP @JMLC1) IS STORED
19 STA 1,0,2 ;IN RANDOM ADDRESS
20 ELEF 1,JP84-,*+2,1 ;ADDR OF (EJMP+3) IS STORED
21
22 STA 1,JMLC1 ;IN JMLC1 FOR GOOD RETURN
23 MOVZ 0,2,SZC ;(RNDM ADDR/2)+BIT IS IS
24 ADI 1,2 ;STORED IN AC2 AND
25 HLV 0 ;(RNDM ADDR/2) IS STORED
26 STA 0,*+2 ;IN EJMP INSTRUCTION
27 EJMP 0,2
28
29 ERROR
30 0,0,0 ;ERROR CALL
31 @ENTLO ;ITERATE TEST ROUTINE
32

```





```

0134 ESPCL
01 ERROR
02 05072 100630 EXR155: XOP 0,0,0 :ERROR CALL
03 LOOP
04 05073 006117 LAN104: JSR :ITERATE TEST ROUTINE

```

```

10133 ESPCL
01
02
03 : CHECKING MULTI-LEVEL INDIRECT "EJMP"
04
05
06 000000 JMP00: JMSMC=0
07 EJMP: LOBFAD, JP00, HIBFAD, 0, LOBFAD, JP000, JP001
08
09 FILL XERR
10 05017 006125 JSR @JFILL :FILL UPPER AND LOWER
11 05020 100030 XERR :SCRATCH BUFFER AREA
12 :WITH (XERR)
13
14 05021 006120 T\M104: JSR @ENTIN :INITIALIZE TEST
15 05022 000144 :ITERATION VALUE
16
17 RAND
18 05023 006121 JSR @ENTRA :C(AC0)=RANDOM #
19 05024 006122 JSR @RNADR :GET A RANDOM ADDRESS IN
20 05025 000132 LOWBF :LOWER BUFFER
21 05026 000123 LOBFU
22 05027 040261 STA 0, JMLC2 :C(JMLC2)=RANDOM ADDR 1
23
24 JP000: HIBFAD
25 RAND
26 05030 006121 JSR @ENTRA :C(AC0)=RANDOM #
27 05031 006122 JSR @RNADR :GET A RANDOM ADDRESS IN
28 05032 000124 HIBSF :THE UPPER BUFFER
29 05033 000133 HIBFU
30 05034 024261 LDA 1, JMLC2
31 05035 122415 SUB# 1,0, SNR
32 05036 000772 JMP JP000
33 05037 040262 STA 0, JMLC3 :C(JMLC3)=RANDOM ADDR 2
34 05040 103770 IORI 100000,0 :C(RNDM ADDR 1)=
35 :@RNDM ADDR 2)
36
37 JP001: LOBFAD
38 RAND
39 05043 006121 JSR @ENTRA :C(AC0)=RANDOM #
40 05044 006122 JSR @RNADR :GET A RANDOM ADDRESS IN
41 05045 000132 LOWBF :LOWER BUFFER
42 05046 000123 LOBFU
43 05047 024262 LDA 1, JMLC3
44 05050 106415 SUB# 0,1, SNR
45 05051 000772 JMP JP001
46 05052 024261 LDA 1, JMLC2
47 05053 122415 SUB# 1,0, SNR
48 05054 000767 JMP JP001
49 05055 042262 STA 0, @JMLC3 :C(RNDM ADDR 2)=RANDOM ADDR 3
50 05056 111000 MOV 0,2 :AC0=AC2=RNDM ADDR 3
51 05057 024257 LDA 1, JMLC0 :C(JMP @JMLC1) IS STORED IN
52 05060 045000 STA 1,0,2 :RNDM ADDR 3 AND ADDR
53 05061 166470 EIEF 1, JP00=-*2,1 :EJMP*3) IS STORED IN JMLC1
54
55 05063 044260 STA 1, JMLC1 :FOR PROPER RETURN
56 05064 020261 LDA 0, JMLC2 :AC0= RNDM ADDR 1
57 05065 103770 IORI 100000,0 :ADD INDIRECT BIT TO AC0
58
59 05067 040400 STA 0, *-2 :AND STORE IN EJMP
60 05070 102070 EJMP @0,0

```

```

10135 ESPCL
01 000001 JMPD1: JMSWC=1
02 000001 JMPD1: EJMP5
03
04
05 05074 006125 FILL XERR :FILL UPPER AND LOWER
06 05075 100030 XERR @JFILL :SCRATCH BUFFER AREA
07 :WITH (XERR)
08
09 05076 006120 T\N105: JSR @ENTIN :INITIALIZE TEST
10 05077 000144 100. :ITERATION VALUE
11 HIRFAD
12 RAND
13 05100 006121 JSR @ENTRA :C(CACO)=RANDOM #
14 05101 006122 JSR @RNADR :SET A RANDOM ADDRESS IN
15 05102 000124 HIRBF :THE UPPER BUFFER
16 05103 000133 HIRFU :
17 05104 040261 STA 0, JMLC2 :C(JMLC2)=RNDM ADDR 1
18 JPD10: HIRFAD
19 RAND
20 05105 006121 JSR @ENTRA :C(CACO)=RANDOM #
21 05106 006122 JSR @RNADR :SET A RANDOM ADDRESS IN
22 05107 000124 HIRBF :THE UPPER BUFFER
23 05110 000133 HIRFU :
24 05111 024261 LDA 1, JMLC2 :
25 05112 122415 SUB# 1,0,SNR :
26 05113 000772 JMP JPD10 :
27 05114 040262 STA 0, JMLC3 :C(JMLC3)=RNDM ADDR 2
28 05115 103770 IORI 100000,0 :C(RNDM ADDR 1)=
29
30 05117 042261 STA 0, @JMLC2 :C(RNDM ADDR 2)
31 RAND
32 JPD11:
33 05120 006121 JSR @ENTRA :C(CACO)=RANDOM #
34 05121 006122 JSR @RNADR :SET A RANDOM ADDRESS IN
35 05122 000124 HIRBF :THE UPPER BUFFER
36 05123 000133 HIRFU :
37 05124 024262 LDA 1, JMLC3 :
38 05125 106415 SUB# 0,1,SNR :
39 05126 000772 JMP JPD11 :
40 05127 024261 LDA 1, JMLC2 :
41 05130 122415 SUB# 1,0,SNR :
42 05131 000767 JMP JPD11 :
43 05132 042262 STA 0, @JMLC3 :C(RNDM ADDR 2)=RNDM ADDR 3
44 05133 111000 MOV 0,2 :ACO=AC2=RNDM ADDR 3
45 05134 024257 LDA 1, JMLC0 :C(JMP @JMLC1) IS STORED IN
46 05135 045000 STA 1,0,2 :RNDM ADDR 3 AND ADDR
47 05136 166470 ELEF 1, JPD1--,+2,1 :E(JMP+3) IS STORED IN JMLC1
48
49 05140 042260 STA 1, JMLC1 :FOR PROPER RETURN
50 05141 020261 LDA 0, JMLC2 :ACO= RNDM ADDR 1
51 05142 166470 ELEF 1, JPD1--,1 :C(RNDM ADDR 1 - PC OF EJMP+1)
52 05144 122400 SUB 1,0 :IS IN ACO
53 05145 103770 IORI 100000,0 :ADD INDIRECT BIT TO ACO
54
55 05147 040402 STA 0, +2 :AND STORE IN EJMP
56 05150 102470 JPD1: EJMP @0,1 :
57
58 ERROR
59 05152 100030 XCP 0,0,0 :ERROR CALL
60 LOOP

```

0137 ESPCL  
 01 000002 JMPD2: JMSWC=2  
 02 EJMP5 EJMP,LOBFAD,JP02,LOBFAD,2,HIBFAD,JP020,JP021  
 03  
 04  
 05 FILL XERR ;FILL UPPER AND LOWER  
 06 JSR @JFILL ;SCRATCH BUFFER AREA  
 07 XERR ;WITH (XERR)  
 08  
 09 SETUP 100.  
 10 JSR @ENTIN ;INITIALIZE TEST  
 11 @ENTIN ;ITERATION VALUE  
 12 LOBFAD  
 13 RAND  
 14 JSR @ENTRA ;C(AC0)=RANDOM #  
 15 JSR @RNADR ;GET A RANDOM ADDRESS IN  
 16 LOWBF ;LOWER BUFFER  
 17 LOBFU ;LOWER BUFFER  
 18 STA 0,JMLC2 ;C(JMLC2)=RNDM ADDR 1  
 19 JP020: LOBFAD  
 20 RAND  
 21 JSR @ENTRA ;C(AC0)=RANDOM #  
 22 JSR @RNADR ;GET A RANDOM ADDRESS IN  
 23 LOWBF ;LOWER BUFFER  
 24 LOBFU ;LOWER BUFFER  
 25 LOA 1,JMLC2 ;  
 26 SUB# 1,0,SNR ;  
 27 JMP JP020 ;  
 28 STA 0,JMLC3 ;C(JMLC3)=RNDM ADDR 2  
 29 TORI 100000\*0 ;C(RNDM ADDR 1)=  
 30 05177 042261 ;(RNDM ADDR 2)  
 31 JP021: HIBFAD  
 32 RAND  
 33 JSR @ENTRA ;C(AC0)=RANDOM #  
 34 JSR @RNADR ;GET A RANDOM ADDRESS IN  
 35 HIGHF ;THE UPPER BUFFER  
 36 HIBFU ;  
 37 LDA 1,JMLC3 ;  
 38 SUB# 0,1,SNR ;  
 39 JMP JP021 ;  
 40 LOA 1,JMLC2 ;  
 41 SUB# 1,0,SNR ;  
 42 JMP JP021 ;  
 43 STA 0,JMLC3 ;C(RNDM ADDR 2)=RNDM ADDR 3  
 44 MOV 0,2 ;AC0=AC2=RNDM ADDR 3  
 45 LDA 1,JMLC0 ;(JMP @JMLC1) IS STORED IN  
 46 STA 1,0,2 ;RNDM ADDR 3 AND ADDR  
 47 ELEF 1,JP02,+,2,1 ;(EJMP+3) IS STORED IN JMLC1  
 48  
 49 STA 1,JMLC1 ;FOR PROPER RETURN  
 50 LOA 0,JMLC2 ;AC0= RNDM ADDR 1  
 51 MOVZ 0,2,9C ;(RNDM ADDR 1)/2+ BIT 15  
 52 ADI 1,2 ;IS IN AC2 AND AC0=  
 53 HLV 0 ;(RNDM ADDR 1)/2=  
 54 TORI 100000\*0 ;ADD INDIRECT BIT TO AC0  
 55  
 56 STA 0,\*,2 ;AND STORE IN EJMP  
 57 EJMP @0,2 ;  
 58 100000  
 59 ERROR  
 60 05232 100030 ENR155: XUP 0,0,0 ;ERROR CALL

0138 ESPCL  
 01  
 02 05233 006117 LNN106: JSR  
 @ENTLO ;ITERATE TEST ROUTINE

```

0140 FSPCL
01
02 05313 006117 L\N107: JSR          LOOP
                                AENTLO ;ITERATE TEST ROUTINE

10159 ESPCL
01 000003 JMP03: JMSWC=3
02 03      EJMP5  EJMP,HIBFAD,JPD3,LOBFAD,3,LOBFAD,JPD30,JPD31
03
04      FILL   XERR          ;FILL UPPER AND LOWER
05 05234 006125 JSR   @JFILL          ;SCRATCH BUFFER AREA
06 05235 100030 XERR          ;WITH (XERR)
07
08      SETUP 100.
09 05236 006120 T\N107: JSR   @ENTIN          ;INITIALIZE TEST
10 05237 000144 @ENTIN          ;ITERATION VALUE
11      HIBFAD
12      RAND
13 05240 006121 JSR   @ENTRA          ;C(AC0)=RANDOM #
14 05241 006122 JSR   @RNADR          ;GET A RANDOM ADDRESS IN
15 05242 000124 HIGBF          ;THE UPPER BUFFER
16 05243 000133 HIBFU
17 05244 000261 STA   0,JMLC2          ;C(JMLC2)=RNDM ADDR 1
18      JPD30: LOHFAD
19      RAND
20 05245 006121 JSR   @ENTRA          ;C(AC0)=RANDOM #
21 05246 006122 JSR   @RNADR          ;GET A RANDOM ADDRESS IN
22 05247 000132 LOWBF          ;LOWER BUFFER
23 05250 000123 LORFU
24 05251 024261 LDA   1,JMLC2          ;
25 05252 124415 SUB#  1,0,SNR          ;
26 05253 000772 JMP   JPD30           ;
27 05254 000262 STA   0,JMLC3          ;C(JMLC3)=RNDM ADDR 2
28 05255 103770 ORI   100000,0      ;C(RNDM ADDR 1)=
29
30 05257 042261 STA   0,@JMLC2       ;@RNDM ADDR 2)
31      JPD31: LOHFAD
32      RAND
33 05260 006121 JSR   @ENTRA          ;C(AC0)=RANDOM #
34 05261 006122 JSR   @RNADR          ;GET A RANDOM ADDRESS IN
35 05262 000132 LOWBF          ;LOWER BUFFER
36 05263 000123 LORFU
37 05264 024262 LDA   1,JMLC3          ;
38 05265 106415 SUB#  0,1,SNR          ;
39 05266 000772 JMP   JPD31           ;
40 05267 024261 LDA   1,JMLC2          ;
41 05270 122415 SUB#  1,0,SNR          ;
42 05271 000767 JMP   JPD31           ;
43 05272 042262 STA   0,@JMLC3       ;C(RNDM ADDR 2)=RNDM ADDR 3
44 05273 111000 MOV   0,2            ;AC0=AC2=RNDM ADDR 3
45 05274 024257 LDA   1,JMLC0          ;C(JMP @JMLC1) IS STORED IN
46 05275 045000 STA   1,0,2          ;RNDM ADDR 3 AND ADDR
47 05276 164470 ELEF  1,JPD3+,2,1      ;EJMP+3) IS STORED IN JMLC1
48
49 05300 000014 STA   1,JMLC1        ;FOR PROPER RETURN
50 05301 020261 LDA   0,JMLC2        ;AC0= RNDM ADDR 1
51 05302 115222 MOVZR  0,3,2C          ;(RNDM ADDR 1)/2+ BIT 15
52 05303 114010 A01   1,3            ;IS IN AC3 AND AC0=
53 05304 143370 HLV   0,0            ;(RNDM ADDR 1)/2
54 05305 103770 ORI   100000,0        ;ADD INDIRECT BIT TO AC0
55      100000
56 05307 040402 STA   0,+2          ;AND STORE IN EJMP
57 05310 103470 EJMP  @0,3          ;
58
59      100000
60 0531P 100030 ENR156: YOP 0,0,0   ;ERROR CALL

```

```

10141 ESPCL
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
; CHECKING OUT "EJSR" - LONG JSR INSTRUCTION
;
JSRA0:
08 05314 006125 JSR @JFILL ;FILL UPPER AND LOWER
09 05315 100030 XERR ;SCRATCH BUFFER AREA
;WITH (XERR)
10 SETUP 100. ;INITIALIZE TEST
11 JSR @JENTIN ;ITERATION VALUE
12 05316 006120 T\N110: JSR ;FILL UPPER AND LOWER
13 05317 000184 RAND ;SCRATCH BUFFER AREA
14 ;WITH (XERR)
15 05320 006121 JSR @ENTRA ;INITIALIZE TEST
16 05321 110400 NEG 0,2 ;ITERATION VALUE
17 05322 144400 NEG 2,1 ;AFTER EXECUTING FJSR,
18 05323 176040 ADC 3,3 ;FACT MUST BE ED, TO
19 05324 106470 EJSR 1,1 ;ADDP OF (JRA1+2)
20 05325 000001 ADD# 0,2,SZR ;
21 05326 113014 ERROR ;
22 05327 100030 E\N157: XOP 0,0,0 ;ERROR CALL
23 05328 147014 ADD# 2,1,SZR ;
24 05329 100030 E\N160: XOP 0,0,0 ;ERROR CALL
25 05330 101003 EROR ;
26 05331 100030 E\N161: XOP 0,0,0 ;ERROR CALL
27 05332 174015 MOV 3,3,SNR ;
28 05333 100030 E\N162: XOP 0,0,0 ;ERROR CALL
29 05334 174015 COM# ;
30 05335 100030 E\N163: XOP 0,0,0 ;ERROR CALL
31 05336 006117 L\N110: JSR @ENTLO ;ITERATE TEST ROUTINE
32
33
34
35
10142 ESPCL
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
JSRA1:
03 05337 006125 JSR @JFILL ;FILL UPPER AND LOWER
04 05340 100030 XERR ;SCRATCH BUFFER AREA
;WITH (XERR)
06 SETUP 100. ;INITIALIZE TEST
07 05341 006120 T\N111: JSR ;ITERATION VALUE
08 05342 000184 ADC 3,3 ;AFTER EXECUTING FJSR,
09 05343 176040 ADC 0,JRA1-.+1,1 ;FACT MUST BE ED, TO
10 05344 162470 ELEF 1,1 ;ADDP OF (JRA1+2)
11 000003 EJSR 1,1 ;
12 05346 106470 JRA1: EJSR 0,3,SZR ;
13 05350 000001 SUB# 0,0,0 ;ERROR CALL
14 05350 116414 ERROR ;
15 05351 100030 E\N163: XOP LOOP
16 05352 006117 L\N111: JSR @ENTLO ;ITERATE TEST ROUTINE

```

10143 ESPCL

```

01
02
03
04
05 05353 006125
06 05354 100030
07
08
09 05355 006120 T\N112: JSR @ENTRA
10 05356 000144 LOBFAD
11
12
13 05357 006121 JSR @ENTRA
14 05360 006122 JSR @ENTRA
15 05361 000132 LOBRF
16 05362 000133 LOBFU
17 05363 111000 MOV 0,2
18 05364 024257 LDA 1,JMLCO
19 05365 045000 STA 1,0,2
20 05366 166470 ELEF 1,JR60--,+2,1
21
22 05370 044260 STA 1,JMLC1
23 05371 040402 STA 0,+2
24 05372 106070 JR60: EJSR 0,0
25
26
27 05374 100030 ERROR
28
29 05375 006117 E\N164: XOP
30
31 05375 006117 L\N112: JSR @ENTLO ;ITERATE TEST ROUTINE

```

10144 ESPCL

```

01
02
03
04
05 05376 006125
06 05377 100030
07
08
09 05400 006120 T\N113: JSR @ENTIN ;INITIALIZE TEST
10 05401 000144 ;ITERATION VALUE
11
12
13 05402 006121 JSR @ENTRA
14 05403 006122 JSR @ENTRA
15 05404 000124 HIGHF
16 05405 000133 HIRFU
17 05406 115000 MOV 0,3
18 05407 024257 LDA 1,JMLCO
19 05410 045400 STA 1,0,3
20 05411 166470 ELEF 1,JR61--,+2,1
21
22 05413 044260 STA 1,JMLC1
23 05414 166470 ELEF 1,JR61--,+1
24
25 05416 122400 SUB 1,0
26 05417 040402 STA 0,+2
27
28
29 05422 100030 E\N165: XOP
30
31 05423 006117 L\N113: JSR @ENTLO ;ITERATE TEST ROUTINE

```

| 10145 | ESPCL        | JSRB2:  | EJMP3  | EJSR,HIBFAD,JRB2.2                | JSRB2: | EJMP3 | EJSR,HIBFAD,JRB2.2 | 10146 | ESPCL        | JSRB3:  | EJMP3  | EJSR,LOBFAD,JRB3.3                |
|-------|--------------|---------|--------|-----------------------------------|--------|-------|--------------------|-------|--------------|---------|--------|-----------------------------------|
| 01    |              |         |        |                                   |        |       |                    | 01    |              |         |        |                                   |
| 02    |              |         |        |                                   |        |       |                    | 02    |              |         |        |                                   |
| 03    |              |         |        |                                   |        |       |                    | 03    |              |         |        |                                   |
| 04    | 05434 006125 |         | FILL   | :FILL UPPER AND LOWER             |        |       |                    | 04    | 05452 006125 |         | FILL   | :FILL UPPER AND LOWER             |
| 05    | 05425 100030 |         | JSR    | :SCRATCH BUFFER AREA              |        |       |                    | 05    | 05453 100030 |         | JSR    | :SCRATCH BUFFER AREA              |
| 06    |              |         | XERR   | :WITH (XERR)                      |        |       |                    | 06    |              |         | XERR   | :WITH (XERR)                      |
| 07    |              |         | SETUP  |                                   |        |       |                    | 07    |              |         | SETUP  |                                   |
| 08    | 05426 006120 | TAN114: | JSR    | :INITIALIZE TEST                  |        |       |                    | 08    | 05454 006120 | TAN115: | JSR    | :INITIALIZE TEST                  |
| 09    | 05427 000144 |         | 100.   | :ITERATION VALUE                  |        |       |                    | 09    | 05455 000144 |         | 100.   | :ITERATION VALUE                  |
| 10    |              |         | HIBFAD |                                   |        |       |                    | 10    |              |         | LOBFAD |                                   |
| 11    |              |         | RAND   |                                   |        |       |                    | 11    |              |         | RAND   |                                   |
| 12    | 05430 006121 |         | JSR    | :C(AC0)=RANDOM #                  |        |       |                    | 12    | 05456 006121 |         | JSR    | :C(AC0)=RANDOM #                  |
| 13    | 05431 006122 |         | JSR    | :GET A RANDOM ADDRESS IN          |        |       |                    | 13    | 05457 006122 |         | JSR    | :GET A RANDOM ADDRESS IN          |
| 14    | 05432 000124 |         | HIBF   | :THE UPPER BUFFER                 |        |       |                    | 14    | 05460 000132 |         | LOBF   | :LOWER BUFFER                     |
| 15    | 05433 000133 |         | HIBFU  |                                   |        |       |                    | 15    | 05461 000123 |         | LOBFU  |                                   |
| 16    | 05434 111000 |         | MOV    | :AC0=AC2-RNDM ADDRESS             |        |       |                    | 16    | 05462 111000 |         | MOV    | :AC0=AC2-RNDM ADDRESS             |
| 17    | 05435 024257 |         | LOA    | :JMP @JMLC1) IS STORED            |        |       |                    | 17    | 05463 024257 |         | LDA    | :JMP @JMLC1) IS STORED            |
| 18    | 05436 045000 |         | STA    | :IN RANDOM ADDRESS                |        |       |                    | 18    | 05464 045000 |         | STA    | :IN RANDOM ADDRESS                |
| 19    | 05437 166470 |         | ELEF   | :ADDR OF (EJSR*3) IS STORED       |        |       |                    | 19    | 05465 166470 |         | ELEF   | :ADDR OF (EJSR*3) IS STORED       |
| 20    |              |         |        |                                   |        |       |                    | 20    |              |         |        |                                   |
| 21    | 05441 044260 |         | STA    | :IN JMLC1 FOR GOOD RETURN         |        |       |                    | 21    | 05467 044260 |         | STA    | :IN JMLC1 FOR GOOD RETURN         |
| 22    | 05442 11222  |         | MOVZR  | :0.2,SZC :(RNDM ADDR/2)+8IT 15 IS |        |       |                    | 22    | 05470 11222  |         | MOVZH  | :0.3,SZC :(RNDM ADDR/2)+8IT 15 IS |
| 23    | 05443 110010 |         | ADJ    | :STORED IN AC2 AND                |        |       |                    | 23    | 05471 114010 |         | ADJ    | :STORED IN AC3 AND                |
| 24    | 05444 143370 |         | HLV    | ::(RNDM ADDR/2) IS STORED         |        |       |                    | 24    | 05472 143370 |         | HLV    | ::(RNDM ADDR/2) IS STORED         |
| 25    | 05445 040402 |         | STA    | :IN EJSR INSTRUCTION              |        |       |                    | 25    | 05473 040402 |         | STA    | :IN EJSR INSTRUCTION              |
| 26    | 05446 107070 | JRB2:   | EJSR   |                                   |        |       |                    | 26    | 05474 107470 | JRB3:   | FJSR   |                                   |
| 27    |              |         |        |                                   |        |       |                    | 27    |              |         |        |                                   |
| 28    |              |         | ERRDM  |                                   |        |       |                    | 28    |              |         | ERRDM  |                                   |
| 29    | 05450 100030 | EVR16:  | XOP    | :ERROR CALL                       |        |       |                    | 29    | 05476 100030 | EVR167: | XOP    | :ERROR CALL                       |
| 30    |              |         | LOOP   |                                   |        |       |                    | 30    |              |         | LOOP   |                                   |
| 31    | 05451 006117 | L\N114: | JSR    | :ITERATE TEST ROUTINE             |        |       |                    | 31    | 05477 006117 | L\N115: | JSR    | :ITERATE TEST ROUTINE             |

```

:0147 ERPCL
01
02
03 ; CHECKING SINGLE LEVEL INDIRECT "EJSR"
04 ;
05 ;
06 ;
07 000001 JSRCL: JMSWC=1 EJMPS EJSR,HIBFAD,JRC1,HIBFAD,1,JRC10
08
09 FILL XERR
10 05504 006125 JSR @JFILL ;FILL UPPER AND LOWER
11 05501 100030 XERR ;SCRATCH BUFFER AREA
12 ;WITH (XERR)
13
14 05502 006120 TANI16: JSR @ENTIN ;INITIALIZE TEST
15 05503 000144 ;ITERATION VALUE
16
17 RAND LOBFAD
18 05504 006121 JSR @ENTRA ;C(ACO)=RANDOM #
19 05505 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
20 05506 000124 HIGBF ;THE UPPER BUFFER
21 05507 000133 HIHFU ;
22 05510 040261 STA 0,JMLC2 ;C(JMLC2)=RNDM ADDR 1
23 JRC10: HIBFAD
24 RAND
25 05511 006121 JSR @ENTRA ;C(ACO)=RANDOM #
26 05512 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
27 05513 000124 HIGBF ;THE UPPER BUFFER
28 05514 000133 HIHFU ;
29 05515 024261 LDA 1,JMLC2 ;
30 05516 122415 SUB# 1,0,SNR ;
31 05517 000772 JMP JRC10 ;
32 05520 042261 STA 0,@JMLC2 ;C(RNDM ADDR.1)=(RNDM ADDR 2)
33 05521 110000 MOV 0,2 ;ACO=AC3=RNDM ADDR 2
34 05522 024257 LDA 1,JMLC0 ;STORE (JMP @JMLC1) IN
35 05523 045000 STA 1,0,2 ;RNDM ADDR 2
36 05524 166470 ELEF 1,JRC1-,*2,1 ;STORE ADDR OF (EJSR+3) IN
37
38 05526 044260 STA 1,JMLC1 ;JMLC1 FOR GOOD RETURN
39 05527 020261 LDA 0,JMLC2 ;ACO=RNDM ADDR 1
40 05530 166470 ELEF 1,JRC1-,*1 ;(RNDM ADDR 1)=PC OF EJSR+1)
41 05532 122400 SUB 1,0 ;IS IN ACO
42 05533 103770 IORI 100000,0 ;ADD THE INDIRECT BIT TO
43 05535 100000 STA 0,*+2 ;ACO AND STORE IN EJSR
44 05536 106470 EJSR @0,*1 ;
45
46 ERROR
47
48 05540 100030 EAP170: XOP ;ERROR CALL
49 LOOP
50 05541 006117 LAN116: JSR @ENTLO ;ITERATE TEST ROUTINE

```

```

:0148 ESPCL
01
02
03 ; CHECKING MULTI-LEVEL INDIRECT "EJSR"
04 ;
05 ;
06 ;
07 000000 JSR00: JMSWC=0 EJMPS EJSR,LOBFAD,JR00,LOBFAD,0,HIBFAD,JR00,JR001
08
09 FILL XERR
10 05542 006125 JSR @JFILL ;FILL UPPER AND LOWER
11 05543 100030 XERR ;SCRATCH BUFFER AREA
12 ;WITH (XERR)
13
14 05544 006120 TANI17: JSR @ENTIN ;INITIALIZE TEST
15 05545 000144 ;ITERATION VALUE
16
17 RAND LOBFAD
18 05546 006121 JSR @ENTRA ;C(ACO)=RANDOM #
19 05547 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
20 05548 000132 LOWBF ;LOWER BUFFER
21 05551 000123 LOBFU ;
22 05552 040261 STA 0,JMLC2 ;C(JMLC2)=RNDM ADDR 1
23 JR000: LOBFAD
24 RAND
25 05553 006121 JSR @ENTRA ;C(ACO)=RANDOM #
26 05554 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
27 05555 000132 LOWBF ;LOWER BUFFER
28 05556 000123 LOBFU ;
29 05557 024261 LDA 1,JMLC2 ;
30 05560 122415 SUB# 1,0,SNR ;
31 05561 000772 JMP JR000 ;
32 05562 040262 STA 0,JMLC3 ;C(JMLC3)=RNDM ADDR 2
33 05563 103770 IORI 100000,0 ;C(RNDM ADDR 1)=
34 100000
35 05565 042261 STA 0,@JMLC2 ;(RNDM ADDR 2)
36
37
38 05566 006121 JSR @ENTRA ;C(ACO)=RANDOM #
39 05567 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
40 05570 000124 HIGBF ;THE UPPER BUFFER
41 05571 000133 HIHFU ;
42 05572 024262 LDA 1,JMLC3 ;
43 05573 106415 SUB# 0,1,SNR ;
44 05574 000772 JMP JR001 ;
45 05575 024261 LDA 1,JMLC2 ;
46 05576 122415 SUB# 1,0,SNR ;
47 05577 000767 JMP JR001 ;
48 05600 042262 STA 0,@JMLC3 ;C(RNDM ADDR 2)=RNDM ADDR 3
49 05601 111000 MOV 0,2 ;ACO=AC3=RNDM ADDR 3
50 05602 024257 LDA 1,JMLC0 ;(JMP @JMLC1) IS STORED IN
51 05603 045000 STA 1,0,2 ;RNDM ADDR 3 AND ADDR
52 05604 166470 ELEF 1,JRND0-,*2,1 ;EJSR+3) IS STORED IN JMLC1
53
54 05606 044260 STA 1,JMLC1 ;FOR PROPER RETURN
55 05607 020261 LDA 0,JMLC2 ;ACO= RNDM ADDR 1
56 05610 103770 IORI 100000,0 ;ADD INDIRECT BIT TO ACO
57
58 05612 040402 STA 0,*+2 ;AND STORE IN EJSR
59 05613 106470 EJSR @0,*0 ;
60

```



```

0149 ESPCL
01 000001 JSND1: JMSWC=1
02 05615 100030 E\R171: XOP 0.0.0 ERROR CALL
03 05616 006117 L\W117: JSR @ENTLO ;ITERATE TEST ROUTINE
04 05617 006125 JSR @ENTLO ;ITERATE TEST ROUTINE
05 05620 100030 XERR
06 05621 006120 T\N120: JSR @ENTIN ;INITIALIZE TEST
07 05622 000144 RAND
08 05623 006121 JSR @ENTRA ;C(AC0)=RANDOM #
09 05624 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
10 05625 000124 HIGBF ;THE UPPER BUFFER
11 05626 000133 HIRFU ;C(JMLC2)=RNDM ADDR 1
12 05627 040261 STA 0,JMLC2
13 05630 006121 JSR @ENTRA ;C(AC0)=RANDOM #
14 05631 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
15 05632 000124 HIGBF ;THE UPPER BUFFER
16 05633 000133 HIRFU ;C(JMLC3)=RNDM ADDR 2
17 05634 04261 LDA 1,JMLC2
18 05635 122415 SUB# 1.0,SNR
19 05636 000772 JMP JRD10 ;C(RNDM ADDR 1)=
20 05640 103770 TORI 100000.0 ;(RNDM ADDR 2)
21 05642 042261 STA 0,JMLC2
22 05643 006121 JSR @ENTRA ;C(AC0)=RANDOM #
23 05644 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
24 05645 000124 HIGBF ;THE UPPER BUFFER
25 05646 000133 HIRFU ;C(RNDM ADDR 2)=RNDM ADDR 3
26 05647 024262 LDA 1,JMLC3
27 05648 106415 SUB# 1.0,SNR
28 05651 000772 JMP JRD11 ;C(AC0)=RNDM ADDR 3
29 05652 024261 LDA 1,JMLC0 ;(JMP @JMLC1) IS STORED IN
30 05653 122415 SUB# 1.0,SNR ;RNDM ADDR 3 AND ADDR
31 05654 000767 JMP JRD11 ;(EJSR*3) IS STORED IN JMLC1
32 05655 042262 STA 0,JJMLC3 ;FOR PROPER RETURN
33 05656 111800 MOV 0.2 ;RNDM ADDR 1 - PC OF EJSR+1
34 05657 024257 LDA 1,JMLC0 ;IS IN AC0
35 05660 045000 STA 1.0.2 ;ADD INDIRECT BIT TO AC0
36 05661 166470 ELEF 1,JRD11-+2.1 ;AND STORE IN EJSR
37 05663 044260 STA 1,JMLC1
38 05664 020261 LDA 0,JMLC2 ;RNDM ADDR 1
39 05665 166470 ELEF 1,JRD11-+.1 ;IS IN AC0
40 05667 122400 SUB# 1.0 ;ADD INDIRECT BIT TO AC0
41 05670 103770 TORI 100000.0 ;AND STORE IN EJSR
42 05672 040402 STA 0.+.2
43 05673 106470 EJSR @0.1
44 05675 100030 E\R172: XOP 0.0.0 ERROR CALL
45 05676 100000 LOOP
46 05677 100000 ERROR
47 05678 100000 ERROR
48 05679 100000 ERROR
49 05680 100000 ERROR
50 05681 100000 ERROR
51 05682 100000 ERROR
52 05683 100000 ERROR
53 05684 100000 ERROR
54 05685 100000 ERROR
55 05686 100000 ERROR
56 05687 100000 ERROR
57 05688 100000 ERROR
58 05689 100000 ERROR
59 05690 100000 ERROR
60 05691 100000 ERROR

```

0151 ESPCL  
01 05676 006117 LAN120: JSR

05NTLO :ITERATE TEST ROUTINE

```
01 000002 JSR02: JMSAC=2
02 EJMPS EJSR,LOHFAD,JR02,LOHFAD.2,HIFAD,JR20,JR021
03
04 FILL XERR :FILL UPPER AND LOWER
05 JSR @JFILL :SCRATCH BUFFER AREA
06 XERR :WITH (XERR)
07
08 SETUP 100, :INITIALIZE TEST
09 @ERTIN :ITERATION VALUE
10 LOHFAD
11 RAND
12 JSR @ENTRA :C(ACO)=RANDOM #
13 @RNADR :GET A RANDOM ADDRESS IN
14 LOWBF :LOWER BUFFER
15 LOBFU
16 STA 0,JMLC2 :C(JMLC2)=RNDM ADDR 1
17 LOBFAD
18 RAND
19 JSR @ENTRA :C(ACO)=RANDOM #
20 @RNADR :GET A RANDOM ADDRESS IN
21 LOWBF :LOWER BUFFER
22 LOBFU
23 LDA 1,JMLC2 ?
24 1,0,SNR ?
25 JSR020 :C(JMLC3)=RNDM ADDR 2
26 STA 0,JMLC3 :C(RNDM ADDR 1)=
27 100000,0
28 STA 0,@JMLC2 :@RNDM ADDR 2)
29 HIFAD
30 RAND
31 JSR @ENTRA :C(ACO)=RANDOM #
32 @RNADR :GET A RANDOM ADDRESS IN
33 HIGHF :THE UPPER BUFFER
34 LOBFU
35 LDA 1,JMLC3 ?
36 0,1,SNR ?
37 JR021 :C(RNDM ADDR 2)=RNDM ADDR 3
38 0,2
39 @JMLC3 :ACO=AC2=RNDM ADDR 3
40 1,JMLC2 :@JMLC1) IS STORED IN
41 1,0,SNR :RNDM ADDR 3 AND ADDR
42 @JMLC3 :((EJSR+3) IS STORED IN JMLC1
43 1,0,2 :FOR PROPER RETURN
44 1,JR02-+2,1 :ACO= RNDM ADDR 1
45 0,JMLC1 :ADDR 1)/2+ BIT 15
46 0,2,SZC :@RNDM ADDR 1)
47 0 :IS IN AC2 AND ACO=
48 100000,0 :RNDM ADDR 1)/2
49 0,+2 :ADD INDIRECT BIT TO ACO
50 @J,2 :AND STORE IN EJSR
51 0,2
52 ERROR
53 0,0,0
54 100000
55 JR02:
56 EJSR
57 100000
58 ERROR
59 100000
60 05755 100030 EVR173: XOP
```

0153 ESPCL

01 LOOP  
02 05756 006117 LAN121: JSR

0ENTLO ;ITERATE TEST ROUTINE

10154 ESPCL

01 U00003 JSR03:  
02 EIMPS  
03

EJSR,HIBFAD,JR03,LOBFAD,3,LOBFAD,JR030,JR031

```

04      FILL      XERR      ;FILL UPPER AND LOWER
05      JSR      @JFILL     ;SCRATCH BUFFER AREA
06      AERR      ;WITH (XERR)
07
08      SETUP     100.      ;INITIALIZE TEST
09      JSR      @ENTIN     ;ITERATION VALUE
10      HIBFAD   100.
11      RAND
12
13      JSR @ENTRA
14      JSR @RNADR
15      HIBFAD   @RNADR
16      HIBFU
17      STA     0,JMLC2
18      JRD30:  LOBFAD
19      RAND
20      JSR @ENTRA
21      JSR @RNADR
22      LOMHF
23      LOMFU
24      LDA     1,JMLC2
25      SUB#    1,0,SNR
26      JMP     JR030
27      STA     0,JMLC3
28      TGR1   100000,0
29
30      STA     0,@JMLC2
31      JRD31:  LOBFAD
32      RAND
33      JSR @ENTRA
34      JSR @RNADR
35      LOMHF
36      LOMFU
37      LDA     1,JMLC3
38      SUB#    0,1,SNR
39      JMP     JR031
40      LDA     1,JMLC2
41      SUB#    1,0,SNR
42      JMP     JR031
43      STA     0,@JMLC3
44      MOV     0,2
45      LDA     1,JMLC0
46      STA     1,0,2
47      ELEF   1,JRD30,*+2,1
48
49      STA     1,JMLC1
50      LDA     0,JMLC2
51      MOVZR   0,3,STC ;(RNDM ADDR 1)/2+ RIT 15
52      ADI     1,3
53      MVI     0
54      TURI   100000,0
55
56      STA     0,*+2
57      EJSR   @0,3
58
59      ERROR
60      XOP    0,0,0
        ;ERROR CALL

```

0154 ESPCL

01 U00003 JSR03:  
02 EIMPS  
03

EJSR,HIBFAD,JR03,LOBFAD,3,LOBFAD,JR030,JR031

```

04      FILL      XERR      ;FILL UPPER AND LOWER
05      JSR      @JFILL     ;SCRATCH BUFFER AREA
06      AERR      ;WITH (XERR)
07
08      SETUP     100.      ;INITIALIZE TEST
09      JSR      @ENTIN     ;ITERATION VALUE
10      HIBFAD   100.
11      RAND
12
13      JSR @ENTRA
14      JSR @RNADR
15      HIBFAD   @RNADR
16      HIBFU
17      STA     0,JMLC2
18      JRD30:  LOBFAD
19      RAND
20      JSR @ENTRA
21      JSR @RNADR
22      LOMHF
23      LOMFU
24      LDA     1,JMLC2
25      SUB#    1,0,SNR
26      JMP     JR030
27      STA     0,JMLC3
28      TGR1   100000,0
29
30      STA     0,@JMLC2
31      JRD31:  LOBFAD
32      RAND
33      JSR @ENTRA
34      JSR @RNADR
35      LOMHF
36      LOMFU
37      LDA     1,JMLC3
38      SUB#    0,1,SNR
39      JMP     JR031
40      LDA     1,JMLC2
41      SUB#    1,0,SNR
42      JMP     JR031
43      STA     0,@JMLC3
44      MOV     0,2
45      LDA     1,JMLC0
46      STA     1,0,2
47      ELEF   1,JRD30,*+2,1
48
49      STA     1,JMLC1
50      LDA     0,JMLC2
51      MOVZR   0,3,STC ;(RNDM ADDR 1)/2+ RIT 15
52      ADI     1,3
53      MVI     0
54      TURI   100000,0
55
56      STA     0,*+2
57      EJSR   @0,3
58
59      ERROR
60      XOP    0,0,0
        ;ERROR CALL

```

0155 ESPCL

01 LDDP  
02 06036 006117 LAM122: JSR  
03

0ENTLO ;ITERATE TEST ROUTINE

:0156 ESPCL

```
01  
02  
03 ; CHECKING OUT "EISZ" - LONG "ISZ" INSTRUCTION  
04 ;  
05  
06 ISZA0: EISZ1 EISZ,0,SNC,LOBFAD  
07  
08 SETUP 100.  
09 06037 006120 TWM123: JSR @ENTIN ;INITIALIZE TEST  
10 06040 000184 ;ITERATION VALUE  
11 LOHFAD  
12 RANO  
13 06041 006121 JSR @ENTRA ;C(CAO)=RANDOM #  
14 06042 004122 JSR @RNADR ;GET A RANDOM ADDRESS IN  
15 06043 006132 LOBF ;LOWER BUFFER  
16 06044 000132 LOBFU  
17 06045 040264 STA 0,ISLCO ;EISZ INSTRUCTION SHOULD  
18 06046 040410 STA 0,.*8. ;NOT CHANGE ANY AC'S  
19 06047 102400 SUB 0,0 ;STATE OF CARRY  
20 06050 042264 STA 0,@ISLCO ;  
21 RANO  
22 06051 006121 JSR @ENTRA ;C(CAO)=RANDOM #  
23 06052 115000 MOV 0,3 ;AC0=AC3=RANDOM DATA  
24 06053 104400 NEG 0,1 ;AC1=AC2=-(RNDM DATA)  
25 06054 131040 MOV0 1,2 ;CARRY IS INITIALIZED  
26 06055 112070 EISZ 0,0 ;RY MOV0  
27 06057 000000 SUB# 0,3,SZR ;  
28 ERROR ;  
29 06060 100030 EXR175: XOP 0,0,0 ;ERROR CALL  
30 06061 132414 SUB# 1,2,SZR ;  
31 ERROR ;  
32 06062 100030 EXR176: XOP 0,0,0 ;ERROR CALL  
33 06063 101003 MOV 0,0,SNC ;  
34 ERROR ;  
35 06064 100030 EXR177: XOP 0,0,0 ;ERROR CALL  
36 LDDP ;  
37 06065 006117 LAM123: JSR @ENTLO ;ITERATE TEST ROUTINE
```

```

10157 ESPCL      000000 ISZ90:  ISZSW=0      EISZ,LOBFAD,IZR0,0,ADI
01 01 000000 ISZ90:  ISZSW=0      EISZ2      EISZ,LOBFAD,IZR0,0,ADI
02 02 000000 ISZ90:  ISZSW=0      EISZ2      EISZ,LOBFAD,IZR0,0,ADI
03 03 000000 ISZ90:  ISZSW=0      EISZ2      EISZ,LOBFAD,IZR0,0,ADI
04 04 06066 006120 T\N124:  SETUP 100.      ;INITIALIZE TEST
05 05 06066 006120 T\N124:  JSR @ENTRIN ;ITERATION VALUE
06 06 06067 000144      RAND      ;ITERATION VALUE
07 07 06070 006121      JSR @ENTRA  ;C(AC0)=RANDOM #
08 08 06070 006121      STA @ENTRA  ;C(ISLCO)=RANDOM DATA
09 09 06071 040264      LOBFAD
10 10 06072 006121      RAND
11 11 06073 006122      JSR @ENTRA  ;C(AC0)=RANDOM #
12 12 06073 006122      JSR @RNADR   ;GET A RANDOM ADDRESS IN
13 13 06074 000132      LOBF      ;LOWER BUFFER
14 14 06075 000123      LOBFU
15 15 06075 000123      STA 0,ISLC1 ;C(ISLC1)=RANDOM ADDRESS
16 16 06076 040265      LDA 1,ISLCO ;STORE RANDOM DATA IN
17 17 06077 024264      STA 1,@ISLC1 ;RANDOM ADDRESS
18 18 06100 046265      STA 0,*+2
19 19 06101 040402      STA 0,0
20 20 06102 112070 IZ80:  EISZ      ;EXECUTE EISZ INSTRUCTION
21 21 000000
22 22 06104 101000      MOV 0,0      ;C(RNDM ADDR) MUST BE
23 23 06105 020264      LDA 1,0      ;=(RNDM DATA+1) FOR EISZ
24 24 06106 100010      ADI 1,0      ;AND =(RNDM DATA-1) FOR
25 25 06107 026265      LDA 1,@ISLC1 ;EDSZ INSTRUCTION
26 26 06110 106414      SUR# 0,1,SZR
27 27 06111 100030      ERROR#
28 28 06111 100030      XOP 0,0,0      ;ERROR CALL
29 29 06112 006117 L\N124:  LOOP
30 30 06112 006117 L\N124:  JSR @ENTLO ;ITERATE TEST ROUTINE
31 31 000000
32 32 06142 006117 L\N125:  LOOP
33 33 06142 006117 L\N125:  JSR @ENTLO ;ITERATE TEST ROUTINE

```

```

:0159 ESPCL      ISZSWF2  ISZ82:  ISZSWF2  ISZ83:  ISZSWF3  ISZ83:
01 000002  ISZ82:  EISZ2  EISZ2  EISZ2  EISZ2  EISZ2  EISZ2
02 000002  ISZ82:  EISZ,LOBFAD,IZB2,2,ADI  EISZ,LOBFAD,IZB3,3,ADI
03 000002  ISZ82:  EISZ,LOBFAD,IZB2,2,ADI  EISZ,LOBFAD,IZB3,3,ADI
04 000002  ISZ82:  EISZ,LOBFAD,IZB2,2,ADI  EISZ,LOBFAD,IZB3,3,ADI
05 06143 006120 T\N126:  T\N126:  T\N127:  T\N127:  T\N127:  T\N127:
06 06144 000144 100. 100. 100. 100. 100. 100.
07 06145 006121 ;C(ACO)=RANDOM # ;C(ACO)=RANDOM # ;C(ACO)=RANDOM #
08 06146 040264 STA 0,ISLCO ;C(ISLCO)=RANDOM DATA ;C(ISLCO)=RANDOM DATA
09 06147 006121 LORFAD ;C(ACO)=RANDOM # ;C(ACO)=RANDOM #
10 06148 000132 JSR ;RNADR ;GET A RANDOM ADDRESS IN ;GET A RANDOM ADDRESS IN
11 06149 000133 LORHFU ;LOWER BUFFER ;THE UPPER BUFFER
12 06150 006122 STA 0,ISLCO ;C(ISLCO)=RANDOM ADDRESS ;C(ISLCO)=RANDOM ADDRESS
13 06151 000132 STA 1,ISLCO ;STORE RANDOM DATA IN ;STORE RANDOM DATA IN
14 06152 000133 STA 1,ISLCO ;STORE RANDOM DATA IN ;STORE RANDOM DATA IN
15 06153 040265 LDA 1,ISLCO ;RANDOM ADDRESS ;RANDOM ADDRESS
16 06154 020264 STA 1,ISLCO ;RANDOM ADDRESS ;RANDOM ADDRESS
17 06155 040265 STA 1,ISLCO ;RANDOM ADDRESS ;RANDOM ADDRESS
18 06156 112222 MOVZR 0,2,5ZC ;(RNDM ADDR/2)+BIT 15 ;(RNDM ADDR/2)+BIT 15
19 06157 110010 ADI 1,2 ;IS STORED IN AC2 AND ;IS STORED IN AC3 AND
20 06158 143370 HLV 0 ; ; ;
21 06159 040402 STA 0,*2 ; ; ;
22 06160 113070 IZ82:  EISZ 0,2 ;EXECUTE EISZ INSTRUCTION ;EXECUTE EISZ INSTRUCTION
23 06161 000000 MOV 0,0 ;(RNDM ADDR) MUST BE ;(RNDM ADDR) MUST BE
24 06162 020264 LDA 0,ISLCO ;=(RNDM DATA+1) FOR EISZ ;=(RNDM DATA+1) FOR EISZ
25 06163 100010 ADI 1,0 ;AND =(RNDM DATA-1) FOR ;AND =(RNDM DATA-1) FOR
26 06164 020265 LDA 1,ISLCO ;EDSZ INSTRUCTION ;EDSZ INSTRUCTION
27 06165 106414 SUB# 0,1,5ZR ; ; ;
28 06166 106414 ERROR ; ; ;
29 06167 100030 XOP 0,0,0 ;ERROR CALL ;ERROR CALL
30 06168 100030 XOP 0,0,0 ;ERROR CALL ;ERROR CALL
31 06169 100030 XOP 0,0,0 ;ERROR CALL ;ERROR CALL
32 06170 006117 L\N126:  JSR ;ITERATE TEST ROUTINE ;ITERATE TEST ROUTINE
33 06171 006117 L\N126:  JSR ;ITERATE TEST ROUTINE ;ITERATE TEST ROUTINE

```

```

:0160 ESPCL      ISZSWF2  ISZ82:  ISZSWF2  ISZ83:  ISZSWF3  ISZ83:
01 000003  ISZ83:  EISZ2  EISZ2  EISZ2  EISZ2  EISZ2  EISZ2
02 000003  ISZ83:  EISZ,LOBFAD,IZB2,2,ADI  EISZ,LOBFAD,IZB3,3,ADI
03 000003  ISZ83:  EISZ,LOBFAD,IZB2,2,ADI  EISZ,LOBFAD,IZB3,3,ADI
04 000003  ISZ83:  EISZ,LOBFAD,IZB2,2,ADI  EISZ,LOBFAD,IZB3,3,ADI
05 06173 006120 T\N127:  T\N127:  T\N127:  T\N127:  T\N127:  T\N127:
06 06174 000144 100. 100. 100. 100. 100. 100.
07 06175 006121 ;C(ACO)=RANDOM # ;C(ACO)=RANDOM # ;C(ACO)=RANDOM #
08 06176 040264 STA 0,ISLCO ;C(ISLCO)=RANDOM DATA ;C(ISLCO)=RANDOM DATA
09 06177 006121 LORFAD ;C(ACO)=RANDOM # ;C(ACO)=RANDOM #
10 06178 000132 JSR ;RNADR ;GET A RANDOM ADDRESS IN ;GET A RANDOM ADDRESS IN
11 06179 000133 LORHFU ;LOWER BUFFER ;THE UPPER BUFFER
12 06180 006122 STA 0,ISLCO ;C(ISLCO)=RANDOM ADDRESS ;C(ISLCO)=RANDOM ADDRESS
13 06181 000132 STA 1,ISLCO ;STORE RANDOM DATA IN ;STORE RANDOM DATA IN
14 06182 000133 STA 1,ISLCO ;STORE RANDOM DATA IN ;STORE RANDOM DATA IN
15 06183 040265 LDA 1,ISLCO ;RANDOM ADDRESS ;RANDOM ADDRESS
16 06184 020264 STA 1,ISLCO ;RANDOM ADDRESS ;RANDOM ADDRESS
17 06185 040265 STA 1,ISLCO ;RANDOM ADDRESS ;RANDOM ADDRESS
18 06186 112222 MOVZR 0,3,5ZC ;(RNDM ADDR/2)+BIT 15 ;(RNDM ADDR/2)+BIT 15
19 06187 110010 ADI 1,3 ;IS STORED IN AC3 AND ;IS STORED IN AC3 AND
20 06188 143370 HLV 0 ; ; ;
21 06189 040402 STA 0,*2 ; ; ;
22 06190 113470 IZ83:  EISZ 0,3 ;EXECUTE EISZ INSTRUCTION ;EXECUTE EISZ INSTRUCTION
23 06191 000000 MOV 0,0 ;(RNDM ADDR) MUST BE ;(RNDM ADDR) MUST BE
24 06192 020264 LDA 0,ISLCO ;=(RNDM DATA+1) FOR EISZ ;=(RNDM DATA+1) FOR EISZ
25 06193 100010 ADI 1,0 ;AND =(RNDM DATA-1) FOR ;AND =(RNDM DATA-1) FOR
26 06194 020265 LDA 1,ISLCO ;EDSZ INSTRUCTION ;EDSZ INSTRUCTION
27 06195 106414 SUB# 0,1,5ZR ; ; ;
28 06196 106414 ERROR ; ; ;
29 06197 100030 XOP 0,0,0 ;ERROR CALL ;ERROR CALL
30 06198 100030 XOP 0,0,0 ;ERROR CALL ;ERROR CALL
31 06199 100030 XOP 0,0,0 ;ERROR CALL ;ERROR CALL
32 06200 006117 L\N127:  JSR ;ITERATE TEST ROUTINE ;ITERATE TEST ROUTINE
33 06201 006117 L\N127:  JSR ;ITERATE TEST ROUTINE ;ITERATE TEST ROUTINE

```

```

10162 ESPCL
01 000002 IZSZ2: ISZM=2
02 EISZ EISZ,IZC2,2,ADI
03
04 SETUP 100.
05 06256 006120 T\N131: INITIALIZE TEST
06 06257 000104 : ITERATION VALUE
07 RAND
08 06260 006121 JSR @ENTRA :C(ACO)=RANDOM #
09 06261 040264 STA 0,ISLC0 :C(ISLC0)=RANDOM DATA
10 LDRFAD
11 RAND
12 06262 006121 JSR @ENTRA :C(ACO)=RANDOM #
13 06263 006122 JSR @RNADR :GET A RANDOM ADDRESS IN
14 06264 000132 LDRHF :LOWER BUFFER
15 06265 000123 LDRFU
16 06266 040265 STA 0,ISLC1 :STORE RANDOM DATA INTO
17 06267 024264 LDA 1,ISLC0 :RANDOM ADDRESS 1
18 06270 046265 STA 1,ISLC1
19 HIRFAD
20 RAND
21 06271 006121 JSR @ENTRA :C(ACO)=RANDOM #
22 06272 006122 JSR @RNADR :GET A RANDOM ADDRESS IN
23 06273 000124 HIRBF :THE UPPER BUFFER
24 06274 000133 HIRFU
25 06275 111000 MOV 0,2 :STORE RANDOM ADDRESS 1 INTO
26 06276 024265 LDA 1,ISLC1 :RANDOM ADDRESS 2
27 06277 045000 MOVZR 0,2,5ZC :{RNDM ADDR 2/2}*BIT 15
28 06300 111222 ADI 1,2 :IS STORED IN AC2 AND
29 06301 110010 MLV 0 :ACO={RNDM ADDR 2}/2
30 06302 143370 ADDOR :ADD INDIRECT BIT TO ACO
31 06303 103240 STA 0,+2 :AND STORE IN EISZ
32 06304 040402 EISZ :EXECUTE EISZ INSTRUCTION
33 06305 113070 IZC2:
34 100000
35 06307 101000 MOV 0,0 :IN CASE IF EISZ SKIPS
36 06310 020264 LDA 0,ISLC0 :C(RNDM ADDR 1) MUST BE
37 06311 100010 ADI 1,0 :{RNDM DATA*1} FOR EISZ AND
38 06312 026265 LDA 1,ISLC1 :{RNDM DATA*1} FOR EOSZ
39 06313 106414 SUB# 0,1,5Z
40 ERROR
41 06314 100030 E\R207: XOP :ERROR CALL
42 LOOP
43 06315 006117 L\N131: JSR @ENTLO :ITERATE TEST ROUTINE

```

```

10161 FSPCL
01 IZ84:
02 SETUP 2.
03 IZ84:
04 06253 006120 T\N130: INITIALIZE TEST
05 06254 000002 : ITERATION VALUE
06 HIRFAD
07 RAND
08 06255 006121 JSR @ENTRA :C(ACO)=RANDOM #
09 06256 006122 JSR @RNADR :GET A RANDOM ADDRESS IN
10 06257 000124 HIRBF :THE UPPER BUFFER
11 06258 000133 HIRFU
12 06259 176400 SUB 3,3 :RNDM ADDR FROM HIRUFF
13 06260 176400 STA 3,ISLC0 :IS STORED IN EISZ INSTRUCTION
14 06261 054264 MOV 0,2 :
15 06262 055000 STA 3,+2 :C(RNDM ADDR) IS SET TO 0
16 06263 040402 EISZ :EISZ MUST INCREASE IT
17 06264 112070 IZ84:
18 000000
19 06240 101001 MOV 0,0,SKP :BY 1 EVERYTIME IT IS
20 06241 000411 JMP IZ84 :EXECUTED AND SKIP WHEN
21 06242 010264 ISZ ISLC0 :C(RNDM ADDR) BECOMES 0
22 06243 101001 MOV 0,0,SKP :C(ISLC0) KEEPS TRACK
23 ERROR
24 06244 100030 E\R208: XOP :ERROR CALL
25 06245 034264 LDA 3,ISLC0 :OF C(RNDM ADDR)
26 06246 025000 LDA 1,0,2 :
27 06247 136414 SUB# 1,3,5Z
28 ERROR
29 06250 100030 E\R205: XOP :ERROR CALL
30 06251 000765 JMP IZ84 :
31 06252 034264 IZ84: LDA 3,ISLC0 :C(ISLC0) MUST BE 17777
32 06253 174014 CGM# 3,3,5Z :AT THIS POINT
33 ERROR
34 06254 100030 E\R206: XOP :ERROR CALL
35 LOOP
36 06255 006117 L\N130: JSR @ENTLO :ITERATE TEST ROUTINE
37

```

```

:0163 ESPCL
01 000000 ISZ00: ISZSW=0
02 EISZ4 EISZ,IZ00,0,ADI,IZ00
03
04
05 SETUP 100,
06 06316 006120 TVN132: JSR @ENTIN ;INITIALIZE TEST
07 06317 000144 100, ;ITERATION VALUE
08 RAND
09 06320 006121 ;C(AC0)=RANDOM #
10 06321 040264 ;C(ISLC0)=RANDOM DATA
11 LOBFAD
12 RAND
13 06322 006121 JSR @ENTRA
14 06323 006122 JSR @RNADR
15 06324 000132 LOWBF ;GET A RANDOM ADDRESS IN
16 06325 000123 LOHFU ;LOWER BUFFER
17 06326 040265 STA 0,ISLC1
18 06327 024264 LDA 1,ISLC0
19 06330 046265 STA 1,@ISLC1
20 HIRFAD
21 RAND
22 06331 006121 JSR @ENTRA
23 06332 006122 JSR @RNADR
24 06333 000124 HIGBF ;GET A RANDOM ADDRESS IN
25 06334 000133 HIRFU ;THE UPPER BUFFER
26 06335 040266 STA 0,ISLC2
27 06336 024265 LDA 1,ISLC1
28 06337 046266 STA 1,@ISLC2
29 IZ000: HIRFAD
30 RAND
31 06340 006121 JSR @ENTRA
32 06341 006122 JSR @RNADR
33 06342 000124 HIGBF ;GET A RANDOM ADDRESS IN
34 06343 000133 HIRFU ;THE UPPER BUFFER
35 06344 024266 LDA 1,ISLC2
36 06345 106415 SUB# 0,1,SNR
37 06346 000772 JMP IZ000
38 06347 111000 MOV 0,2
39 06350 127240 ADDR, 1,1
40 06351 045000 STA 1,0,2
41 06352 000000 -NOLOC 0
42 06352 103240 ADDR, 0,0
43 06353 040402 STA 0,+2
44 06354 112070 IZ00: EISZ @0,0
45 06356 100000 MOV 0,0
46 06356 101000 LDA 0,ISLC0
47 06357 020264 LDA 1,0
48 06360 100010 ADI 1,0
49 06361 026265 LDA 1,@ISLC1
50 06362 106414 SUB# 0,1,SRZ
51 ERROR
52 06363 100030 ENR210: XOP 0,0,0
53 06364 006117 LWN132: LOOP
54 06364 006117 LWN132: JSR @ENTLO ;ITERATE TEST ROUTINE
55 06366 006117 LWN133: JSR @ENTLO ;ITERATE TEST ROUTINE

:0164 ESPCL
01 000001 ISZD1: ISZSW=1
02 EISZ4 EISZ,IZD1,1,ADI,IZD10
03
04
05 SETUP 100,
06 06365 006120 TVN133: JSR @ENTIN ;INITIALIZE TEST
07 06366 000144 100, ;ITERATION VALUE
08 RAND
09 06367 006121 JSR @ENTRA
10 06370 040264 STA 0,ISLC0
11 LOBFAD
12 RAND
13 06371 006121 JSR @ENTRA
14 06372 006122 JSR @RNADR
15 06373 000132 LOWBF ;GET A RANDOM ADDRESS IN
16 06374 000123 LOHFU ;LOWER BUFFER
17 06375 040265 STA 0,ISLC1
18 06376 024264 LDA 1,ISLC0
19 06377 046265 STA 1,@ISLC1
20 HIRFAD
21 RAND
22 06400 006121 JSR @ENTRA
23 06401 006122 JSR @RNADR
24 06402 000124 HIGBF ;GET A RANDOM ADDRESS IN
25 06403 000133 HIRFU ;THE UPPER BUFFER
26 06404 040266 STA 0,ISLC2
27 06405 024265 LDA 1,ISLC1
28 06406 046266 STA 1,@ISLC2
29 IZD10: HIRFAD
30 RAND
31 06407 006121 JSR @ENTRA
32 06411 000124 HIGBF ;GET A RANDOM ADDRESS IN
33 06412 000133 HIRFU ;THE UPPER BUFFER
34 06413 024266 LDA 1,ISLC2
35 06414 106415 SUB# 0,1,SNR
36 06415 000772 JMP IZD10
37 06416 111000 MOV 0,2
38 06417 127240 ADDR, 1,1
39 06420 045000 STA 1,0,2
40 06421 166470 ELEF 1,IZD1-,,1
41 06423 122400 SUB 0
42 06423 000000 -NOLOC 0
43 06424 103240 ADDR, 0,0
44 06425 040402 STA 0,+2
45 06426 112470 IZD1: EISZ @0,1
46 06430 100000 MOV 0,0
47 06430 101000 LDA 0,ISLC0
48 06431 020264 LDA 1,0
49 06432 100010 ADI 1,0
50 06433 026265 LDA 1,@ISLC1
51 06434 106414 SUB# 0,1,SRZ
52 ERROR
53 06435 100030 ENR211: XOP 0,0,0
54 06436 006117 LWN133: LOOP
55 06436 006117 LWN133: JSR @ENTLO ;ITERATE TEST ROUTINE

```



```

10165 ESPCL      000002  ISZD2:  ISZSW=2  EISZ4  EISZ,IZD2.2,ADI,IZD20
01 05 06437 006120 T\N134:  SETUP 100.  ;INITIALIZE TEST
02 06 06440 000144 RND 100.  ;ITERATION VALUE
03 07 06441 006121 JSR @ENTRA ;C(AC0)=RANDOM #
04 08 06442 040264 STA 0,ISLCO ;C(ISLCO)=RANDOM DATA
05 09 06443 006121 JSR @ENTRA ;C(AC0)=RANDOM #
06 10 06444 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
07 11 06445 000132 LOWBF ;FLOWER BUFFER
08 12 06446 000123 LOBFU ;C(ISLCO)=RANDOM ADDR 1
09 13 06447 040265 STA 1,ISLCO ;STORE RNDM DATA INTO
10 14 06448 024264 LDA 1,@ISLCO ;RNDM ADDR 1
11 15 06449 046265 STA 1,@ISLCO ;RNDM ADDR 1
12 16 06450 006121 RNDM ;
13 17 06451 006122 JSR @ENTRA ;C(AC0)=RANDOM #
14 18 06452 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
15 19 06453 000124 HIGHF ;THE UPPER BUFFER
16 20 06454 000133 LOBFU ;C(ISLCO)=RANDOM ADDR 2
17 21 06455 040266 STA 0,ISLCO ;STORE RNDM DATA INTO
18 22 06456 024265 LDA 1,@ISLCO ;RNDM ADDR 2
19 23 06457 046266 STA 1,@ISLCO ;RNDM ADDR 2
20 24 06458 006121 RNDM ;
21 25 06459 006122 JSR @ENTRA ;C(AC0)=RANDOM #
22 26 06460 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
23 27 06461 000124 HIGHF ;THE UPPER BUFFER
24 28 06462 000133 LOBFU ;C(ISLCO)=RANDOM ADDR 1
25 29 06463 040266 STA 0,ISLCO ;STORE RNDM DATA INTO
26 30 06464 024265 LDA 1,@ISLCO ;RNDM ADDR 1
27 31 06465 046266 STA 1,@ISLCO ;RNDM ADDR 1
28 32 06466 006121 RNDM ;
29 33 06467 006122 JSR @ENTRA ;C(AC0)=RANDOM #
30 34 06468 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
31 35 06469 000124 HIGHF ;THE UPPER BUFFER
32 36 06470 000133 LOBFU ;C(ISLCO)=RANDOM ADDR 2
33 37 06471 040266 STA 0,ISLCO ;STORE RNDM DATA INTO
34 38 06472 024265 LDA 1,@ISLCO ;RNDM ADDR 2
35 39 06473 046266 STA 1,@ISLCO ;RNDM ADDR 2
36 40 06474 110000 MOV 0,2 ;AND SAVE @RNDM ADDR 2
37 41 06475 110000 ADDOR 1,1 ;IN RNDM ADDR 3
38 42 06476 127240 STA 1,0,2 ;
39 43 06477 045000 MOVZR 0,2,SZC ;(RNDM ADDR 3/2)+BIT 15 IS
40 44 06478 112222 ADI 1,2 ;STORED IN AC2 AND
41 45 06479 110010 HLV 0 ;FAC0=(RNDM ADDR 3/2)
42 46 06480 143370 -NOLCO ;
43 47 06481 000000 ADDOR 0,0 ;ADD INDIRECT BIT TO AC0
44 48 06482 040402 STA 0,*,2 ;AND STORE IN EISZ
45 49 06483 113070 EISZ ;EXECUTE EISZ INSTRUCTION
46 50 06484 100000 ;
47 51 06485 101000 MOV 0,0 ;
48 52 06486 020264 LDA 0,ISLCO ;C(RNDM ADDR 1) MUST BE
49 53 06487 100010 ADI 1,0 ;=(RNDM DATA+1) FOR EISZ
50 54 06488 026265 LDA 0,ISLCO ;AND =(RNDM DATA-1) FOR EDSZ
51 55 06489 106414 SUB# 0,1,SZR ;
52 56 06490 000000 ERROR ;
53 57 06491 100030 ENR212: XOP 0,0,0 ;ERROR CALL
54 58 06492 100030 LOOP ;
55 59 06493 006117 L\N134: JSR @ENTLO ;ITERATE TEST ROUTINE
56 60 06494 006117 L\N135: JSR @ENTLO ;ITERATE TEST ROUTINE

```

```

10166 ESPCL      000003  ISZD3:  ISZSW=3  EISZ4  EISZ,IZD3.3,ADI,IZD30
01 05 06511 006120 T\N135:  SETUP 100.  ;INITIALIZE TEST
02 06 06512 000144 RND 100.  ;ITERATION VALUE
03 07 06513 006121 JSR @ENTRA ;C(AC0)=RANDOM #
04 08 06514 040264 STA 0,ISLCO ;C(ISLCO)=RANDOM DATA
05 09 06515 006121 JSR @ENTRA ;C(AC0)=RANDOM #
06 10 06516 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
07 11 06517 000132 LOWBF ;FLOWER BUFFER
08 12 06518 000123 LOBFU ;C(ISLCO)=RANDOM ADDR 1
09 13 06519 040265 STA 1,ISLCO ;STORE RNDM DATA INTO
10 14 06520 024264 LDA 1,@ISLCO ;RNDM ADDR 1
11 15 06521 046265 STA 1,@ISLCO ;RNDM ADDR 1
12 16 06522 006121 RNDM ;
13 17 06523 006122 JSR @ENTRA ;C(AC0)=RANDOM #
14 18 06524 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
15 19 06525 000124 HIGHF ;THE UPPER BUFFER
16 20 06526 000133 LOBFU ;C(ISLCO)=RANDOM ADDR 2
17 21 06527 040266 STA 0,ISLCO ;STORE RNDM DATA INTO
18 22 06528 024265 LDA 1,@ISLCO ;RNDM ADDR 2
19 23 06529 046266 STA 1,@ISLCO ;RNDM ADDR 2
20 24 06530 006121 RNDM ;
21 25 06531 006122 JSR @ENTRA ;C(AC0)=RANDOM #
22 26 06532 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
23 27 06533 000124 HIGHF ;THE UPPER BUFFER
24 28 06534 000133 LOBFU ;C(ISLCO)=RANDOM ADDR 1
25 29 06535 040266 STA 0,ISLCO ;STORE RNDM DATA INTO
26 30 06536 024265 LDA 1,@ISLCO ;RNDM ADDR 1
27 31 06537 046266 STA 1,@ISLCO ;RNDM ADDR 1
28 32 06538 006121 RNDM ;
29 33 06539 006122 JSR @ENTRA ;C(AC0)=RANDOM #
30 34 06540 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
31 35 06541 000124 HIGHF ;THE UPPER BUFFER
32 36 06542 000133 LOBFU ;C(ISLCO)=RANDOM ADDR 2
33 37 06543 040266 STA 0,ISLCO ;STORE RNDM DATA INTO
34 38 06544 024265 LDA 1,@ISLCO ;RNDM ADDR 2
35 39 06545 046266 STA 1,@ISLCO ;RNDM ADDR 2
36 40 06546 110000 MOV 0,2 ;AND SAVE @RNDM ADDR 2
37 41 06547 110000 ADDOR 1,1 ;IN RNDM ADDR 3
38 42 06548 127240 STA 1,0,2 ;
39 43 06549 045000 MOVZR 0,3,SZC ;(RNDM ADDR 3/2)+BIT 15 IS
40 44 06550 112222 ADI 1,3 ;STORED IN AC3 AND
41 45 06551 110010 HLV 0 ;FAC0=(RNDM ADDR 3/2)
42 46 06552 143370 -NOLCO ;
43 47 06553 000000 ADDOR 0,0 ;ADD INDIRECT BIT TO AC0
44 48 06554 040402 STA 0,*,2 ;AND STORE IN EISZ
45 49 06555 113470 EISZ ;EXECUTE EISZ INSTRUCTION
46 50 06556 100000 ;
47 51 06557 101000 MOV 0,0 ;
48 52 06558 020264 LDA 0,ISLCO ;C(RNDM ADDR 1) MUST BE
49 53 06559 100010 ADI 1,0 ;=(RNDM DATA+1) FOR EISZ
50 54 06560 026265 LDA 0,ISLCO ;AND =(RNDM DATA-1) FOR EDSZ
51 55 06561 106414 SUB# 0,1,SZR ;
52 56 06562 000000 ERROR ;
53 57 06563 100030 ENR213: XOP 0,0,0 ;ERROR CALL
54 58 06564 100030 LOOP ;
55 59 06565 006117 L\N135: JSR @ENTLO ;ITERATE TEST ROUTINE
56 60 06566 006117 L\N135: JSR @ENTLO ;ITERATE TEST ROUTINE

```

```

10167  ESPCL
01 000000 DSZB0: ISZSW=0
02 EISZ2 EDSZ,LORFAD,DZ80,0,SBI
03
04
05
06
07
08
09 06563 006120 TAN136: JSR #ENTIN ;INITIALIZE TEST
10 06564 000144 100. ;ITERATION VALUE
11 LORFAD
12 RAND
13 JSR #ENTRA ;C(ACO)=RANDOM #
14 JSR #RNADR ;GET A RANDOM ADDRESS IN
15 LORBF ;LOWER BUFFER
16 LORFU
17 STA 0,ISLCO ;EDSZ INSTRUCTION SHOULD
18 JSR #ENTRA ;NOT CHANGE ANY AC'S
19 SUB 0,0 ;STATE OF CARRY
20 STA 0,ISLCO ;
21 RAND
22 JSR #ENTRA ;C(ACO)=RANDOM #
23 MOV 0,3 ;ACO=AC3=RANDOM DATA
24 NEG 0,1 ;AC1=AC2=(RANDOM DATA)
25 MOVZ 1,2 ;CARRY IS INITIALISED
26 EDSZ 0,0
27
28 SUB# 0,3,SZR
29 ERROR
30 06604 100030 EAR214: XOP 0,0,0 ;ERROR CALL
31 06605 132414 SUB# 1,2,SZR ;
32 ERROR
33 06606 100030 EAR215: XOP 0,0,0 ;ERROR CALL
34 06607 101002 MOV 0,0,SZC ;
35 06610 100030 EAR216: XOP 0,0,0 ;ERROR CALL
36
37 LOOP
38 06611 006117 LAN136: JSR #ENTLO ;ITERATE TEST ROUTINE

```

```

10168  ESPCL
01 000000 DSZB0: ISZSW=0
02 EISZ2 EDSZ,LORFAD,DZ80,0,SBI
03
04
05
06
07
08
09 06612 006120 TAN137: JSR #ENTIN ;INITIALIZE TEST
10 06613 000144 100. ;ITERATION VALUE
11 LORFAD
12 RAND
13 JSR #ENTRA ;C(ACO)=RANDOM #
14 JSR #RNADR ;GET A RANDOM ADDRESS IN
15 LORBF ;LOWER BUFFER
16 LORFU
17 STA 0,ISLCO ;EDSZ INSTRUCTION SHOULD
18 JSR #ENTRA ;NOT CHANGE ANY AC'S
19 SUB 0,0 ;STATE OF CARRY
20 STA 0,ISLCO ;
21 RAND
22 JSR #ENTRA ;C(ACO)=RANDOM #
23 MOV 0,0 ;C(RNDM ADDR) MUST BE
24 LDA 0,ISLCO ;=(RNDM DATA+1) FOR EISZ
25 SBI 1,0 ;AND =(RNDM DATA-1) FOR
26 LOA 1,ISLCO ;EDSZ INSTRUCTION
27 SUB# 0,1,SZR ;
28 ERROR
29 XOP 0,0,0 ;ERROR CALL
30 06635 100030 EAR217: XOP 0,0,0 ;ERROR CALL
31
32 LOOP
33 06636 006117 LAN137: JSR #ENTLO ;ITERATE TEST ROUTINE

```

```

10170 FSPCL
01 000001 0SZ81: ISZSW=1
02 EISZ2 EDSZ.HI8FAD,DZ81,1.SB1
03
04 SETUP 100.
05 JSR @ENTRIN ;INITIALIZE TEST
06 06640 000144 ;ITERATION VALUE
07 RAND
08 06641 006121 ;C(AC0)=RANDOM #
09 06642 040264 ;C(ISLCO)=RANDOM DATA
10 HI8FAD
11 RAND
12 06643 006121 ;C(AC0)=RANDOM #
13 06644 006122 ;GET A RANDOM ADDRESS IN
14 06645 000124 ;THE UPPER BUFFER
15 06646 000133
16 06647 040265 ;C(ISLCO)=RANDOM ADDRESS
17 06650 024264 ;STORE RANDOM DATA IN
18 06651 046265 ;RANDOM ADDRESS
19 06652 166470 ELEF 1,DZ81-.,1 ;(RNDM ADDR-PC OF EDSZ*1)
20 06654 122400 SUB 1,0,+2 ;IS STORED IN EDSZ
21 06655 040402 STA 0,+2 ;
22 06656 116470 DZ81: EDSZ 0,1 ;EXECUTE EDSZ INSTRUCTION
23 000000
24 06660 101000 MOV 0,0 ;C(RNDM ADDR) MUST BE
25 06661 020264 LDA 0,ISLCO ;=(RNDM DATA*1) FOR EISZ
26 06662 100110 SRI 1,0 ;AND =(RNDM DATA-1) FOR
27 06663 026265 LDA 1,@ISLCO ;EDSZ INSTRUCTION
28 06664 106414 SUB# 0,1,SZR ;
29 ERROR
30 06665 100030 E\VR220: YDP 0,0,0 ;ERROR CALL
31 LOOP
32 06666 006117 L\N140: JSR @ENTLO ;ITERATE TEST ROUTINE

```

```

10170 FSPCL
01 000002 0SZ82: ISZSW=2
02 EISZ2 EDSZ.HI8FAD,DZ82,2.SB1
03
04 SETUP 100.
05 JSR @ENTRIN ;INITIALIZE TEST
06 06670 000144 ;ITERATION VALUE
07 RAND
08 06671 006121 ;C(AC0)=RANDOM #
09 06672 040264 ;C(ISLCO)=RANDOM DATA
10 HI8FAD
11 RAND
12 06673 006121 ;C(AC0)=RANDOM #
13 06674 006122 ;GET A RANDOM ADDRESS IN
14 06675 000124 ;THE UPPER BUFFER
15 06676 000133
16 06677 040265 ;C(ISLCO)=RANDOM ADDRESS
17 06700 024264 ;STORE RANDOM DATA IN
18 06701 046265 ;RANDOM ADDRESS
19 06702 111222 MOVZR 0,2,SZC ;(RNDM ADDR/2)+BIT 15
20 06703 110010 ADI 1,2 ;IS STORED IN AC2 AND
21 06704 143370 HLV 0 ;
22 06705 040402 STA 0,+2 ;
23 06706 117070 DZ82: EDSZ 0,2 ;EXECUTE EDSZ INSTRUCTION
24 000000
25 06710 101000 MOV 0,0 ;C(RNDM ADDR) MUST BE
26 06711 020264 LDA 0,ISLCO ;=(RNDM DATA*1) FOR EISZ
27 06712 100110 SRI 1,0 ;AND =(RNDM DATA-1) FOR
28 06713 026265 LDA 1,@ISLCO ;EDSZ INSTRUCTION
29 06714 106414 SUB# 0,1,SZR ;
30 ERROR
31 06715 100030 E\VR221: XOP 0,0,0 ;ERROR CALL
32 06716 006117 L\N141: JSR @ENTLO ;ITERATE TEST ROUTINE

```

```

10171 ESPL
01 000003 05ZB3: 1S7SW=3
02 01S2Z  EDSZ,LOBFAD,DZB3,3,SBI
03
04 06717 006120 T\N142: 100.
05 06720 000144 0.
06 06721 006121 JSR @ENTRA
07 06722 040264 STA 0,ISLCO
08 06723 006121 JSR @ENTRA
09 06724 006122 JSR @RNADR
10 06725 000132 LOBF
11 06726 000123 LOBFU
12 06727 040265 STA 0,ISLCO
13 06728 024264 LDA 1,ISLCO
14 06729 040265 MOVZ 0,3,SZC ;(RNDM ADDR/2)*8BIT IS
15 06730 114010 ADI 1,3
16 06731 143370 MVL 0
17 06732 040402 STA 0,+2
18 06733 117470 DZB3: EDSZ 0,3
19 06734 000000 MOV 0,0
20 06735 101000 LDA 0,ISLCO
21 06736 020264 SBI 1,0
22 06737 026265 LDA 0,1,ISLCO
23 06738 106414 SUB# 0,1,SZR
24 06739 100030 ERROR
25 06740 100030 XOP
26 06741 020264 LDA 0,ISLCO
27 06742 100110 SBI 1,0
28 06743 026265 LDA 0,1,ISLCO
29 06744 106414 SUB# 0,1,SZR
30 06745 100030 ERROR
31 06746 100030 XOP
32 06747 100030 E\R223: XOP
33 06748 006117 L\N142: JSR @ENTLO ;ITERATE TEST ROUTINE
34
35
36
37
38

```

```

10172 ESPL
01
02
03
04 06707 006120 T\N143: 2.
05 06730 000002 0.
06 06731 000002 LOBFAD
07 06732 000002 RAND
08 06733 006121 JSR @ENTRA
09 06734 006122 JSR @RNADR
10 06735 000132 LOBF
11 06736 000123 LOBFU
12 06737 176400 SUB 3,3
13 06738 054264 STA 3,ISLCO
14 06739 111000 MOV 0,2
15 06740 055000 STA 3,0,2
16 06741 040402 STA 0,+2
17 06742 116070 DZB4: EDSZ 0,0
18 06743 000000
19 06744 101001 MOV 0,0,SKP
20 06745 000411 JMP 0ZB40
21 06746 014264 DSZ ISLCO
22 06747 101001 MOV 0,0,SKP
23 06748 100030 ERROR
24 06749 100030 E\R223: XOP
25 06750 034264 LDA 3,ISLCO
26 06751 025000 LDA 1,0,2
27 06752 136414 SUB# 1,3,SZR
28 06753 100030 ERROR
29 06754 100030 E\R224: XOP
30 06755 000765 JMP 0ZB4
31 06756 034264 DZB40: LDA 3,ISLCO
32 06757 020276 LDA 0,ONE
33 06758 162414 SUB# 3,0,SZR
34 06759 100030 ERROR
35 06760 100030 XOP
36 06761 100030 E\R225: XOP
37 06762 006117 L\N143: JSR @ENTLO ;ITERATE TEST ROUTINE
38

```

```

10173 ESPL
01
02
03
04 06707 006120 T\N143: 2.
05 06730 000002 0.
06 06731 000002 LOBFAD
07 06732 000002 RAND
08 06733 006121 JSR @ENTRA
09 06734 006122 JSR @RNADR
10 06735 000132 LOBF
11 06736 000123 LOBFU
12 06737 176400 SUB 3,3
13 06738 054264 STA 3,ISLCO
14 06739 111000 MOV 0,2
15 06740 055000 STA 3,0,2
16 06741 040402 STA 0,+2
17 06742 116070 DZB4: EDSZ 0,0
18 06743 000000
19 06744 101001 MOV 0,0,SKP
20 06745 000411 JMP 0ZB40
21 06746 014264 DSZ ISLCO
22 06747 101001 MOV 0,0,SKP
23 06748 100030 ERROR
24 06749 100030 E\R223: XOP
25 06750 034264 LDA 3,ISLCO
26 06751 025000 LDA 1,0,2
27 06752 136414 SUB# 1,3,SZR
28 06753 100030 ERROR
29 06754 100030 E\R224: XOP
30 06755 000765 JMP 0ZB4
31 06756 034264 DZB40: LDA 3,ISLCO
32 06757 020276 LDA 0,ONE
33 06758 162414 SUB# 3,0,SZR
34 06759 100030 ERROR
35 06760 100030 XOP
36 06761 100030 E\R225: XOP
37 06762 006117 L\N143: JSR @ENTLO ;ITERATE TEST ROUTINE
38

```

```

:0173 ESPCL      000001 0SZC1:  ISZSM=1  EDZS,DZC1,1,SBI
01 01 000000 0SZ10:  EISZ4      EDZS,DZ00,0,SBI,DZ00
02 02 000000 0SZ10:  EISZ4      EDZS,DZ00,0,SBI,DZ00
03 03 000000 0SZ10:  EISZ4      EDZS,DZ00,0,SBI,DZ00
04 04 000000 0SZ10:  EISZ4      EDZS,DZ00,0,SBI,DZ00
05 05 07003 006120 T\N144:  SETUP 100.  ;INITIALIZE TEST
06 06 07004 000144  ;JSR @ENTRA ;ITERATION VALUE
07 07 07004 000144  RAND 100.  ;ITERATION VALUE
08 08 07005 006121  ;JSR @ENTRA ;C(AC0)=RANDOM #
09 09 07006 040264  STA 0,ISLCO ;C(ISLCO)=RANDOM DATA
10 10 07006 040264  LOBFAD
11 11 07007 006121  RAND
12 12 07007 006121  ;JSR @ENTRA ;C(AC0)=RANDOM #
13 13 07010 006122  JSR @RNADR ;GET A RANDOM ADDRESS IN
14 14 07011 000132  LOWBF ;LOWER BUFFER
15 15 07012 000132  LOBFU
16 16 07013 040265  STA 0,ISLCO ;STORE RANDOM DATA INTO
17 17 07014 024264  LDA 1,ISLCO ;RANDOM ADDRESS 1
18 18 07015 044265  STA 1,@ISLCO
19 19 07015 044265  HIBFAD
20 20 07016 006121  RAND
21 21 07016 006121  ;JSR @ENTRA ;C(AC0)=RANDOM #
22 22 07017 006122  JSR @RNADR ;GET A RANDOM ADDRESS IN
23 23 07020 000124  HIBF ;THE UPPER BUFFER
24 24 07021 000133  MOV 0,2
25 25 07022 111000  LDA 1,ISLCO ;STORE RANDOM ADDRESS 1 INTO
26 26 07023 04265  STA 1,0+2 ;RANDOM ADDRESS 2
27 27 07024 045000  ELEF 1,DZC1-.+1
28 28 07025 166470  SUB 1,0 ;(RNDM ADDR 2-PC OF EDZ+1)
29 29 07027 122400  ADDOR 0,0 ;IS IN ACO
30 30 07030 103240  STA 0,+2 ;ADD INDIRECT BIT TO ACO
31 31 07031 040402  STA 20,1 ;AND STORE IN EDZ
32 32 07032 116470  OZC1:  EDZ ;EXECUTE EDZ INSTRUCTION
33 33 100000
34 34 07034 101000  MOV 0,0 ;IN CASE IF EDZ SKIPS
35 35 07035 020264  LDA 0,ISLCO ;C(RNDM ADDR 1) MUST BE
36 36 07036 100110  SBI 1,0 ;C(RNDM DATA+1) FOR EISZ AND
37 37 07037 026265  LDA 1,@ISLCO ;(RNDM DATA-1) FOR EDZ
38 38 07040 106414  SUR# 0,1,SZP
39 39 07040 106414  ERROR
40 40 07041 100030  E\R226:  XOP ;ERROR CALL
41 41 07042 006117  L\N144:  JSR @ENTLO ;ITERATE TEST ROUTINE
42 42 07042 006117  L\N144:  JSR @ENTLO ;ITERATE TEST ROUTINE
43 43 07100 040402  STA 0,+2 ;ADD STORE IN EDZ
44 44 07101 116070  OZD00:  EDZ ;EXECUTE EDZ INSTRUCTION
45 45 100000
46 46 07103 101000  MOV 0,0 ;C(RNDM ADDR 1) MUST BE
47 47 07104 020264  LDA 0,ISLCO ;C(RNDM DATA+1) FOR EISZ
48 48 07105 100110  SBI 1,0 ;C(RNDM DATA-1) FOR EDZ
49 49 07106 026265  LDA 1,@ISLCO
50 50 07107 106414  SUR# 0,1,SZP
51 51 07110 100030  E\R227:  XOP ;ERROR CALL
52 52 07110 100030  E\R227:  XOP ;ERROR CALL
53 53 07111 006117  L\N145:  JSR @ENTLO ;ITERATE TEST ROUTINE
54 54 07111 006117  L\N145:  JSR @ENTLO ;ITERATE TEST ROUTINE

```

```

:0175 ESPCL      DSZD1:  ISZSW=1      EDSZ,DZD1,1,SBI,DZD10
01 000001 000001 000001 000001 000001 000001 000001 000001 000001
02 E1SZ4      EDSZ,DZD1,1,SBI,DZD10
03
04
05 07112 006120 1N146:  INITIALIZE TEST
06 07113 000140 100:      ITERATION VALUE
07
08 07114 006121  JSR @ENTRA      ;C(AC0)=RANDOM #
09 07115 040264  STA 0,ISLCO      ;C(ISLCO)=RANDOM DATA
10 LOBFAD
11 RAND
12 07116 006121  JSR @ENTRA      ;C(AC0)=RANDOM #
13 07117 006122  JSR @RNADR      ;GET A RANDOM ADDRESS IN
14 07120 000132  LOBF          ;LOWER BUFFER
15 07121 000123  LOBFU
16 07122 040265  STA 0,ISLC1
17 07123 024264  LDA 1,ISLCO
18 07124 046265  STA 1,@ISLC1
19 HIRFAD
20 RAND
21 07125 006121  JSR @ENTRA      ;C(AC0)=RANDOM #
22 07126 006122  JSR @RNADR      ;GET A RANDOM ADDRESS IN
23 07127 000132  HIGBF          ;THE UPPER BUFFER
24 07130 000133  HIRFU
25 07131 040266  STA 0,ISLC2
26 07132 024265  LDA 1,ISLC1
27 07133 046266  STA 1,@ISLC2
28
29
30 07134 006121  RAND
31 07135 006122  JSR @ENTRA      ;C(AC0)=RANDOM #
32 07136 000124  HIGBF          ;GET A RANDOM ADDRESS IN
33 07137 000133  HIRFU          ;THE UPPER BUFFER
34 07140 024266  LDA 1,ISLC2
35 07141 064415  SUB#           ;MAKE SURE THAT RNDM ADDR
36 07142 000772  JMP           ;2 AND 3 ARE DIFFERENT
37 07143 110000  MOV 0,2
38 07144 127240  ADDR 1,1
39 07145 045000  STA 1,0,2
40 07146 166470  ELEF 1,DZD1=..1
41 07150 122400  SUB 1,0
42 07151 060000  WLOC
43 07152 103240  ADDR 0,0
44 07153 040402  STA 0,+2
45 07153 116470  DZD1:  EDSZ
46 07155 100000  MOV 0,0
47 07155 101000  LDA 0,ISLCO
48 07156 020264  SBI 1,0
49 07157 100110  LDA 1,@ISLC1
50 07160 026265  SUB#          ;AC0=(RNDM ADDR 3-PC OF
51 07161 106414  ERROR        ;EDSZ+1)
52 07162 100030  EDROR
53 07162 100030  EDR250:  XOP
54
55 07163 006117  LWN146:  JSR @ENTLO ;ITERATE TEST ROUTINE

```

```

:0176 ESPCL      DSZD2:  ISZSW=2      EDSZ,DZD2,2,SBI,DZD20
01 000002 000002 000002 000002 000002 000002 000002 000002 000002
02 E1SZ4      EDSZ,DZD2,2,SBI,DZD20
03
04
05 07164 006120 1N147:  INITIALIZE TEST
06 07165 000144 100:      ITERATION VALUE
07
08 07166 006121  JSR @ENTRA      ;C(AC0)=RANDOM #
09 07167 040264  STA 0,ISLCO      ;C(ISLCO)=RANDOM DATA
10 LOBFAD
11 RAND
12 07170 006121  JSR @ENTRA      ;C(AC0)=RANDOM #
13 07171 006122  JSR @RNADR      ;GET A RANDOM ADDRESS IN
14 07172 000132  LOBF          ;LOWER BUFFER
15 07173 000123  LOBFU
16 07174 040265  STA 0,ISLC1
17 07175 024264  LDA 1,ISLCO
18 07176 046265  STA 1,@ISLC1
19 HIRFAD
20 RAND
21 07177 006121  JSR @ENTRA      ;C(AC0)=RANDOM #
22 07178 006122  JSR @RNADR      ;GET A RANDOM ADDRESS IN
23 07201 000132  HIGBF          ;THE UPPER BUFFER
24 07202 000133  HIRFU
25 07203 040266  STA 0,ISLC2
26 07204 024265  LDA 1,ISLC1
27 07205 046266  STA 1,@ISLC2
28
29
30 07206 006121  RAND
31 07207 006122  JSR @ENTRA      ;C(AC0)=RANDOM #
32 07210 000124  HIGBF          ;GET A RANDOM ADDRESS IN
33 07211 000133  HIRFU          ;THE UPPER BUFFER
34 07212 024266  LDA 1,ISLC2
35 07213 064415  SUB#           ;MAKE SURE THAT RNDM ADDR
36 07214 000772  JMP           ;2 AND 3 ARE DIFFERENT
37 07215 110000  MOV 0,2
38 07216 127240  ADDR 1,1
39 07217 045000  STA 1,0,2
40 07220 111222  MOVZR 0,2,SZC ;(RNDM ADDR 3/2)+BIT 15 IS
41 07221 110010  ADI 1,2
42 07222 143370  HLV 0
43 07223 060000  WLOC
44 07223 103240  ADDR 0,0
45 07224 040402  STA 0,+2
46 07225 117070  DZD2:  EDSZ
47 07225 100000  MOV 0,0
48 07227 101000  LDA 0,ISLCO
49 07230 020264  SBI 1,0
50 07231 100110  LDA 1,@ISLC1
51 07232 026265  SUB#          ;AC0=(RNDM ADDR 3/2)
52 07233 106414  ERROR        ;AND STORE IN EDSZ
53 07234 100030  EDROR
54 07234 100030  EDR231:  XOP
55 07235 006117  LWN147:  JSR @ENTLO ;ITERATE TEST ROUTINE

```

```

10177 ESPCL
01 000003 DSZD3: ISZSN=3
02 EISZ4 ED SZ,DZ03,3,SBI,DZ030
03
04 SETUP 100.
05 JSR @ENTIN ;INITIALIZE TEST
06 000144 ;ITERATION VALUE
07 RAND
08 07240 006121 JSR @ENTRA
09 07241 040264 STA 0,I,ISLC0 ;C(AC0)=RANDOM #
10 LOBFAD ;C(ISLC0)=RANDOM DATA
11 RAND
12 07242 006121 JSR @ENTRA
13 07243 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
14 07244 000132 LOBF ;LOWER BUFFER
15 07245 000123 STA 0,I,ISLC1 ;C(ISLC1)=RANDOM ADDR 1
16 07246 040265 LDA 1,ISLC0 ;STORE RNDM DATA INTO
17 07247 024264 STA 1,@ISLC1 ;RNDM ADDR 1
18 07250 046265 HIBFAD
19 RAND
20
21 07251 006121 JSR @ENTRA
22 07252 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
23 07253 000124 HIBF ;THE UPPER BUFFER
24 07254 000133 STA 0,I,ISLC2 ;C(ISLC2)=RANDOM ADDR 2
25 07255 040266 LDA 1,ISLC1 ;STORE RNDM ADDR 1 IN
26 07256 024265 STA 1,@ISLC2 ;RNDM ADDR 2
27 07257 046266 HIBFAD
28 RAND
29
30 07260 006121 JSR @ENTRA
31 07261 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
32 07262 000124 HIBF ;THE UPPER BUFFER
33 07263 000133 STA 0,I,ISLC2 ;MAKE SURE THAT RNDM ADDR
34 07264 024266 LDA 1,ISLC1 ;2 AND 3 ARE DIFFERENT
35 07265 106415 SUB# DZ030 ;
36 07266 000772 JMP 0,2 ;AND SAVE @RNDM ADDR 2
37 07267 111000 MOV 0,2 ;IN RNDM ADDR 3
38 ADDOR 1,1
39 07270 127240 STA 1,0,2
40 07271 045000 STA 0,3,SZC ;(RNDM ADDR 3/2)*BIT 15 IS
41 07272 115222 MOVZR ;STORED IN AC3 AND
42 07273 114010 ADI 1,3 ;ACO=(RNDM ADDR 3/2)
43 07274 143370 HLV 0
44 07275 000000 ;NOLOC
45 07276 103240 ADDOR 0,0
46 07277 117470 STA 0,+2 ;AND STORE IN ED SZ
47 07278 117470 ED SZ ;EXECUTE ED SZ INSTRUCTION
48 07301 101000 MOV 0,0
49 07302 020264 LDA 0,I,ISLC0 ;C(RNDM ADDR 1) MUST BE
50 07303 100110 SBI 1,0 ;=(RNDM DATA+1) FOR EISZ
51 07304 026265 LDA 1,@ISLC1 ;AND =(RNDM DATA-1) FOR ED SZ
52 07305 106414 SHM# 0,1,SZR
53 ERROR
54 07306 100930 E\@R23: XOP ;ERROR CALL
55 LOOP
56 07307 006117 L\@N151: JSR @ENTLO ;ITERATE TEST ROUTINE
57
58 07357 100030 E\@R23: XOP ;ERROR CALL
59 LOOP
60 07360 006117 L\@N151: JSR @ENTLO ;ITERATE TEST ROUTINE

```

```

10178 ESPCL
01 000003 DSZD3: ISZSN=3
02 EISZ4 ED SZ,DZ03,3,SBI,DZ030
03
04 SETUP 100.
05 JSR @ENTIN ;INITIALIZE TEST
06 000144 ;ITERATION VALUE
07 RAND
08 07310 006125 JSR @ENTRA
09 07311 177777 STA 0,I,ISLC0 ;C(AC0)=RANDOM #
10 LOBFAD ;C(ISLC0)=RANDOM DATA
11 RAND
12 07312 006125 JSR @ENTRA
13 07313 000144 JSR @RNADR ;GET A RANDOM ADDRESS IN
14 07314 020132 LOBF ;LOWER BUFFER
15 07315 024123 STA 0,I,ISLC1 ;C(ISLC1)=RANDOM ADDR 1
16 07316 106400 LDA 1,ISLC0 ;STORE RNDM DATA INTO
17 07317 044301 STA 1,@ISLC1 ;RNDM ADDR 1
18 07318 044301 HIBFAD
19 RAND
20
21 07320 006121 JSR @ENTRA
22 07321 006122 JSR @RNADR ;GET A RANDOM ADDRESS IN
23 07322 000304 HIBF ;THE UPPER BUFFER
24 07323 000301 STA 0,I,ISLC2 ;C(ISLC2)=RANDOM ADDR 2
25 07324 120110 LDA 1,ISLC1 ;STORE RNDM ADDR 1 IN
26 07325 040302 STA 1,@ISLC2 ;RNDM ADDR 2
27 07326 006121 JSR @ENTRA
28 07327 030132 JSR @RNADR ;GET A RANDOM ADDRESS IN
29 07328 041000 HIBF ;THE UPPER BUFFER
30 07329 024302 STA 0,I,ISLC2 ;MAKE SURE THAT RNDM ADDR
31 07330 107000 ADD 0,1 ;ACTUAL TABLE BEGIN ADDR
32 07331 104110 STA 1,1
33 07332 104110 STA 1,1,2
34 07333 045001 HLV 0
35 07334 143370 CLM 1,2
36 07335 152370 JMP *-8. ;C(TABLE-1)
37 07337 000770 RAND
38
39 07340 006121 JSR @ENTRA
40 07341 024132 LDA 1,L,OWBF ;GET RANDOM DATA 2 IN
41 07342 106370 CLM 0,1 ;THE RANGE OF C(TABLE-2)
42 07343 000775 JMP *-3 ;AND C(TABLE-1)
43 07344 130010 ADI 2,2 ;ACO=DISPATCH ADDEND
44 07345 040267 STA 0,D,SLC0 ;AC2=TABLE BEGIN ADDR
45 07346 006121 RAND
46 JSR @ENTRA ;C(AC0)=RANDOM #
47 07347 114400 HEG 0,3 ;AC3=-(RNDM DATA)
48 07348 105040 MOVU 0,1 ;AC1= RNDM DATA
49 07349 020267 LDA 0,D,SLC0 ;CARRY IS SET TO 1
50 07350 143170 DSPA 0,0,2 ;DSPA SHOULD NOT CHANGE
51 07351 000000 MOV 0,0,SNC
52 07352 101003 ERROR
53 07353 100030 E\@R23: XOP ;ERROR CALL
54 07354 137014 ADD# 1,5,SZR
55 07355 100030 E\@R23: XOP ;ERROR CALL
56 07356 137014 ADD# 1,5,SZR
57 07357 100030 E\@R23: XOP ;ERROR CALL
58 07358 100030 E\@R23: XOP ;ERROR CALL
59 07359 100030 E\@R23: XOP ;ERROR CALL
60 07360 006117 L\@N151: JSR @ENTLO ;ITERATE TEST ROUTINE

```

```

01 DSPAL: EDSP1  HIGBF,HIBFU,17777
02
03
04 FILL 17777          ;FILL UPPER AND LOWER
05 JSR @JFILL        ;SCRATCH BUFFER AREA
06 07361 006125     ;WITH (17777)
07 07362 17777
08
09 SETUP 100.
10 @ENTIN ;INITIALIZE TEST
11 @ENTIN ;ITERATION VALUE
12 0, HIGBF ;MAX. AVAILABLE AREA
13 1, HIBFU ;FOR TABLE IS =
14 0, 1 ;C(HIBFU)=C(HIGBF)
15
16 RADADR 03,MXTHL
17 07371 006121     ;C(AC0)=RANDOM #
18 07372 006122     ;GET ADDRESS IN THE RANGE
19 07373 006304     ;C(O3) AND C(MXTBL)
20 07374 006301     MXTBL
21 07375 120110     SBI 2,0
22 07376 040302     STA 0,TBLSIZ
23
24 JSR @ENTRA
25 07400 030124     LDA 2,HIGBF
26 07401 041000     STA 0,0,2
27 07402 024302     LDA 1,TBLSIZ
28 07403 107000     ADD 0,1
29 07404 104110     SBI 1,1
30 07405 045001     STA 1,1,2
31 07406 145370     HLV 0
32 07407 152370     CLM 1,2
33 07410 000770     JMP *-8.
34
35 JSR @ENTRA
36 07411 006121     LDA 1,HIGBF
37 07413 106370     CLM 0,1
38 07414 000775     JMP *-3
39 07415 130010     ADI 2,2
40
41 STA 0,DSLCO
42 07417 050270     STA 2,DSLCL1
43
44 JSR @ENTRA
45 07420 006121     MOVZ 0,2
46 07421 111020     LDA 1,DSLCO
47 07422 024267     LDA 3,DSLCL1
48 07424 147570     DSPA 1,0,3
49 000000
50 07426 101002     MOV 0,0,SZC
51
52 07427 100030     ERROR
53 07430 112414     XOP 0,0,0 ;ERROR CALL
54
55 07431 100030     SUB# 0,2,SZR
56
57 07432 006117     ERROR
58 07433 100030     XOP 0,0,0 ;ERROR CALL
59
60 07434 006117     @ENTLO ;ITERATE TEST ROUTINE

```



10181 ESPCL

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
    DSBP0:  LOMBF,L09FU,XERR
    FILL    XERR
    JSR     @JFILL
    XERR
    SETUP  100.
    JSR     @ENTIN
    @ENTIN  :INITIALIZE TEST
    :ITERATION VALUE
    LDA     0,L0MBF
    :MAX. AVAILABLE AREA
    LDA     1,L09FU
    :FOR TABLE IS =
    SUB     0,1
    STA     1,MXTBL
    RNDADR  D3,MXTBL
    RAND
    JSR     @ENTRA
    @ENTRA  :C(ACO)=RANDOM #
    @RNRAD :GET ADDRESS IN THE RANGE
    D3      :C(D3) AND C(MXTBL)
    MXTBL
    SBI     2,0
    STA     0,TBLSIZ
    RAND
    JSR     @ENTRA
    @ENTRA  :C(ACO)=RANDOM #
    LDA     2,L0MBF
    STA     0,0,2
    LDA     1,TBLSIZ
    ADD     0,1
    SBI     0,1
    STA     1,1
    HLVLV  0
    CLM    1,2
    JMP     -8.
    RAND
    JSR     @ENTRA
    @ENTRA  :C(ACO)=RANDOM #
    LDA     1,L0MBF
    STA     0,1
    CLM    0,1
    JMP     -3
    ADI     2,2
    STA     2,DPB0+1
    LDA     1,-2,2
    ADD     0,2
    SUB     1,2
    ELEF    1,DPB0+3,-,1,1
    :STORE ADDR (DPB0+3)
    STA     1,0,2
    DSPA   0,0,0
    :IN DISPATCH LOCATION
    :EXECUTE DSPA, INDEX=00
    ERROR
    STA     0,0,0
    XOP    :ERROR CALL
    LDA     1,DSPER
    :RESTORE DISPATCH LOC
    STA     1,0,2
    :XERR, XOP ERROR CALL
    LOOP
    @ENTLO :ITERATE TEST ROUTINE
    JSR
  
```

10182 ESPCL

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
    DSP#1:  HIGBF,M19FU,XERR
    FILL    XERR
    JSR     @JFILL
    XERR
    SETUP  100.
    JSR     @ENTIN
    @ENTIN  :INITIALIZE TEST
    :ITERATION VALUE
    LDA     0,HIGBF
    :MAX. AVAILABLE AREA
    LDA     1,M19FU
    :FOR TABLE IS =
    SUB     0,1
    STA     1,MXTBL
    RNDADR  D3,MXTBL
    RAND
    JSR     @ENTRA
    @ENTRA  :C(ACO)=RANDOM #
    @RNRAD :GET ADDRESS IN THE RANGE
    D3      :C(D3) AND C(MXTBL)
    MXTBL
    SBI     2,0
    STA     0,TBLSIZ
    RAND
    JSR     @ENTRA
    @ENTRA  :C(ACO)=RANDOM #
    LDA     2,HIGBF
    STA     0,0,2
    LDA     1,TBLSIZ
    ADD     0,1
    SBI     0,1
    STA     1,1
    HLVLV  0
    CLM    1,2
    JMP     -8.
    RAND
    JSR     @ENTRA
    @ENTRA  :C(ACO)=RANDOM #
    LDA     1,HIGBF
    STA     0,1
    CLM    0,1
    JMP     -3
    ADI     2,2
    STA     2,DSLCO
    ELEF    1,DPB1+1
    SUB     1,2
    STA     2,DPB1+1
    ADD     0,2
    LDA     1,-2,2
    ADD     0,2
    SUB     1,2
    ELEF    1,DPB1+3,-,1,1
    :STORE ADDR (DPB1+3)
    STA     1,0,2
    MOV     0,3
    DSPA   3,0,1
    :EXECUTE DSPA, INDEX=01
    ERROR
    STA     0,0,0
    XOP    :ERROR CALL
    LDA     1,DSPER
    :RESTORE DISPATCH LOC
    STA     1,0,2
    :XERR, XOP ERROR CALL
    LOOP
  
```

0183 ESPCL  
01 07564 006117 L\NISA: JSR  
02

02NTLO ;ITERATE TEST ROUTINE

```
01 0184 ESPCL
02
03
04
05 07565 006125      LORWF,LORFU,XERR
06 07456 108030      ;FILL UPPER AND LOWER
07                      ;SCRATCH BUFFER AREA
08                      ;WITH (XERR)
09
10
11
12
13
14
15
16
17 07575 006121      ;INITIALIZE TEST
18 07576 006122      ;ITERATION VALUE
19 07577 000304      ;MAX. AVAILABLE AREA
20 07600 000301      ;FOR TABLE IS =
21 07601 120110      ;C(LORFU)=C(LORWF)
22 07602 040302      ;
23
24 07603 006121      ;TABLE SIZE =
25 07604 030132      ;1<RANDOM SIZE< C(MXTBL)-2
26 07605 041000      ;C(IACO)=RANDOM #
27 07606 020302      ;SET ADDRESS IN THE RANGE
28 07607 107000      ;C(O3) AND C(MXTBL)
29 07610 104110      ;TABLE SIZE =
30 07611 045001      ;ACTUAL TABLE BEGIN ADDR
31 07612 143370      ;IS AT C(LORWF)
32 07613 132370      ;MAKE SURE C(TABLE-2) <
33 07614 000770      ;C(TABLE-1)
34
35 07615 006121      ;C(IACO)=RANDOM #
36 07616 024132      ;GET RANDOM DATA 2 IN
37 07617 108370      ;THE RANGE OF C(TABLE-2)
38 07620 000775      ;AND C(TABLE-1)
39 07621 130010      ;ACO=DISPATCH ADDEND
40
41 07622 050267      ;STORE ADDR (DPB2+3) IN
42 07623 025376      ;DISPATCH LOCATION
43 07624 113000      ;
44 07625 132400      ;
45 07626 050270      ;
46 07627 166470      ;
47 000013
48 07631 045000      ;
49 07632 030267      ;
50 07633 145000      ;
51 07634 151222      ;AC2=(TABLE ADDR/2)+
52 07635 110010      ;BIT 15
53 07636 147370      ;ALSO STORE (TABLE ADDR/2)
54 07637 044402      ;IN DSPA INSTRUCTION
55 07640 143170      ;EXECUTE DSPA,INDEX=02
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

0185 ESPL  
01 07645 055000  
02 STA LOOP  
03 07646 006117 L\N155: JSR  
04

3.0,2  
QENTLO :ITERATE TEST ROUTINE

DSR3:  
EUSP1  
FILL  
JSR  
XERR

0186 ESPL

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

05 07647 006125 :FILL UPPER AND LOWER  
06 07650 100030 :SCRATCH BUFFER AREA  
07 :WITH (XERR)  
08 100. :INITIALIZE TEST  
09 @ENTIN :ITERATION VALUE  
10 100. :MAX. AVAILABLE AREA  
11 LDA 0, HIGBF :FOR TABLE IS =  
12 07653 020124 LDA 1, HIRFU :C(HIRFU)-C(HIGBF)  
13 07654 024133 SUB 0, 1 :  
14 07655 106400 STA 1, MXTBL :  
15 RNOADR D3, MXTBL  
16 RAND  
17 07657 006121 JSR @ENTRA :C(CAO)=RANDOM #  
18 07660 006122 JSR @RNADR :GET ADDRESS IN THE RANGE  
19 07661 000304 D3 :C(D3) AND C(MXTBL)  
20 07662 000301 MXTBL :TABLE SIZE =  
21 07663 120110 STA 2, 0 :1-RANDOM SIZE< C(MXTBL)-2  
22 07664 040302 STA 0, TBLISZ :C(CAO)=RANDOM #  
23 RAND  
24 07665 006121 JSR @ENTRA :C(CAO)=RANDOM #  
25 07666 030124 LDA 2, HIGBF :GET RANDOM DATA 2 IN  
26 07667 041000 STA 0, 0, 2 :THE RANGE OF C(TABLE-2)  
27 07670 024302 LDA 1, TBLISZ :AND C(TABLE-1)  
28 07671 107000 ADD 0, 1 :CAC=DISPATCH ADDEND  
29 07672 104110 SBI 1, 1 :AC2=TABLE BEGIN ADDR  
30 07673 045001 STA 1, 1, 2 :IS AT C(HIGBF)  
31 07674 143370 HLV 0 :MAKE SURE C(TABLE-2) <  
32 07675 132370 CLM 1, 2 :C(TABLE-1)  
33 07676 000770 JMP --5. :  
34 RAND  
35 07677 006121 JSR @ENTRA :C(CAO)=RANDOM #  
36 07700 024124 LDA 1, HIGBF :GET RANDOM DATA 2 IN  
37 07701 106370 CLM 0, 1 :THE RANGE OF C(TABLE-2)  
38 07702 000775 JMP --3 :AND C(TABLE-1)  
39 07703 130010 ADI 2, 2 :CAC=DISPATCH ADDEND  
40 :AC2=TABLE BEGIN ADDR  
41 07704 050267 STA 2, DSLC0 :STORE ADDR (DPB3+3) IN  
42 07705 025376 LDA 1, 2, 2 :DISPATCH LOCATION  
43 07706 115000 ADD 0, 2 :  
44 07707 132400 SUB 1, 2 :  
45 07710 050270 STA 2, DSLC1 :  
46 07711 166870 ELEF 1, DPB3+3, --1, 1 :  
47 000012 :  
48 07713 045000 STA 1, 0, 2 :  
49 07714 030267 LDA 2, DSLC0 :  
50 07715 152322 MOVZ 2, 3, SZC :CAC3=(TABLE ADDR/2)+  
51 07716 114010 ADI 1, 3 :HIT 15  
52 07717 153370 HLV 2 :FALSO STORE (TABLE ADDR/2)  
53 07720 050402 STA 2, DPB3+1 :TIN DSPA INSTRUCTION  
54 07721 143570 DSPA 0, 0, 3 :EXECUTE DSPA, INDEX=03  
55 000000 :  
56 ERROR  
57 07723 100030 EYR242: XOP 0, 0, 0 :ERROR CALL  
58 07724 024311 LDA 1, DSPER :RESTORE DISPATCH  
59 07725 030270 LDA 2, DSLC1 :LOCATION TO  
60 07726 045000 STA 1, 0, 2 :'YERR', XOP ERROR CALL

0187 ESPCL

01  
02 07727 006117 LAN156: JSR  
03

SEN1LO :ITERATE TEST ROUTINE

0188 ESPCL

01  
02  
03  
04  
05 07730 006125  
06 07731 100030  
07  
08  
09 07732 006120 TAN157: JSR  
10 07733 000144 :SEN1LO :ITERATE TEST  
11 07734 020124 :MAX. AVAILABLE AREA  
12 07735 024133 :FOR TABLE IS =  
13 07736 106400 :C(HIBFU)-C(HIGRF)  
14 07737 044301 :  
15  
16  
17 07740 006121 :C(ACO)=RANDOM #  
18 07741 006122 :GET ADDRESS IN THE RANGE  
19 07742 000304 :C(D3) AND C(MXTBL)  
20 07743 000501 MXTBL  
21 07744 120110 :TABLE SIZE =  
22 07745 040302 :1-RANDOM SIZE< C(MXTBL)-2  
23  
24 07746 006121 :C(ACO)=RANDOM #  
25 07747 030124 LDA 2,HIGRF  
26 07750 041000 STA 0,0,2  
27 07751 024302 LDA 1,TBLSIZ  
28 07752 107000 ADD 0,1  
29 07753 104110 SBI 1,1  
30 07754 045001 STA 1,1,2  
31 07755 143370 HLV 0  
32 07756 132570 CLM 1,2  
33 07757 000770 JMP .-8.  
34  
35 07760 006121 :C(ACO)=RANDOM #  
36 07761 024124 LDA 1,HIGRF  
37 07762 106370 CLM 0,1  
38 07763 000775 JMP -3  
39 07764 130010 ADI 2,2  
40  
41 07765 050411 STA 2,DPB4+1  
42 07766 025376 LDA 1,-2,2  
43 07767 113000 ADD 0,2  
44 07770 132400 SUB 1,2  
45 07771 166470 ELEF 1,DPB4+3,-1,1 :STORE ADDR (DPB4+3)  
46  
47 07773 045000 STA 1,0,2 :IN DISPATCH LOCATION  
48 07774 115000 MOV 0,3 :AC0EAC3  
49 07775 156170 DSPA :EXECUTE DSPA  
50  
51  
52 07777 100030 EAR245: XOP 0,0,0 :ERROR CALL  
53 10000 024311 LDA 1,DSPER :RESTORE DISPATCH LOC  
54 10001 045000 STA :XERR' XOP ERROR CALL  
55  
56 10002 006117 LAN157: JSR :SEN1LO :ITERATE TEST ROUTINE

DSPB4: EDSP1 HIGRF,HIBFU,XERR  
FILL XERR :FILL UPPER AND LOWER  
XRR :SCRATCH BUFFER AREA  
XRR :WITH (XERR)  
SETUP 100.  
:SEN1LO :INITIALIZE TEST  
:ITERATION VALUE  
0,HIGRF :MAX. AVAILABLE AREA  
1,HIBFU :FOR TABLE IS =  
0,1 :C(HIBFU)-C(HIGRF)  
STA 1,MXTBL :  
RNDADR 03,MXTBL  
RAND  
JSR @ENTRA :C(ACO)=RANDOM #  
:GET ADDRESS IN THE RANGE  
:C(D3) AND C(MXTBL)  
2,0 :TABLE SIZE =  
0,TBLSIZ :1-RANDOM SIZE< C(MXTBL)-2  
JSR @ENTRA :C(ACO)=RANDOM #  
LDA 2,HIGRF :C(TABLE-2)=RANDOM DATA 1  
STA 0,0,2 :C(TABLE-1)=(RANDOM DATA 1+  
LDA 1,TBLSIZ :TABLE SIZE-1)  
ADD 0,1 :ACTUAL TABLE BEGIN ADDR  
SBI 1,1 :IS IT C(HIGRF)  
HLV 0 :MAKE SURE C(TABLE-2) <  
CLM 1,2 :C(TABLE-1)  
JMP .-8.  
RAND  
JSR @ENTRA :C(ACO)=RANDOM #  
LDA 1,HIGRF :GET RANDOM DATA 2 IN  
CLM 0,1 :THE RANGE OF C(TABLE-2)  
JMP -3 :AND C(TABLE-1)  
ADI 2,2 :AC0=DISPATCH ADDEND  
STA 2,DPB4+1 :AC2=TABLE BEGIN ADDR  
LDA 1,-2,2 :STORE TBL ADDR IN  
ADD 0,2 :DISPA INSTRUCTION  
SUB 1,2 :  
ELEF 1,DPB4+3,-1,1 :STORE ADDR (DPB4+3)  
STA 1,0,2 :IN DISPATCH LOCATION  
MOV 0,3 :AC0EAC3  
DSPA :EXECUTE DSPA  
ERROR  
XOP 0,0,0 :ERROR CALL  
LDA 1,DSPER :RESTORE DISPATCH LOC  
STA :XERR' XOP ERROR CALL  
LOOP  
LAN157: JSR :SEN1LO :ITERATE TEST ROUTINE

```

01 DSP85: EDSP1 HIGBF,HIBFU,XERR
02
03
04 FILL XERR ;FILL UPPER AND LOWER
05 JSR @JFILL ;SCRATCH BUFFER AREA
06 XERR ;WITH (XERR)
07
08 SETUP 100.
09 JSR @ENTRA ;INITIALIZE TEST
10 JSR @ENTRA ;ITERATION VALUE
11 LDA 0,HIGBF ;MAX. AVAILABLE AREA
12 LDA 1,HIBFU ;FOR TABLE IS =
13 SUB 0,1 ;C(HIBFU)-C(HIGBF)
14 STA 1,MXTBL ;
15 RNDADR 03,MXTBL
16 RAND
17 JSR @ENTRA ;C(ACO)=RANDOM #
18 JSR @RNADR ;GET ADDRESS IN THE RANGE
19 D3 ;C(D3) AND. C(MXTBL)
20 MKTHL
21 STA 2,0 ;TABLE SIZE =
22 RAND 0,TBLSIZ ;1<RANDOM SIZE< C(MXTBL)-2
23
24 JSR @ENTRA ;C(ACO)=RANDOM #
25 LDA 2,HIGBF
26 STA 0,0,2 ;C(TABLE-2)= RANDOM DATA 1
27 LDA 1,TBLSIZ ;C(TABLE-1)=(RANDOM DATA 1+
28 ADD 0,1 ;TABLE SIZE-1)
29 SBI 1,1 ;ACTUAL TABLE BEGIN ADDR
30 STA 1,1,2 ;IS AT C(HIGBF)
31 HLV 0 ;MAKE SURE C(TABLE-2) <
32 CLM 1,2 ;C(TABLE-1)
33 JMP *-8.
34 RAND
35 JSR @ENTRA ;C(ACO)=RANDOM #
36 LDA 1,HIGBF ;GET RANDOM DATA 2 IN
37 CLM 0,1 ;THE RANGE OF C(TABLE-2)
38 JMP *-3 ;AND C(TABLE-1)
39 ADI 2,2 ;ACO=DISPATCH ADDEND
40 STA 2,DSLCO ;AC2=TABLE BEGIN ADDR
41 ELEF 1,DP85=-,1 ;SAVE AC2
42 ;STORE (TBL ADDR-PC
43
44 SUB 1,2 ;OF DSPA+1) IN DSPA
45 STA 2,DP85+1 ;INSTRUCTION
46 LDA 2,DSLCO ;
47 LDA 1,-2,2 ;
48 ADD 0,2 ;
49 SUB 1,2 ;
50 ELEF 1,DP85+3=-,1,1 ;STORE ADDR (DP85+3)
51
52 STA 1,0,2 ;IN DISPATCH LOCATION
53 DSPA 0,0,1 ;EXECUTE DSPA
54
55 ERROR
56 10056 100030 E\R244: XOP ;ERROR CALL
57 10057 024311 LDA ;RESTORE DISPATCH LOC
58 10060 045000 STA ;'XERR' XOP ERROR CALL
59 LOOP
60 10061 006117 L\W160: JSR @ENTLO ;ITERATE TEST ROUTINE

```

```

:0191 ESPCL
01 10142 030270 LDA 2,0SLC1 ;'XERR' XOP ERROR CALL
02 10143 041000 STA 0,0,2
03
04 10144 006117 L\N161: JSR @ENTLO ;ITERATE TEST ROUTINE
05

DSP86: EDSPI HIGRF,HIBFU,XERR
FILL XERR ;FILL UPPER AND LOWER
JSR @JFILL ;SCRATCH BUFFER AREA
XERR ;WITH (XERR)

SETUP 100.
JSR @ENTIN ;INITIALIZE TEST
;ITERATION VALUE
LDA 0,HIGBF ;MAX. AVAILABLE AREA
LDA 1,HIBFU ;FOR TABLE IS =
SUB 0,1 ;C(HIGFU)-C(HIGBF)
STA 1,MXTBL ;
RNDADR 03,MXTBL
RAND
JSR @ENTRA ;C(ACO)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(D3) AND C(MXTBL)
D3
MXTBL
SBI 2,0 ;TABLE SIZE =
STA 0,TBLSIZ ;1<RANDOM SIZE< C(MXTBL)-2
RAND
JSR @ENTRA ;C(ACO)=RANDOM #
LDA 2,HIGBF ;C(TABLE-2)= RANDOM DATA 1
STA 0,0,2 ;C(TABLE-1)= (RANDOM DATA 1)
LDA 1,TBLSIZ ;TABLE SIZE-1)
ADD 0,1 ;ACTUAL TABLE BEGIN ADDR
SBI 1,1 ;IS AT C(HIGBF)
STA 1,1,2 ;MAKE SURE C(TABLE-2) <
HLV 0 ;C(TABLE-1)
CLM 1,2
JMP --8.
RAND
JSR @ENTRA ;C(ACO)=RANDOM #
LDA 1,HIGBF ;GET RANDOM DATA 2 IN
CLM 0,1 ;THE RANGE OF C(TABLE-2)
JMP --3 ;AND C(TABLE-1)
ADI 2,2 ;ACO=DISPATCH ADDEND
STA 2,0SLC0 ;AC2=TABLE BEGIN ADDR
LDA 1,-2,2 ;STORE ADDR (DPB6+3)
ADD 0,2 ;IN DISPATCH LOCATION
SUB 1,2
STA 2,0SLC1 ;
ELEM 1,DPB6+3,-1,1 ;
STA 1,0,2 ;
LDA 2,DSLCO ;
MOV 2,1 ;AC2=(TBL ADDR/2)+
MOVZR 2,2,-SZC ;BIT 15
ADI 1,2 ;ALSO STORE (TBL ADDR/2)
HLV 1 ;IN DSPA INSTRUCTION
STA 1,DPB6+1 ;ACO=AC3
MOV 0,1 ;EXECUTE DSPA INSTRUCTION
DP86: DSPA 3,0,2
ERROR
58 10140 100030 ENR245: XOP ;ERROR CALL
59 10141 020311 LDA 0,0,DSPER ;RESTORE DISPATCH LOC.
60

```

0194 ESPCL  
 01 10225 045000 STA LOOP  
 02  
 03 10226 006117 L\N162: JSR JENTLO ;ITERATE TEST ROUTINE  
 04

```

10193 ESPCL
01 DSPB7: EDSP1 LOWBF,LOBFU,XERR
02
03
04 FILL XERR ;FILL UPPER AND LOWER
05 JSR @JFILL ;SCRATCH BUFFER AREA
06 XERR ;WITH (XERR)
07
08 SETUP 100. ;INITIALIZE TEST
09 @ENTIN @ENTIN ;ITERATION VALUE
10 10150 000144 100. ;MAX. AVAILABLE AREA
11 LDA 0,LOBMF ;FOR TABLE IS =
12 10151 020132 LDA 1,LOBFU ;C(LOBFU)-C(LOBMF)
13 10152 024123 SUB 0,1
14 10153 106400 STA 1,MXTRL
15 RNDADR D3,MXIBL
16 RAND
17 JSR @ENTRA ;C(ACO)=RANDOM #
18 10155 006121 JSR @RNADR ;GET ADDRESS IN THE RANGE
19 10156 006122 D3 ;C(D3) AND C(MXIBL)
20 10160 000301 MXIBL
21 10161 120110 SBI 2,0
22 10162 040302 STA 0,TRLSIZ
23 RAND
24 10163 006121 JSR @ENTRA ;C(ACO)=RANDOM #
25 10164 030132 LDA 2,LOBMF
26 10165 041000 STA 0,0,2
27 10166 024302 LDA 1,TBLSIZ
28 10167 107000 ADD 0,1
29 10170 104110 SBI 1,1
30 10171 045001 STA 1,1,2
31 10172 143370 HLV 0
32 10173 132370 CLM 1,2
33 10174 000770 JMP *-8.
34 RAND
35 10175 006121 JSR @ENTRA ;C(ACO)=RANDOM #
36 10176 024132 LDA 1,LOBMF
37 10177 106370 CLM 0,1
38 10200 000775 JMP *-3
39 10201 130010 ADI 2,2
40
41 10202 050267 STA 2,DSLCO
42 10203 025376 LDA 1,-2,2
43 10204 113000 ADD 0,2
44 10205 132400 SUB 1,2
45 10206 050270 STA 2,DSLCL
46 10207 166870 ELEF 1,OPB7+3--1,1 ;
47 10211 000013 STA 1,0,2
48 10212 030267 LDA 2,DSLCO
49 10213 155222 MOVZR 2,3,SZC
50 10214 114010 ADI 1,3
51 10215 153370 HLV 2
52 10216 050403 STA 2,OPB7+1
53 10217 111000 MOV 0,2
54 10220 153570 DSPA 2,0,3
55 000000 ;EXECUTE DSPA
56 ERROR
57 10222 100030 E\R246: XOP 0,0,0 ;ERROR CALL
58 10223 024311 LDA 1,DSPEN ;RESTORE DISPATCH LOC
59 10224 030270 LDM 2,DSLCL ;XERR XOP ERROR CALL
60

```

```

10195 ESPCL
01 000000 DSPC0:
02 000000 DSPC0:
03 000000 DSPC0:
04 000000 DSPC0:
05 000000 DSPC0:
06 000000 DSPC0:
07 000000 DSPC0:
08 000000 DSPC0:
09 000000 DSPC0:
10 000000 DSPC0:
11 000000 DSPC0:
12 000000 DSPC0:
13 000000 DSPC0:
14 000000 DSPC0:
15 000000 DSPC0:
16 000000 DSPC0:
17 000000 DSPC0:
18 000000 DSPC0:
19 000000 DSPC0:
20 000000 DSPC0:
21 000000 DSPC0:
22 000000 DSPC0:
23 000000 DSPC0:
24 000000 DSPC0:
25 000000 DSPC0:
26 000000 DSPC0:
27 000000 DSPC0:
28 000000 DSPC0:
29 000000 DSPC0:
30 000000 DSPC0:
31 000000 DSPC0:
32 000000 DSPC0:
33 000000 DSPC0:
34 000000 DSPC0:
35 000000 DSPC0:
36 000000 DSPC0:
37 000000 DSPC0:
38 000000 DSPC0:
39 000000 DSPC0:
40 000000 DSPC0:
41 000000 DSPC0:
42 000000 DSPC0:
43 000000 DSPC0:
44 000000 DSPC0:
45 000000 DSPC0:
46 000000 DSPC0:
47 000000 DSPC0:
48 000000 DSPC0:
49 000000 DSPC0:
50 000000 DSPC0:
51 000000 DSPC0:
52 000000 DSPC0:
53 000000 DSPC0:
54 000000 DSPC0:
55 000000 DSPC0:
56 000000 DSPC0:
57 000000 DSPC0:
58 000000 DSPC0:
59 000000 DSPC0:
60 000000 DSPC0:

:GET ADDRESS IN THE RANGE
:C(HOSTB) AND C(HIRFU)
:SAVE RNDM ADDR 1
:MAKE SURE IT'S NOT
:EQUAL TO DISPATCH
:ADDR
:STORE TABLE ADDR IN
:RNDM ADDR 1
:
:AC0=AC2=RNDM ADDR
:ADD INDIRCT BIT TO AC0
:STORE THIS IN DSPA
:AC1=DISPATCH ADDRND
:EXECUTE DSPA,INDEX#00
:
:ERROR CALL
:OPPER:RESTORE THE RNDM ADDRESSES
:AND DISPATCH LOC TO
:;XERR' XOP ERROR CALL
:
:ENTLO ;ITERATE TEST ROUTINE

```

```

01 10301 006122 JSR
02 10302 000275 HDSTB
03 10303 000133 HIRFU
04 10304 040272 LDA
05 10305 024271 LDA
06 10306 106415 SUB#
07 10307 000771 JMP
08 10310 030270 LDA
09 10311 140710 XCH
10 10312 041000 STA
11 10313 141000 MOV
12 10314 103290 ADDUR
13 10315 040403 STA
14 10316 024267 LDA
15 10317 146170 OPC0:
16 100000
17
18 10321 100030 ENR247: XOP
19 10322 020311 LDA
20 10323 042271 STA
21 10324 042272 STA
22
23 10325 006117 LNI163: JSR
24
25

```

```

0196 ESPCL
01 10301 006122 JSR
02 10302 000275 HDSTB
03 10303 000133 HIRFU
04 10304 040272 LDA
05 10305 024271 LDA
06 10306 106415 SUB#
07 10307 000771 JMP
08 10310 030270 LDA
09 10311 140710 XCH
10 10312 041000 STA
11 10313 141000 MOV
12 10314 103290 ADDUR
13 10315 040403 STA
14 10316 024267 LDA
15 10317 146170 OPC0:
16 100000
17
18 10321 100030 ENR247: XOP
19 10322 020311 LDA
20 10323 042271 STA
21 10324 042272 STA
22
23 10325 006117 LNI163: JSR
24
25

```





```

10199 ESPCL
01 000002 DSPC2: DSSW0=2
02 EDSP3 LOWBF,LOBFU,XERR,DPC2,HITBAD,2
03 DSSW1=0
04 EDSP1 LOWBF,LOBFU,XERR
05
06 FILL XERR
07 JSR @JFILL
08 XERR
09
10 SETUP
11 JSR @ENTRA
12 JSR @ENTRA
13 JSR @ENTRA
14 JSR @ENTRA
15 JSR @ENTRA
16 JSR @ENTRA
17 JSR @ENTRA
18 JSR @ENTRA
19 JSR @ENTRA
20 JSR @ENTRA
21 JSR @ENTRA
22 JSR @ENTRA
23 JSR @ENTRA
24 JSR @ENTRA
25 JSR @ENTRA
26 JSR @ENTRA
27 JSR @ENTRA
28 JSR @ENTRA
29 JSR @ENTRA
30 JSR @ENTRA
31 JSR @ENTRA
32 JSR @ENTRA
33 JSR @ENTRA
34 JSR @ENTRA
35 JSR @ENTRA
36 JSR @ENTRA
37 JSR @ENTRA
38 JSR @ENTRA
39 JSR @ENTRA
40 JSR @ENTRA
41 JSR @ENTRA
42 JSR @ENTRA
43 JSR @ENTRA
44 JSR @ENTRA
45 JSR @ENTRA
46 JSR @ENTRA
47 JSR @ENTRA
48 JSR @ENTRA
49 JSR @ENTRA
50 JSR @ENTRA
51 JSR @ENTRA
52 JSR @ENTRA
53 JSR @ENTRA
54 JSR @ENTRA
55 JSR @ENTRA
56 JSR @ENTRA
57 JSR @ENTRA
58 JSR @ENTRA
59 JSR @ENTRA
60 JSR @ENTRA

```

```

01 10503 000275
02 10504 000133
03 10505 040272
04 10506 024271
05 10507 106415
06 10510 000771
07 10511 030270
08 10512 140710
09 10513 041000
10 10514 141000
11 10515 111222
12 10516 110910
13 10517 143370
14 10520 103240
15 10521 040403
16 10522 024267
17 10523 147170 DPC2:
18 100000
19
20 10525 100030 ENR251: XOP
21 10526 020311
22 10527 042271
23 10528 042272
24
25 10531 006117 L/W165: JSR
26
27

```

```

;C(HDSTB) AND C(HIRFU)
;SAVE RNDM ADDR 1
;MAKE SURE IT'S NOT
;EQUAL TO DISPATCH
;ADDR
;STORE TABLE ADDR IN
;RNDM ADDR 1
;
;AC0=AC2=RANDOM ADDR
;BIT 15
;AC0=(RNDM ADDR/2)
;ADD INDIRECT HIT TO AC0
;STORE THIS IN DSPA
;AC1=DISPATCH ADDEND
;EXECUTE DSPA,INDEX=02

```

```

;C(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(03) AND C(MXTBL)
;TABLE SIZE =
;1-RANDOM SIZE< C(MXTBL)-2
;C(AC0)=RANDOM #
;C(TABLE-2)=RANDOM DATA 1
;TABLE SIZE=1
;ACTUAL TABLE BEGIN ADDR
;IS AT C(LOWBF)
;MAKE SURE C(TABLE-2) <
;C(TABLE-1)
;C(AC0)=RANDOM #
;GET RANDOM DATA 2 IN
;THE RANGE OF C(TABLE-2)
;AND C(TABLE-1)
;AC0=DISPATCH ADDEND
;AC2=TABLE BEGIN ADDR
;SAVE AC0
;SAVE AC1
;
;
;
;SAVE DISPATCH ADDR
;AND STORE (DPC2+3) IN
;DISPATCH ADDR LOCATION
;MAKE SURE THAT RNDM
;ADDR IS NOT EQ. TO
;(TABLE-1) OR (TABLE-2)
;C(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE

```

```

10201 ESHPL
01 000003 DSPC3: DSSM0=3
02 DSP3 HIGBF,HIGFU,XERR,DPC3,LOTBAD,3
03 DSSM1=0
04 EDSPI HIGBF,HIGFU,XERR
05
06
07 10532 066125 XERR
08 10533 100030 JPFILL
09
10 SETUP 100,
11 10534 066120 T\N166: JSR @ENTIN ;INITIALIZE TEST
12 10535 000144 ;ITERATION VALUE
13 10536 020124 LDA 0,HIGBF
14 10537 020135 LDA 1,HIGFU
15 10540 106400 SUB 0,1
16 10541 040301 STA 1,MXTBL
17 RNDADR D3,MXTBL
18 RAND
19 10542 066121 JSR @ENTRA
20 10543 066122 JSR @RNADR
21 10544 000304 D3
22 10545 000301 MXTBL
23 10546 120110 SBI 2,0
24 10547 040302 STA 0,TBLSIZ
25 RAND
26 10550 066121 JSR @ENTRA
27 10551 030124 LDA 2,HIGBF
28 10552 041000 STA 0,0,2
29 10553 024302 LDA 0,TBLSIZ
30 10554 107000 ADD 0,1
31 10555 109110 SBI 1,1
32 10556 045001 STA 1,1,2
33 10557 143370 HLW 0
34 10560 132370 CLW 1,2
35 10561 000770 JMP *-8,
36 RAND
37 10562 066121 JSR @ENTRA
38 10563 024124 LDA 1,HIGBF
39 10564 106370 CLM 0,1
40 10565 000775 JMP *-3
41 10566 130010 ADI 2,2
42
43 EOSP2 DPC3,LOTBAD,3
44 10567 040267 STA 0,OSLCO
45 10570 050270 STA 2,OSLCO
46 10571 025376 LDA 1,-2,2
47 10572 113000 ADD 0,2
48 10573 132400 SUB 1,2
49 10574 050271 STA 2,DSSV0
50 10575 166470 ELEF 1,DPC3+3,-1,1
51 10576 000032 STA 1,0,2
52 10577 045000 LOTBAD
53
54 10600 020132 LDA 0,LOWBF
55 10601 120010 ADI 2,0
56 10602 040274 STA 0,LDSTB
57 RNDADR LDSTB,LOBFU
58 RAND
59 10603 066121 JSR @ENTRA
60 10604 066122 JSR @RNADR

```

```

0202 ESPCL
01 10605 000274 LDSTB
02 10606 000123 LOBFU
03 10607 040272 STA
04 10610 024271 LDA
05 10611 106415 SUB#
06 10612 000771 JMP
07 10613 030270 LDA
08 10614 140710 XCH
09 10615 041000 STA
10 10616 141000 MOV
11 10617 115222 MOVZR
12 10620 114010 ADI
13 10621 143370 HLW
14 10622 103200 ADDOK
15 10623 040403 STA
16 10624 024267 LDA
17 10625 147570 DPC3:
18 100000
19 ERROR
20 10627 100030 E\N252: XOP
21 10630 020311 LDA
22 10631 042271 STA
23 10632 042272 STA
24 LOOP
25 10633 006117 L\N166: JSR
26
27

```

```

;C(LDSTR) AND C(LORFU)
;SAVE RNDM ADDR 1
;MAKE SURE IT'S NOT
;EQUAL TO DISPATCH
;ADDR
;STORE TABLE ADDR IN
;RNDM ADDR 1
;
;AC0=AC2=RANDOM ADDR
;BIT 15
;AC0=(RNDM ADDR/2)
;ADD INDIRECT BIT TO AC0
;STORE THIS IN DSPA
;AC1=DISPATCH ADDEND IN DSPA
;EXECUTE DSPA,INDEX=03

```

```

;C(LDSTR) AND C(LORFU)
;SAVE RNDM ADDR 1
;MAKE SURE IT'S NOT
;EQUAL TO DISPATCH
;ADDR
;STORE TABLE ADDR IN
;RNDM ADDR 1
;
;AC0=AC2=RANDOM ADDR
;BIT 15
;AC0=(RNDM ADDR/2)
;ADD INDIRECT BIT TO AC0
;STORE THIS IN DSPA
;AC1=DISPATCH ADDEND IN DSPA
;EXECUTE DSPA,INDEX=03

```

```

;C(LDSTR) AND C(LORFU)
;SAVE RNDM ADDR 1
;MAKE SURE IT'S NOT
;EQUAL TO DISPATCH
;ADDR
;STORE TABLE ADDR IN
;RNDM ADDR 1
;
;AC0=AC2=RANDOM ADDR
;BIT 15
;AC0=(RNDM ADDR/2)
;ADD INDIRECT BIT TO AC0
;STORE THIS IN DSPA
;AC1=DISPATCH ADDEND IN DSPA
;EXECUTE DSPA,INDEX=03

```

```

;C(LDSTR) AND C(LORFU)
;SAVE RNDM ADDR 1
;MAKE SURE IT'S NOT
;EQUAL TO DISPATCH
;ADDR
;STORE TABLE ADDR IN
;RNDM ADDR 1
;
;AC0=AC2=RANDOM ADDR
;BIT 15
;AC0=(RNDM ADDR/2)
;ADD INDIRECT BIT TO AC0
;STORE THIS IN DSPA
;AC1=DISPATCH ADDEND IN DSPA
;EXECUTE DSPA,INDEX=03

```

```

;C(LDSTR) AND C(LORFU)
;SAVE RNDM ADDR 1
;MAKE SURE IT'S NOT
;EQUAL TO DISPATCH
;ADDR
;STORE TABLE ADDR IN
;RNDM ADDR 1
;
;AC0=AC2=RANDOM ADDR
;BIT 15
;AC0=(RNDM ADDR/2)
;ADD INDIRECT BIT TO AC0
;STORE THIS IN DSPA
;AC1=DISPATCH ADDEND IN DSPA
;EXECUTE DSPA,INDEX=03

```

```

10203 ESPCL
01 10634 006125 JSR @ENTRA
02 10635 100030 XERR
03 000000 DSPD0: DSSWU=0
04 000001 DSSP4: HIGBF,HIBFU,XERR,DP00,LOTBAD,0,HITBAD
05 DSSW1=1
06 EDSP1 HIGBF,HIBFU,XERR
07 10634 006125 JSR @ENTRA
08 10635 100030 XERR
09 10634 006125 JSR @ENTRA
10 10635 100030 XERR
11 10636 006120 T\N167: SETUP
12 10637 000144 100.
13 10640 020124 LDA
14 10641 024135 LDA
15 10642 106400 SUB
16 10643 044501 STA
17 10644 006121 RAND
18 10645 006122 JSR @ENTRA
19 10646 000304 D3
20 10647 000301 MXTBL
21 10648 120110 SBI
22 10649 040302 STA
23 10650 006121 RAND
24 10651 030124 LDA
25 10652 041000 STA
26 10653 024302 LDA
27 10654 107000 ADD
28 10655 104110 SBI
29 10656 045001 STA
30 10657 143370 HLV
31 10658 132370 CLM
32 10659 000770 JMP
33 10660 006121 RAND
34 10661 024124 LDA
35 10662 106370 CLM
36 10663 000775 JMP
37 10664 130010 AUI
38 10665 040267 EDSP2
39 10666 050270 STA
40 10667 025376 LDA
41 10668 113000 ADD
42 10669 132000 SUB
43 10670 050271 STA
44 10671 000050 ELEF
45 10672 045000 STA
46 10673 020132 LOTBAD
47 10674 120010 LDA
48 10675 040274 ADI
49 10676 106400 STA
50 10677 006121 RAND
51 10678 006121 JSR @ENTRA
52 10679 006121 JSR @ENTRA
53 10680 006121 JSR @ENTRA
54 10681 006121 JSR @ENTRA
55 10682 006121 JSR @ENTRA
56 10683 006121 JSR @ENTRA
57 10684 006121 JSR @ENTRA
58 10685 006121 JSR @ENTRA
59 10686 006121 JSR @ENTRA
60 10687 006121 JSR @ENTRA

```

```

0204 ESPCL
01 10706 006122 JSR
02 10707 000274 LOTB
03 10710 000123 LOBFU
04 10711 040271 STA
05 10712 024271 LDA
06 10713 106415 SUB#
07 10714 000771 JMP
08 10715 030270 LOA
09 10716 140710 XCH
10 10717 041000 STA
11 10720 120012 LOA
12 10721 020124 STA
13 10722 040275 ADI
14 10723 006121 JSR @ENTRA
15 10724 006122 JSR @ENTRA
16 10725 000275 HDSTB
17 10726 000133 HIBFU
18 10727 040273 STA
19 10730 024271 LDA
20 10731 106415 SUR#
21 10732 000771 JMP
22 10733 030272 LDA
23 10734 112415 SUB#
24 10735 000765 JMP
25 10736 110710 XCH
26 10737 103240 ADDOR
27 10740 041000 STA
28 10741 141000 MOV
29 10742 103240 ADDOR
30 10743 040403 STA
31 10744 024267 DP00:
32 10745 146170 DSPA
33 10746 100000 DP00:
34 10747 100030 ERROR
35 10748 020311 XOP
36 10750 020311 LDA
37 10751 042271 STA
38 10752 042272 STA
39 10753 042273 STA
40 10754 006117 L\N167: JSR
41 10755 006117 L\N167: JSR
42 10756 006117 L\N167: JSR
43 10757 006117 L\N167: JSR

```

```

;GET ADDRESS IN THE RANGE
;C(LOSTB) AND C(LOBFU)
;SAVE RNDM ADDR 1
;MAKE SURE IT'S NOT
;EQUAL TO DISPATCH
;ADDR
;STORE TABLE ADDR IN
;RNDM ADDR 1
;
;MAKE SURE THAT RNDM
;ADDR IS NOT EQ. TO
;(TABLE-1) OR (TABLE-2)
;C(ACO)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(HOSTB) AND C(HIBFU)
;
;SAVE RANDOM ADDR 2
;MAKE SURE THAT
;RANDOM ADDR 2 IS NOT
;EQ. TO RNDM ADDR 1 OR
;DISPATCH ADDR
;
;STORE (RANDOM ADDRESS 1)
;IN RANDOM ADDR 2
;
;AC0=FAC2=RANDOM ADDR
;ADD INDIRECT BIT TO AC0
;STORE THIS IN DSPA
;AC1=DISPATCH ADDR
;EXECUTE DSPA.INDEX=00
;
;ERROR CALL
;RESTORE THE RNDM ADDRESSES
;AND DISPATCH LOC TO
;XERR. XOP ERROR CALL
;
@ENTLO ;ITERATE TEST ROUTINE

```

```

10205 ESPCL      000001 0SP01:
01              DSSW0=1  HIGBF,HIBFU,XERR,DPD1,HITRAD,1,HITBAD
02              EDSP4=1  HIGBF,HIBFU,XERR,DPD1,HITRAD,1,HITBAD
03              EDSP1=1  HIGBF,HIBFU,XERR
04              EDSP1  HIGBF,HIBFU,XERR
05              FILL XERR
06              XERR 0,FILL
07 10755 006125 JSR @ENTRA
08 10756 100030 XERR
09
10          SETUP 100.
11 10757 006120 T\N170:
12 10760 000144 JSR @ENTIN
13 10761 020124 LDA 0,HIGBF
14 10762 024133 LDA 1,HIBFU
15 10763 106400 SUB 0,1
16 10764 044301 STA 1,MXTBL
17          RNDADR D3,MXTBL
18          RAND
19 10765 006121 JSR @ENTRA
20 10766 006122 JSR @RNADR
21 10767 000304 D3
22 10770 000301 MXTBL
23 10771 120110 STA 2,0
24 10772 040302 STA 0,TBLSIZ
25          RAND
26 10773 006121 JSR @ENTRA
27 10774 030124 LDA 2,HIGBF
28 10775 041000 STA 0,0,2
29 10776 024302 LDA 1,TBLSIZ
30 10777 107000 ADD 0,1
31 11000 104110 SBI 1,1
32 11001 045001 STI 1,1,2
33 11002 143370 HLV 0
34 11003 132370 CLM 1,2
35 11004 000770 JMP
36          RAND
37 11005 006121 JSR @ENTRA
38 11006 024124 LDA 1,HIGBF
39 11007 106370 CLM 0,1
40 11010 000775 JMP 2,3
41 11011 150010 ADI 2,2
42
43          EDSP2  DPD1,HITBAD,1,HITBAD
44 11012 040267 STA 0,DSLCO
45 11013 050270 STA 2,DSLCO
46 11014 025376 LDA 1,-2,2
47 11015 113000 ADD 0,2
48 11016 132400 SUB 1,2
49 11017 050271 STA 2,0SSV0
50 11020 166470 ELEM 1,DPD1+3--1,1
51 11021 000053 STA 1,0,2
52 11022 045000 HITBAD
53
54 11023 020124 LDA 0,HIGBF
55 11024 120010 ADI 2,0
56 11025 040275 RNDADR 0,HOSTR
57          RNDADR  HOSTB,HIBFU
58          RAND
59 11026 006121 JSR @ENTRA
60 11027 006122 JSR @RNADR

```

```

0206 ESPCL
01 11030 000275 HOSTB
02 11031 000135 HIBFU
03 11032 040272 STA
04 11033 024271 LDA
05 11034 106415 SUB#
06 11035 000771 JMP
07 11036 030270 LDA
08 11037 140710 XCH 2,0
09 11040 041000 STA 0,0,2
10 11041 020124 LDA 0,HIGRF
11 11042 120010 ADI 2,0
12 11043 040275 STA @ENTRA
13 11044 006121 JSR @ENTRA
14 11045 006122 JSR @RNADR
15 11046 000275 HOSTB
16 11047 000135 HIBFU
17 11050 040273 STA
18 11051 024271 LDA
19 11052 106415 SUB#
20 11053 000771 JMP
21 11054 030272 LDA
22 11055 112415 SUB#
23 11056 000766 JMP
24 11057 110710 XCH 0,2
25 11060 103240 ADDOR
26 11061 041000 STA 0,0,2
27 11062 141000 MOV 2,0
28 11063 166470 ELEM 1,DPD1--1,1
29 11065 122400 SUB 1,0
30 11066 103240 ADDOR
31 11067 040403 STA 0,DPD1+1
32 11070 024272 LDA 1,DSLCO
33 11071 146370 DPD1: DSPA 1,0,1
34          ERROR
35 11073 100030 E\N254: XDP
36 11074 020311 LDA
37 11075 042271 STA
38 11076 042272 STA
39 11077 042273 STA
40
41          @ENTLO ;ITERATE TEST ROUTINE
42 11100 006117 L\N170: JSR
43
44

```

```

;C(HOSTB) AND C(HIBFU)
;SAVE RNDM ADDR 1
;MAKE SURE IT'S NOT
;EQUAL TO DISPATCH
;ADDR
;STORE TABLE ADDR IN
;RNDM ADDR 1
;
;MAKE SURE THAT RNDM
;ADDR IS NOT EQ. TO
;(TABLE-1) OR (TABLE-2)
;C(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(HOSTB) AND C(HIBFU)
;
;SAVE RANDOM ADDR 2
;MAKE SURE THAT
;RANDOM ADDR 2 IS NOT
;EQ. TO RNDM ADDR 1 OR
;DISPATCH ADDR
;
;STORE (RANDOM ADDRESS 1)
;IN RANDOM ADDR 2
;
;AC0=AC2=RANDOM ADDR
;AC0=(RNDM ADDR-PC OF
;DSP4+1)
;ADD INDIRECT BIT TO AC0
;STORE THIS IN DSP4
;AC1=DISPATCH ADDR#
;EXECUTE DSP4,INDEX=01
;
;ERROR CALL
;RESTORE THE RNDM ADDRESSES
;AND DISPATCH LOC TO
;XERR' XOP ERROR CALL
;
@ENTLO ;ITERATE TEST ROUTINE

```

```

;C(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(D3) AND C(MXTBL)
;TABLE SIZE =
;1<RANDOM SIZE< C(MXTBL)-2
;C(AC0)=RANDOM #
;C(TABLE-2)=RANDOM DATA 1+
;C(TABLE-1)=RANDOM DATA 1+
;TABLE SIZE-1)
;ACTUAL TABLE BEGIN ADDR
;IS AT C(HIGBF)
;MAKE SURE C(TABLE-2) <
;C(TABLE-1)
;
;C(AC0)=RANDOM #
;GET RANDOM DATA 2 IN
;THE RANGE OF C(TABLE-2)
;AND C(TABLE-1)
;AC0=DISPATCH ADDR#
;AC2=TABLE BEGIN ADDR
;SAVE AC0
;SAVE AC1
;
;SAVE DISPATCH ADDR
;AND STORE (DPD1+3) IN
;DISPATCH ADDR LOCATION
;MAKE SURE THAT RNDM
;ADDR IS NOT EQ. TO
;(TABLE-1) OR (TABLE-2)
;C(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE

```

```

10207 ESPCL      0000002 DSP02:
01 11154 000275  DSSW02
02 11155 000133  LONBF,LOBFU,XERR,DP02,HITBAD,2,LOTBAD
03 11156 040272  DSSW1=1
04 11157 024271  EDSP1  LOMRF,LOBFU,XERR
05 11160 108415  FILL XERR
06 11161 008771  @JFILL
07 11101 006125  JSR @ENTRA
08 11102 100030  XERR
09 11103 006120  T\W171:
10 11104 000144  SETUP
11 11105 020132  LDA
12 11106 024123  LDA
13 11107 106400  SUB
14 11110 044301  RNDADR 03,MXTHL
15 11111 006121  JSR @ENTRA
16 11112 006122  JSR @RNDADR
17 11113 000504  DS
18 11114 000501  SBI
19 11115 120110  STA
20 11116 040302  RAND
21 11117 006121  JSR @ENTRA
22 11120 030132  LDA
23 11121 041000  STA
24 11122 024302  LDA
25 11123 107000  ADD
26 11124 104110  SBI
27 11125 045001  STA
28 11126 143370  HLVD
29 11127 132370  CLM
30 11130 000770  JMP *-8.
31 11131 006121  JSR @ENTRA
32 11132 024132  LDA
33 11133 106370  CLM
34 11134 000775  JMP *-3
35 11135 130010  ADI
36 11136 040267  EDSP2  DP02,HITBAD,2,LOTBAD
37 11137 050270  STA
38 11140 025376  LOA
39 11141 113000  ADD
40 11142 132400  SUB
41 11143 050271  STA
42 11144 166470  ELEF
43 11146 045000  STA
44 11147 020124  HITBAD
45 11150 120010  LDA
46 11151 040275  ADI
47 11152 006121  STA
48 11153 006122  JSR @RNDADR
49 11152 006121  JSR @ENTRA
50 11153 006122  JSR @RNDADR
10208 ESPCL
01 11154 000275  HOSTB
02 11155 000133  HIBFU
03 11156 040272  STA
04 11157 024271  LDA
05 11160 108415  SUB#
06 11161 008771  JMP
07 11162 030270  XCH
08 11163 140710  STA
09 11164 041000  LDA
10 11165 020132  LDA
11 11166 120010  ADI
12 11167 040274  JSR @ENTRA
13 11170 006121  JSR @ENTRA
14 11171 006122  JSR @ENTRA
15 11172 000274  LDSTB
16 11173 000123  LDSTB
17 11174 040273  STA
18 11175 024271  LDA
19 11176 106415  SUB#
20 11177 000771  JMP
21 11200 030272  LDA
22 11201 112415  SUB#
23 11202 000766  JMP
24 11203 110710  XCH
25 11204 103240  ADDR
26 11205 041000  STA
27 11206 141000  MOVZ
28 11207 111222  ADI
29 11210 110010  HLVD
30 11211 143370  ADDUR
31 11212 103240  STA
32 11213 040403  LDA
33 11214 024267  DP02:
34 11215 147170  DSPA
35 100000  ERROR
36 11217 100030  E\R255:
37 11220 020311  LDA
38 11221 042271  STA
39 11222 042272  STA
40 11223 042273  STA
41 11224 006117  L\W171:
42 11224 006117  JSR
43 11224 006117  L\W171:
44
45

```

```

;C(HOSTB) AND C(HIBFU)
;SAVE RNDM ADDR 1
;MAKE SURE IT'S NOT
;EQUAL TO DISPATCH
;ADDR
;STORE TABLE ADDR IN
;RNDM ADDR 1
;
;MAKE SURE THAT RNDM
;ADDR IS NOT EQ. TO
;(TABLE-1) OR (TABLE-2)
;C(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(LOBFU) AND C(LOBFU)
;
;SAVE RANDOM ADDR 2
;MAKE SURE THAT
;RANDOM ADDR 2 IS NOT
;EQ. TO RNDM ADDR 1 OR
;DISPATCH ADDR
;
;STORE @RANDOM ADDRESS 1
;IN RANDOM ADDR 2
;
;AC0=AC2=RANDOM ADDR
;MOVZ
;BIT 15
;AC0=(RNDM ADDR/2)
;ADD INDIRECT BIT TO AC0
;STORE THIS IN DSPA
;AC1=DISPATCH ADDR
;EXECUTE DSPA,INDEX=02
;
;ERROR CALL
;RESTORE THE RNDM ADDRESSES
;AND DISPATCH LOC TO
;XERR; KOP ERROR CALL
;
;ENTLO ;ITERATE TEST ROUTINE

```

```

;FILL UPPER AND LOWER
;SCRATCH BUFFER AREA
;WITH (XERR)
;
;INITIALIZE TEST
;ITERATION VALUE
;MAX. AVAILABLE AREA
;FOR TABLE IS =
;C(LOBFU)-C(LOBMF)
;
;C(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(O3) AND C(MXTBL)
;TABLE SIZE =
;1<RANDOM SIZE< C(MXTBL)-2
;
;C(AC0)=RANDOM #
;C(TABLE-2)= RANDOM DATA 1+
;C(TABLE-1)=RANDOM DATA 1+
;TABLE SIZE-1
;ACTUAL TABLE BEGIN ADDR
;IS AT C(LOBMF)
;MAKE SURE C(TABLE-2) <
;C(TABLE-1)
;
;C(AC0)=RANDOM #
;GET RANDOM DATA 2 IN
;THE RANGE OF C(TABLE-2)
;AND C(TABLE-1)
;AC0=DISPATCH ADDR
;FACE=TABLE BEGIN ADDR
;SAVE AC0
;SAVE AC1
;
;SAVE DISPATCH ADDR
;AND STORE (DP02+3) IN
;DISPATCH ADDR LOCATION
;MAKE SURE THAT RNDM
;ADDR IS NOT EQ. TO
;(TABLE-1) OR (TABLE-2)
;C(AC0)=RANDOM #
;GET ADDRESS IN THE RANGE

```



10211 ESPCL

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
OSPE0:
EDSP1
FILL
JSR
XERR
SETUP
JSR
LDA
LDA
LDA
LDA
SUB
STA
RAND
JSR
D3
MXTBL
SBI
STA
CLM
JMP
RAND
JSR
LDA
LDA
LDA
LDA
LDA
SUB
STA
DSPA
MOV
ERROR
LDA
STA
LOOP
JSR
;HIGBF,HIBFU,XERR
;FILL UPPER AND LOWER
;SCRATCH BUFFER AREA
;WITH (XERR)
;INITIALIZE TEST
;ITERATION VALUE
;MAX. AVAILABLE AREA
;FOR TABLE IS =
;C(HIGBF)-C(HIBFU)
;C(HIGBF)
;MXTBL
;TABLE SIZE =
;1-RANDOM SIZE* C(MXTBL)-2
;C(ACO)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(D3) AND C(MXTBL)
;TABLE SIZE =
;1-RANDOM SIZE* C(MXTBL)-2
;C(ACO)=RANDOM #
;C(TABLE-2)= RANDOM DATA 1
;C(TABLE-1)=(RANDOM DATA 1+
;TABLE SIZE-1)
;ACTUAL TABLE BEGIN ADDR
;IS AT C(HIGBF)
;MAKE SUPE C(TABLE-2) <
;C(TABLE-1)
;
;C(ACO)=RANDOM #
;GET RANDOM DATA 2 IN
;THE RANGE OF C(TABLE-2)
;AND C(TABLE-1)
;ACO=DISPATCH ADDEND
;AC2=TABLE BEGIN ADDR
;AC1=17777
;STORE TBL ADDR IN
;DSPA INSTRUCTION AND
;17777 IN DISPATCH LOC.
;DSPA SHOULD RETURN TO
;DPE0+2
;ERROR CALL
;RESTORE DISPATCH LOC.
;XERR, XOP ERROR CALL
;ITERATE TEST ROUTINE

```

10212 ESPCL

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
OSPE1:
EDSP1
FILL
JSR
XERR
SETUP
JSR
LDA
LDA
LDA
LDA
SUB
STA
RAND
JSR
D3
MXTBL
SBI
STA
CLM
JMP
RAND
JSR
LDA
LDA
LDA
LDA
LDA
SUB
STA
DSPA
MOV
ERROR
LDA
STA
LOOP
JSR
;LOWBF,LOBFU,XERR
;FILL UPPER AND LOWER
;SCRATCH BUFFER AREA
;WITH (XERR)
;INITIALIZE TEST
;ITERATION VALUE
;MAX. AVAILABLE AREA
;FOR TABLE IS =
;C(LOBFU)-C(LOWBF)
;MXTBL
;TABLE SIZE =
;1-RANDOM SIZE* C(MXTBL)-2
;C(ACO)=RANDOM #
;GET ADDRESS IN THE RANGE
;C(D3) AND C(MXTBL)
;TABLE SIZE =
;1-RANDOM SIZE* C(MXTBL)-2
;C(ACO)=RANDOM #
;C(TABLE-2)= RANDOM DATA 1
;C(TABLE-1)=(RANDOM DATA 1+
;TABLE SIZE-1)
;ACTUAL TABLE BEGIN ADDR
;IS AT C(LOWBF)
;MAKE SURE C(TABLE-2) <
;C(TABLE-1)
;
;C(ACO)=RANDOM #
;GET RANDOM DATA 2 IN
;THE RANGE OF C(TABLE-2)
;AND C(TABLE-1)
;ACO=DISPATCH ADDEND
;AC2=TABLE BEGIN ADDR
;AC3=(TABLE ADDR/2)+
;BIT 15
;STORE 17777 IN DISPATCH
;LOCATION
;STORE (TABLE ADDR/2)
;IN DSPA INSTRUCTION
;DSPA SHOULD RETURN TO
;NEXT INSTRUCTION
;ERROR CALL
;RESTORE DISPATCH LOC TO
;JSR SENTER,
;ITERATE TEST ROUTINE

```



0213 ESPCL  
01

10214 ESPCL

```
01      D$PFO:  E$SP1  L$MBF,L$ORFU,X$ERR
02
03      X$ERR
04      @JFILL      :FILL UPPER AND LOWER
05 11502 006125   JSR      :SCRATCH BUFFER AREA
06 11503 100030   X$ERR      :WITH (X$ERR)
07
08      SETUP      100.
09 11504 006120   JSR @ENTR  :INITIALIZE TEST
10 11505 000144   LDA 100.      :ITERATION VALUE
11 11506 020132   LDA 0,L$MBF   :MAX. AVAILABLE AREA
12 11507 024123   LDA 1,L$ORFU  :FOR TABLE IS =
13 11510 106400   SUB 0,1
14 11511 044301   STA 1,M$TBL  :C(L$ORFU)=C(L$MBF)
15      RNDADR D3,M$TBL
16
17 11512 006121   JSR @ENTRA   :C(AC0)=RANDOM #
18 11513 006122   JSR @RNDADR  :GET ADDRESS IN THE RANGE
19 11514 000304   D3          :C(D3) AND C(M$TBL)
20 11515 000301   M$TBL
21 11516 120110   S#1 2,0
22 11517 000302   STA 0,T$LSIZ :TABLE SIZE =
23      RAND      :1-RANDOM SIZE< C(M$TBL)-2
24 11520 006121   JSR @ENTRA   :C(AC0)=RANDOM #
25 11521 030132   LDA 2,L$MBF   :C(M$TBL-2)=RANDOM DATA 1
26 11522 041000   STA 0,0,2     :C(M$TBL-1)=(RANDOM DATA 1+
27 11523 024302   LDA 1,T$LSIZ  :TABLE SIZE-1)
28 11524 107000   ADD 0,1
29 11525 104110   S#1 1,1
30 11526 045001   STA 1,1,2
31 11527 143370   HLV 0
32 11530 132370   CLM 1,2
33 11531 000770   JMP --8.
34      RAND
35 11532 006121   JSR @ENTRA   :C(AC0)=RANDOM #
36 11533 024132   LDA 1,L$MBF   :GET RANDOM DATA 2 IN
37 11534 106370   CLM 0,1       :THE RANGE OF C(TABLE-2)
38 11535 000775   JMP --3
39 11536 130010   ADI 2,2
40
41 11537 050427   STA 2,D$PFO+1
42 11540 028376   LDA 1,-2,2
43 11541 113000   ADD 0,2
44 11542 132400   SUB 1,2
45 11543 040267   STA 0,$SLC0
46 11544 050270   STA 2,$SLC1
47      HITBAD
48 11545 020124   LDA 0,H$IGBF  :MAKE SURE THAT RNDM
49 11546 120010   ADI 2,0       :ADDR IS NOT EQ. TO
50 11547 040275   STA 0,H$OSTB : (TABLE-1) OR (TABLE-2)
51      RNDADR H$OSTB,H$IGBF
52      RAND
53 11550 006121   JSR @ENTRA   :C(AC0)=RANDOM #
54 11551 006122   JSR @RNDADR  :GET ADDRESS IN THE RANGE
55 11552 000275   H$OSTB      :C(H$OSTB) AND C(H$IGBF)
56 11553 000133   H$IGBF
57 11554 105000   MOV 0,1
58 11555 127240   ADDUR 1,1
59 11556 034270   LDA 3,$SLC1  :RNDM ADDR IS STORED
60 11557 045400   STA 1,0,3    :IN DISPATCH ADDR LOC
```



10217 ESPCL

```

01
02
03
04 11620 006120 T\N177: JSR      100-      SETUP
05 11621 000144 100-      @ENTIN  ;INITIALIZE TEST
06 11622 162470 ELEF      0,DPF2+3--1,1 ;STORE ADDR (DPF2+3) IN
07 000012
08 11624 040413 STA      0,DPF20 ;DPF20
09 11625 162470 ELEF      0,DPF20--1,1 ;STORE (@DPF20) IN
10 000011
11 11627 103240 ADDR      0,0 ;DISPATCH LOC
12 11630 040412 STA      0,DP1B1+2 ;DP1B1+2
13 11631 150510 XDR      2,2 ;DISPATCH ADDEND IN AC2=0
14 11632 152570 DSPA      2,DP1B1+2--1,1 ;EXECUTE DSPA
15 000007
16
17 11634 100030 E\N263: XOP      0,0,0 ;ERROR CALL
18
19 11635 006117 L\N177: JSR      LOOP
20 11636 000406 JMP
21 11637 000000 DPF20: 0
22 11640 000000 DPTB1: 0
23 11641 000001
24 11642 000000
25 11643 177777
26

```

10218 ESPCL

```

01
02
03
04 11644 006120 T\N200: JSR      100-      SETUP
05 11645 000144 100-      @ENTIN  ;INITIALIZE TEST
06 11646 162470 ELEF      0,DPF3+3--1,1 ;STORE ADDR (DPF3+3) IN
07 000015
08 11650 040416 STA      0,DPF30 ;DPF30
09 11651 162470 ELEF      0,DPF30--1,1 ;STORE (@DPF30) IN
10 000014
11 11653 103240 ADDR      0,0 ;DISPATCH LOCATION
12 11654 040416 STA      0,DP1R2+3 ;DP1R2+3
13 11655 152520 SUBZL      2,2 ;DISPATCH ADDEND IN AC2=1
14 11656 162470 ELEF      0,DP1R2+1--1,1 ;STORE (TBL ADDR-1) IN
15 000011
16 11660 040402 STA      0,DPF3+1 ;(DPF3+1)
17 11661 153170 DPF3:  DSPA      2,0,2 ;EXECUTE DSPA
18 000000
19
20 11663 100030 E\N264: XOP      0,0,0 ;ERROR CALL
21 11664 006117 L\N200: JSR      LOOP
22 11665 000406 JMP
23 11666 000000 DPF30: 0
24 11667 000000 DPTB2: 0
25 11667 000001
26 11670 000001
27 11671 177777
28 11672 000000

```

```

10219 ESPCL
01
02
03
04 11673 006120 T&N201: JSR          SETUP
05 11674 000144 100.              :INITIALIZE TEST
06 11675 162470 ELEF          :ITERATION VALUE
07 11676 000015 0.0DPF40 :DPF40
08 11677 040416 0.0DPF40--1,1 :STORE (0DPF40) IN
09 11700 162470 0.0DPF40--1,1 :STORE (0DPF40) IN
10 000014
11 11702 103240 ADDR          :DISPATCH LOCATION
12 11703 040416 0.0OPTB3+3 :DPTB3+3
13 11704 176520 SUBZL          :DISPATCH ADDRESS IN AC3=1
14 11705 162470 ELEF          :STORE (T&L ADDR-1) IN
15 000011
16 11707 040402 STA          :(DPF4+1)
17 11710 157570 DSPA          :EXECUTE DSPA
18 000000
19
20 11712 100030 E&R265: XOP          ERROR
21 LOOP
22 11713 006117 L&N201: JSR          :ITERATE TEST ROUTINE
23 11714 000406 0.0DPF40 :DPF40: 0
24 11715 000000 OPTB3: 0
25 11716 000000 OPTB3: 0
26 11717 000001 1
27 11720 177777 177777
28 11721 000000 0
29
10220 ESPCL
01
02 11722 102470 JMP          CMLEND :*
03 002224
04 11724 054213 LOP:
05 11725 014207 DSZ ITRCT
06 11726 000433 JMP LOP3
07 11727 034210 LDA 3,ITRER :ITERATION COMPLETE
08 11730 175005 MOV 3,3,SNR :WHEN NO ERROR, EXIT TO NEXT
09 11731 002213 JMP @LOPRET
10 11732 034206 LDA 3,ITR
11 11733 054207 STA 3,ITRCT :RESET ITERATION COUNTER
12 11734 074277 LOP1: :LOOK AT SWITCH "0" TO SEE WHETHER SWREG
13 11735 176113 MOVL# 3,3,SNC
14 11736 036235 LDA 3,@ISWREG :*
15 11737 177100 ADDL 3,3
16 11740 177103 ADDL 3,3,SNC :IF SWITCH "3" PRINT "%M
17 11741 000416 JMP LOP2 :NO "%M PRINT OUT REQUIRED
18 11742 113110 PSH 0,2
19 PRINT PERCENT
20 11743 174430 XOP 3,3,4 :*PRINT MESSAGE PERCENT
21 11744 014717 PERCENT
22 11745 102400 SUB 0,0
23 11746 024211 LDA 1,ITREC :GET ERROR COUNT
24 11747 040211 STA 0,ITREC :CLEAR ERROR COUNT.
25 11750 030140 LDA 2,C144
26 11751 143710 MUL
27 11752 030206 LDA 2,ITR :(COUNT X 100)/ ITERATIONS=
28 11753 153710 DIV :PERCENTAGE OF FAILURE
29 11754 044435 STA 1,LOP4
30 DECP
31 11755 124330 XOP 1,1,3 :**PRINT (AC1) DECIMAL ZERO($) SUPPRESSED
32 11756 143210 POP 2,0
33 11757 176400 LOP2:
34 11760 054211 STA 3,ITREC
35 11761 036235 LOP3:
36 11762 175222 MOVZR 3,3,SZC :*SUPPRESS ITERATIONS SWT (15) = 1
37 11763 002213 JMP @LOPRET :*(YES) CONTINUE
38 11764 034210 LDA 3,ITRER :IF NO ERROR, ITERATE
39 11765 175004 MOV 3,3,SZR :OTHERWISE LOOK AT DATA
40 11766 074477 READS 3 : "1" SWITCH FOR PROCEED,
41 11767 175115 MOVL# 3,3,SNC
42 11770 036235 LDA 5,@ISWREG
43 11771 177113 JMP @ITRET
44 11772 002201 ADDL# 3,3,SNC
45 11773 157770 ANDI 40001,3 :*
46 040001
47 11775 054415 STA 3,LOP7 :*
48 11776 175015 MOV# 3,3,SNR :*SPECIAL
49 11777 034210 LDA 3,ITRER :*
50 12000 175404 INC 3,3,SZR :*
51 12001 000404 JMP LOP6 :*TEST
52 12002 034410 LDA 3,LOP7 :*
53 12003 175223 MOVZR 3,3,SNC :*
54 12004 002201 JMP @ITRET :*ERR ITERATE
55 12005 174510 XOR 3,3 :*CLR
56 12006 054373 STA 3,ERR :*SPECIAL
57 12007 002213 JMP @LOPHET :*EXIT
58 12010 063077 HALT
59 12011 000000 LOP4: 0
60 12012 000000 LOP7: 0

```



0223 FSPCL

```

01 XOP 1,1,1 CALL
02 MOVZ 3,3,3 ;RESET CARRY
03 JMP POCI+1 ;CONTINUE
04 XOP 1,1,3 CALL
05 MOVZ 3,3,3 ;RESET CARRY
06 JSR PDEC3 ;RESET C(CARRY) FOR ZERO SUPPRESSION
07 JSR PDEC3 ;SET C(CARRY) IF NOT
08 JSR PDEC3
09 JSR PDEC3
10 JSR PDEC3
11 JSR PDEC3
12 JSR PDEC3
13 JSR PDEC3
14 XOP 1,1,2 CALL
15 MOVZ 3,3,3 ;SET CARRY
16 JSR PDEC3 ;RESET C(CARRY) FOR ZERO SUPPRESSION
17 JSR PDEC3 ;SET C(CARRY) IF NOT
18 JSR PDEC3
19 JSR PDEC3
20 JSR PDEC3
21 JSR PDEC3
22 JSR PDEC3
23 JSR PDEC3
24 JSR PDEC3
25 JSR PDEC3
26 JSR PDEC3
27 JSR PDEC3
28 JSR PDEC3
29 JSR PDEC3
30 JSR PDEC3
31 JSR PDEC3
32 JSR PDEC3
33 JSR PDEC3
34 JSR PDEC3
35 JSR PDEC3
36 JSR PDEC3
37 JSR PDEC3
38 JSR PDEC3
39 JSR PDEC3
40 JSR PDEC3
41 JSR PDEC3
42 JSR PDEC3
43 JSR PDEC3
44 JSR PDEC3
45 JSR PDEC3
46 JSR PDEC3
47 JSR PDEC3
48 JSR PDEC3
49 JSR PDEC3
50 JSR PDEC3
51 JSR PDEC3

```

0224 ESPCL

```

01 XOP 3,3,4 CALL
02 JSR 1,5 ;ADVANCE RETURN
03 JMP *+2 ;***** FATAL ERROR HALT *****
04 JSR 3,3,3 ;***** FATAL ERROR HALT *****
05 JSR 3,1,3 ;SUGGEST YOU RUN DIAGNOSTICS
06 JSR 7777,3 ;REMOVE CRY
07 JSR 7777,3 ;MESSAGE ADDRESS
08 JSR 7777,3 ;MESSAGE ADDRESS
09 JSR 7777,3 ;MESSAGE ADDRESS
10 JSR 7777,3 ;MESSAGE ADDRESS
11 JSR 7777,3 ;MESSAGE ADDRESS
12 JSR 7777,3 ;MESSAGE ADDRESS
13 JSR 7777,3 ;MESSAGE ADDRESS
14 JSR 7777,3 ;MESSAGE ADDRESS
15 JSR 7777,3 ;MESSAGE ADDRESS
16 JSR 7777,3 ;MESSAGE ADDRESS
17 JSR 7777,3 ;MESSAGE ADDRESS
18 JSR 7777,3 ;MESSAGE ADDRESS
19 JSR 7777,3 ;MESSAGE ADDRESS
20 JSR 7777,3 ;MESSAGE ADDRESS
21 JSR 7777,3 ;MESSAGE ADDRESS
22 JSR 7777,3 ;MESSAGE ADDRESS

```

```

0225 ESPCL
01
02 ; RANDOM NUMBER GENERATOR
03
04 ;RANDOM DATA GENERATOR
05
06 :GENERATE A NEW RANDOM
07 :NUMBER IN C(AC0) AND C(RAN),
08 :IF C(ITRER)=0, OTHERWISE
09 :LOAD C(AC0) WITH OLD #.
10
11 :RAND: LDA 0,RAN
12 :MOV 1,1,ITRER
13 :MOV 0,1,SZR
14 :MOV 0,1
15 :ANDI 55,1
16
17 :MOVZL 1,1
18 :COB 1,1
19 :MOVR 1,1
20 :MOVR 0,0
21 :STA 0,RAN
22 :JMP 0,3
23
24 :PLSB OF "COB"
25 :MSR OF "RAN"
26 :NEW RAN. # IN AC0 & RAN
27
28 ; RANDOM ADDRESS GENERATOR
29
30 RNDMAD: STA 1,AC1
31 :IN THE RANGE (ARG1) TO (ARG2).
32 :EXIT WITH RESULTS IN AC0.
33
34 :ADD 1,0
35 :INC 2,2
36
37 :RNDM: LDA 1,AC1
38 :MOV 1,1
39 :LDA 2,AC2
40 :MOV 1,1
41 :LDA 2,AC2
42 :MOV 1,1
43 :LDA 2,AC2
44 :MOV 1,1
45 :LDA 2,AC2
46 :MOV 1,1
47 :LDA 2,AC2
48 :MOV 1,1
49 :LDA 2,AC2
50 :MOV 1,1
51 :LDA 2,AC2
52 :MOV 1,1
53 :LDA 2,AC2
54 :MOV 1,1
55 :LDA 2,AC2
56 :MOV 1,1
57 :LDA 2,AC2
58 :MOV 1,1
59 :LDA 2,AC2
60 :MOV 1,1
61 :LDA 2,AC2
62 :MOV 1,1
63 :LDA 2,AC2
64 :MOV 1,1
65 :LDA 2,AC2
66 :MOV 1,1
67 :LDA 2,AC2
68 :MOV 1,1
69 :LDA 2,AC2
70 :MOV 1,1
71 :LDA 2,AC2
72 :MOV 1,1
73 :LDA 2,AC2
74 :MOV 1,1
75 :LDA 2,AC2
76 :MOV 1,1
77 :LDA 2,AC2
78 :MOV 1,1
79 :LDA 2,AC2
80 :MOV 1,1
81 :LDA 2,AC2
82 :MOV 1,1
83 :LDA 2,AC2
84 :MOV 1,1
85 :LDA 2,AC2
86 :MOV 1,1
87 :LDA 2,AC2
88 :MOV 1,1
89 :LDA 2,AC2
90 :MOV 1,1
91 :LDA 2,AC2
92 :MOV 1,1
93 :LDA 2,AC2
94 :MOV 1,1
95 :LDA 2,AC2
96 :MOV 1,1
97 :LDA 2,AC2
98 :MOV 1,1
99 :LDA 2,AC2
100 :MOV 1,1

```

```

0226 ESPCL
01
02 ; RANDOM NUMBER GENERATOR
03
04 ;RANDOM DATA GENERATOR
05
06 :GENERATE A NEW RANDOM
07 :NUMBER IN C(AC0) AND C(RAN),
08 :IF C(ITRER)=0, OTHERWISE
09 :LOAD C(AC0) WITH OLD #.
10
11 :RAND: LDA 0,RAN
12 :MOV 1,1,ITRER
13 :MOV 0,1,SZR
14 :MOV 0,1
15 :ANDI 55,1
16
17 :MOVZL 1,1
18 :COB 1,1
19 :MOVR 1,1
20 :MOVR 0,0
21 :STA 0,RAN
22 :JMP 0,3
23
24 :PLSB OF "COB"
25 :MSR OF "RAN"
26 :NEW RAN. # IN AC0 & RAN
27
28 ; RANDOM ADDRESS GENERATOR
29
30 RNDMAD: STA 1,AC1
31 :IN THE RANGE (ARG1) TO (ARG2).
32 :EXIT WITH RESULTS IN AC0.
33
34 :ADD 1,0
35 :INC 2,2
36
37 :RNDM: LDA 1,AC1
38 :MOV 1,1
39 :LDA 2,AC2
40 :MOV 1,1
41 :LDA 2,AC2
42 :MOV 1,1
43 :LDA 2,AC2
44 :MOV 1,1
45 :LDA 2,AC2
46 :MOV 1,1
47 :LDA 2,AC2
48 :MOV 1,1
49 :LDA 2,AC2
50 :MOV 1,1
51 :LDA 2,AC2
52 :MOV 1,1
53 :LDA 2,AC2
54 :MOV 1,1
55 :LDA 2,AC2
56 :MOV 1,1
57 :LDA 2,AC2
58 :MOV 1,1
59 :LDA 2,AC2
60 :MOV 1,1
61 :LDA 2,AC2
62 :MOV 1,1
63 :LDA 2,AC2
64 :MOV 1,1
65 :LDA 2,AC2
66 :MOV 1,1
67 :LDA 2,AC2
68 :MOV 1,1
69 :LDA 2,AC2
70 :MOV 1,1
71 :LDA 2,AC2
72 :MOV 1,1
73 :LDA 2,AC2
74 :MOV 1,1
75 :LDA 2,AC2
76 :MOV 1,1
77 :LDA 2,AC2
78 :MOV 1,1
79 :LDA 2,AC2
80 :MOV 1,1
81 :LDA 2,AC2
82 :MOV 1,1
83 :LDA 2,AC2
84 :MOV 1,1
85 :LDA 2,AC2
86 :MOV 1,1
87 :LDA 2,AC2
88 :MOV 1,1
89 :LDA 2,AC2
90 :MOV 1,1
91 :LDA 2,AC2
92 :MOV 1,1
93 :LDA 2,AC2
94 :MOV 1,1
95 :LDA 2,AC2
96 :MOV 1,1
97 :LDA 2,AC2
98 :MOV 1,1
99 :LDA 2,AC2
100 :MOV 1,1

```

```

0227 ESPCL
01
02 ; RANDOM NUMBER GENERATOR
03
04 ;RANDOM DATA GENERATOR
05
06 :GENERATE A NEW RANDOM
07 :NUMBER IN C(AC0) AND C(RAN),
08 :IF C(ITRER)=0, OTHERWISE
09 :LOAD C(AC0) WITH OLD #.
10
11 :RAND: LDA 0,RAN
12 :MOV 1,1,ITRER
13 :MOV 0,1,SZR
14 :MOV 0,1
15 :ANDI 55,1
16
17 :MOVZL 1,1
18 :COB 1,1
19 :MOVR 1,1
20 :MOVR 0,0
21 :STA 0,RAN
22 :JMP 0,3
23
24 :PLSB OF "COB"
25 :MSR OF "RAN"
26 :NEW RAN. # IN AC0 & RAN
27
28 ; RANDOM ADDRESS GENERATOR
29
30 RNDMAD: STA 1,AC1
31 :IN THE RANGE (ARG1) TO (ARG2).
32 :EXIT WITH RESULTS IN AC0.
33
34 :ADD 1,0
35 :INC 2,2
36
37 :RNDM: LDA 1,AC1
38 :MOV 1,1
39 :LDA 2,AC2
40 :MOV 1,1
41 :LDA 2,AC2
42 :MOV 1,1
43 :LDA 2,AC2
44 :MOV 1,1
45 :LDA 2,AC2
46 :MOV 1,1
47 :LDA 2,AC2
48 :MOV 1,1
49 :LDA 2,AC2
50 :MOV 1,1
51 :LDA 2,AC2
52 :MOV 1,1
53 :LDA 2,AC2
54 :MOV 1,1
55 :LDA 2,AC2
56 :MOV 1,1
57 :LDA 2,AC2
58 :MOV 1,1
59 :LDA 2,AC2
60 :MOV 1,1
61 :LDA 2,AC2
62 :MOV 1,1
63 :LDA 2,AC2
64 :MOV 1,1
65 :LDA 2,AC2
66 :MOV 1,1
67 :LDA 2,AC2
68 :MOV 1,1
69 :LDA 2,AC2
70 :MOV 1,1
71 :LDA 2,AC2
72 :MOV 1,1
73 :LDA 2,AC2
74 :MOV 1,1
75 :LDA 2,AC2
76 :MOV 1,1
77 :LDA 2,AC2
78 :MOV 1,1
79 :LDA 2,AC2
80 :MOV 1,1
81 :LDA 2,AC2
82 :MOV 1,1
83 :LDA 2,AC2
84 :MOV 1,1
85 :LDA 2,AC2
86 :MOV 1,1
87 :LDA 2,AC2
88 :MOV 1,1
89 :LDA 2,AC2
90 :MOV 1,1
91 :LDA 2,AC2
92 :MOV 1,1
93 :LDA 2,AC2
94 :MOV 1,1
95 :LDA 2,AC2
96 :MOV 1,1
97 :LDA 2,AC2
98 :MOV 1,1
99 :LDA 2,AC2
100 :MOV 1,1

```







```

0231 ESPCL
01 1254 106470 MAPIT: EJSR STKO
02 000674 DIA 1,MMPU :GET THE CURRENT USER MAP
03 12526 064803 XORI 404.1
04 12527 12770 XORI 404.1
05 12531 040162 STA 1,MMPUI
06 12532 12770 XORI 404.1
07 12534 14770 ANDI 20000.1 :GET THE USER
08 020000
09 12536 105510 HXR 1.1 ?
10 12537 125220 MOVZR 1.1 ?
11 12540 12770 XORI 400.1 :COMPLEMENT THE USER
12 000400
13 12542 10770 IORI 1000.1 ?
14 001000
15 12544 04157 STA 1,USRMAP:SAVE THE NEW USER
16 12545 020154 LDA 8.0 ?
17 12546 163770 ADDI
18 000010
19 12550 030153 LDA 2,DTOSLK:IS THIS DTOS LK?
20 12551 151004 MOV 2,2,SZR ?
21 12552 11110 SGE 0.2 ?
22 12553 000405 JMP +5 ?
23 12554 115000 MOV 0.3 ?
24 12555 17770 ADDI -8.3 ?
25 17770
26 12557 155010 SGT 2.3 ?
27 12560 000402 JMP +2 ?
28 12561 100010 AOI 1.0 ?
29 12562 040154 STA 0,MPBGN :SET UP MAP BEGIN
30 12563 176070 ELEF 3.20000.0 ?
31 020000
32 12565 152400 SUB 2.2 ?
33 12566 050216 STA 2,TESTN :TALLY
34 12567 050375 STA 2,ERRR :LINK
35 12570 166070 ELEF 1,PRGEND+1.0 ?
36 015553
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51 12572 117110 PSH 0.3 ?
52 12573 133710 BLM :MOVE PROGRAM
53 12574 163210 POP 3.0 ?
54 12575 117110 PSH 0.3 ?
55 12576 173770 ADDI 17.2 :OFFSET
56 000017
57 12600 177770 ADDI 17.3 :SCAN
58 000017
59 12602 167770 ADDI -20.1 :BY 20
60 177760

; MAPIIT, MOVES 1ST BK TO NEXT 8K AND MAPS
; NEXT 32K TO 1ST 32K
;
MAPIT: EJSR STKO
DIA 1,MMPU :GET THE CURRENT USER MAP
XORI 404.1
STA 1,MMPUI
XORI 404.1
ANDI 20000.1 :GET THE USER
HXR 1.1 ?
MOVZR 1.1 ?
XORI 400.1 :COMPLEMENT THE USER
IORI 1000.1 ?
STA 1,USRMAP:SAVE THE NEW USER
LDA 8.0 ?
ADDI
LDA 2,DTOSLK:IS THIS DTOS LK?
MOV 2,2,SZR ?
SGE 0.2 ?
JMP +5 ?
MOV 0.3 ?
ADDI -8.3 ?
SGT 2.3 ?
JMP +2 ?
AOI 1.0 ?
STA 0,MPBGN :SET UP MAP BEGIN
ELEF 3.20000.0 ?
SUB 2.2 ?
STA 2,TESTN :TALLY
STA 2,ERRR :LINK
ELEF 1,PRGEND+1.0 ?
BLM TRANSFER VERIFICATION SECTION
;VERIFY SOURCE BUFFER EQUALS DESTINATION
;FOLLOWING BLM EXECUTION
;
NOTE:
UNLIKE RELOCATION NON-MAPPED IN WHICH A "COB"
CHECK WORD WAS USED, THIS XFER VERIFICATION IS WORD
FOR WORD, WITH THE EXCEPTION OF LOCATIONS 0 THRU 16
PSH 0.3 ?
BLM :MOVE PROGRAM
POP 3.0 ?
PSH 0.3 ?
ADDI 17.2 :OFFSET
ADDI 17.3 :SCAN
ADDI -20.1 :BY 20

```

```

0232 ESPCL
01 12604 044007 STA 1,XAC3 :SCAN COUNT
02 12605 151400 INC 2.2 :SOURCE ADDRESS (NEXT)
03 12606 175400 INC 5.3 :DEST ADDRESS (NEXT)
04 12607 021000 LDA 0,0.2 :SOURCE ENTRY > ACO
05 12610 021000 LDA 1,0.3 :DEST ENTRY > AC1
06 12610 025400 SURZ# 1,0,SNR :EQUAL ?
07 12612 000403 JMP XMOK :OK BYPASS
08
09 ERROR
10 12613 100030 EARR267: XOP 0,0.0 :ERROR CALL
11
12 *NOTE "CAUTION" BELOW !!!
13
14 *
15
16 ERROR SIGNIFICANCE
17
18 (AC0) (AC1) (AC2) (AC3)
19 CONTENT CONTENT ADDRESS ADDRESS
20 SOURCE DEST. SOURCE DEST.
21 LOC. LOC. LOC. LOC.
22
23 CAUTION!
24
25 *****
26 A RESTART AT LABEL "RETRY:" MAY BE USED TO VERIFY THE
27 ABILITY OF THE BLOCK MOVE ROUTINE TO XFER INFORMATION
28 FOLLOWING AN ERROR AT LOCATIONS "XFERB" OR "XFERM".
29 THE INTEGRITY OF THE INFORMATION AND/OR RESULTS WILL
30 BE IN QUESTION HOWEVER; SO THE USER IS ADVISED TO
31 RUN THE BASIC ECLIPSE DIAGNOSTICS FOLLOWING ERRORS
32 AT "XFERB:" OR "XFERM:".
33
34 XFERB: HALTA 2 ***** FATAL ERROR HALT *****
35 MAF040: :FATAL ERROR XFERM "BLM"
36 :SUGGEST THAT YOU RUN
37 :BASIC ECLIPSE DIAGNOSTICS
38
39 XNOK:
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

```

```

01 12604 044007 STA 1,XAC3 :SCAN COUNT
02 12605 151400 INC 2.2 :SOURCE ADDRESS (NEXT)
03 12606 175400 INC 5.3 :DEST ADDRESS (NEXT)
04 12607 021000 LDA 0,0.2 :SOURCE ENTRY > ACO
05 12610 021000 LDA 1,0.3 :DEST ENTRY > AC1
06 12610 025400 SURZ# 1,0,SNR :EQUAL ?
07 12612 000403 JMP XMOK :OK BYPASS
08
09 ERROR
10 12613 100030 EARR267: XOP 0,0.0 :ERROR CALL
11
12 *NOTE "CAUTION" BELOW !!!
13
14 *
15
16 ERROR SIGNIFICANCE
17
18 (AC0) (AC1) (AC2) (AC3)
19 CONTENT CONTENT ADDRESS ADDRESS
20 SOURCE DEST. SOURCE DEST.
21 LOC. LOC. LOC. LOC.
22
23 CAUTION!
24
25 *****
26 A RESTART AT LABEL "RETRY:" MAY BE USED TO VERIFY THE
27 ABILITY OF THE BLOCK MOVE ROUTINE TO XFER INFORMATION
28 FOLLOWING AN ERROR AT LOCATIONS "XFERB" OR "XFERM".
29 THE INTEGRITY OF THE INFORMATION AND/OR RESULTS WILL
30 BE IN QUESTION HOWEVER; SO THE USER IS ADVISED TO
31 RUN THE BASIC ECLIPSE DIAGNOSTICS FOLLOWING ERRORS
32 AT "XFERB:" OR "XFERM:".
33
34 XFERB: HALTA 2 ***** FATAL ERROR HALT *****
35 MAF040: :FATAL ERROR XFERM "BLM"
36 :SUGGEST THAT YOU RUN
37 :BASIC ECLIPSE DIAGNOSTICS
38
39 XNOK:
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

```

```

0233 ESPCL
01 12642 175015 MOV# 3,3,SNR
02 12643 000421 JMP XMAPW
03 12644 020162 LDA 0,MMPUI
04 12645 143770 ANDI 400,0
05 12646 000400 IORI 10,0
06 12647 103770 DDA 0,MMPU
07 12648 061003 STA 0,LAST
08 12649 040372 XDR 0,0
09 12650 100510 ELEF 2,MAPW
10 12651 172470 SUBZL 1,1
11 12652 126520 LMP 14
12 12653 113410 LDA 0,MAPW
13 12654 020535 ADDI 185*1815,0
14 12655 163770 JMP *+5
15 12656 000405 DDA 0,MMPU
16 12657 061003 STA 0,LAST
17 12658 163770 ADDI 185*1815,0
18 12659 002001 STA 0,MAPW
19 12660 040525 POP 3,3
20 12661 177210 MOVL# 0,SZC
21 12662 101112 JMP DN32K
22 12663 000416 LDA 2,DTOSIK
23 12664 115000 MOV 0,3
24 12665 157770 MAPM: ANDI 377,3
25 12666 156414 SUB# 2,3,SZR
26 12667 000404 JMP *+4
27 12668 101400 INC 0,0
28 12669 060512 STA 0,MAPW
29 12670 175400 INC 3,3
30 12671 030152 LDA 2,HIGIK
31 12672 171010 SGT 3,2
32 12673 000731 JMP LMAPX
33 12674 000446 JMP DNALL

0234 ESPCL
01 12711 143770 UN32K: ANDI 187,0
02 12712 000400 MXL 1,0
03 12713 101410 MOVOL 0,0
04 12714 101140 LDA 1,MAPF
05 12715 024161 MOV# 1,1,SZR
06 12716 125014 LDA 0,MMPUI
07 12717 020162 IORI 11,0
08 12718 103770 MOV# 1,1,SNR
09 12719 000011 JMP *+3
10 12720 125015 ANDI 415,0
11 12721 000403 LDA 1,NSCAN
12 12722 143770 DDA 0,MMPU
13 12723 000415 STA 0,LAST
14 12724 143770 LDA 2,MAPW
15 12725 061003 LDA 0,MMPW
16 12726 040372 LDA 0,MMPW
17 12727 024283 LDA 2,MAPW
18 12728 024283 STA 1,NSCAN
19 12729 044374 STA 0,HIGIK
20 12730 020152 MOV# 3,3,SNR
21 12731 175015 XCM 0,3
22 12732 114710 STA 3,MPEND
23 12733 050462 LDA 0,PASSV
24 12734 050462 STA 0,PASSV
25 12735 020205 LDA 0,PASSV
26 12736 040204 STA 0,PASSV
27 12737 020087 LDA 0,PASSV
28 12738 040132 STA 0,PASSV
29 12739 102000 ADC 0,0
30 12740 040150 STA 0,MPSW
31 12741 102400 SUB 0,0
32 12742 040241 STA 0,MDFLG
33 12743 040241 STA 0,RELOC
34 12744 106470 EJSR STK0
35 12745 000406 JMP BAMBZ
36 12746 000073 JMP BAMBZ
37 12747 013015 MAPM: ROUTINES AND START TEST
38 12748 013015 DUMMY: MAPM
39 12749 000446 JMP DUMMY LOCATION

```

```

0235 ESPCL
01 12756 020161 DNALL: LDA
02 12757 101005 MOV 0,MAPP
03 12760 000407 JMP 0,0,SNR
04 12761 020162 LDA DALL
05 12762 123770 XORI 0,MMPU1
06 000404 ANDI 404,0
07 12764 143770 ANDI 404,0
08 000404 OR 0,MMPU
09 12766 061003 DOA ?
10 12767 062677 DALL: IDRS ?
11 12770 040425 STA 0,MAPW
12 12771 102000 ADC 0,0
13 12772 062177 DOBS 0,CPU
14 12773 020156 LDA 0,MRSB
15 12774 040154 STA 0,MPRGN
16 12775 102400 SUB 0,0
17 12776 040214 STA 0,RELOC
18 12777 040241 STA 0,MDFLG
19 13000 040155 STA 0,MPEND
20 13001 040150 STA 0,MPSM
21 13002 024134 LDA 1,BLKZ
22 13003 172470 LDA 2,RAMBLK
23 13005 065100 ELEM 3,BAMBLK
24 13005 176470 ELEM 3,BAMBLK
25 065111 ?
26 13007 113710 ?
27 13010 022354 LDA 0,ATCATSW
28 13011 101004 MOV 0,0,SZR
29 13012 006236 JSR ATCAT
30 13013 102470 NJTPAS
31 001147 ?
32 13015 000000 MAPW: 0

0236 ESPCL
01 13016 030217 SIZE: LDA 2,MINLOC
02 13017 151400 INC 2,2
03 13020 151112 MOVL# 2,2,SZC
04 13021 000411 JMP CKMAP
05 13022 021000 LDA 0,0,2
06 13023 051000 STA 2,0,2
07 13024 025000 LDA 1,0,2
08 13025 041000 STA 0,0,2
09 13026 132414 SUB# 1,2,SZR
10 13027 000403 JMP CKMAP
11 13030 050062 STA 2,MENTOP
12 13031 000766 JMP SIZE+1
13 000766 ?
14 13032 172070 ?
15 13033 076000 ?
16 13034 021000 LDA 0,0,2
17 13035 040425 STA 0,MAPT1
18 13036 020423 LDA 0,MAPC1
19 13037 041000 STA 0,0,2
20 13040 100000 COM 0,0
21 13041 040000 STA 0,0
22 13042 126400 SUH 1,1
23 13043 065003 DOA 1,MMPU
24 13044 044161 STA 1,MAPP
25 13045 044750 STA 1,MAPW
26 13046 044162 STA 1,MMPU1
27 13047 066003 OOB 1,MMPU
28 13050 025000 LDA 0,0,2
29 13051 106414 SUB# 0,1,SZR
30 13052 000402 JMP NOMAP
31 13053 000410 GOTMAP
32 13054 126400 NO:MAP
33 13055 044151 STA 1,1
34 13056 030062 LDA 1,MAPS
35 13057 151400 INC 2,2
36 13060 001400 JMP 0,3
37 13061 125252 MAPC1: 125252
38 000000 MAPT1: 0

: SIZE, THIS ROUTINE CHECKS FOR PRESENCE OF MMPU AND SIZES
: THE MEMORY FOR 80TH CASES
:
: LDA 2,MINLOC : SIZE THE LOGICAL MEMORY
: INC 2,2
: MOVL# 2,2,SZC
: JMP CKMAP
: MEM IS 32K ,CHECK FOR MMPU
: LDA 0,0,2
: STA 2,0,2
: LDA 1,0,2
: STA 0,0,2
: SUB# 1,2,SZR
: JMP CKMAP
: STA 2,MENTOP:SAVE MEMORY TOP ADDR
: JMP SIZE+1
:
: CHECK IF MMPU PRESENT
: CKMAP: ELEM 2,76000,0:
: LDA 0,0,2
: STA 0,MAPT1 :SAVE C(76000)
: LDA 0,MAPC1
: STA 0,0,2
: COM 0,0
: STA 0,0
: SUH 1,1
: DOA 1,MMPU :PRIME MAP STATUS INITIALLY = 0
: STA 1,MAPP
: STA 1,MAPW
: STA 1,MMPU1
: OOB 1,MMPU :MAP PHYS 37 TO PHYS 0
: LDA 0,0,2
: SUB# 0,1,SZR
: JMP NOMAP :MAP NOT PRESENT
: GOTMAP :MAP PRESENT
: NO:MAP
: STA 1,1
: LDA 1,MAPS :CLEAR MAP SW
: LDA 2,MENTOP :RETURN WITH C(MENTOP)+1 IN
: INC 2,2
: JMP 0,3 :AC2 IF NO MAP
: MAPC1: 125252
: MAPT1: 0

```

```

0237 ESPCL
01
02
03
04 13063 040151 GOTMAP: STA 0,MAPSW ;SET MAPSW
05 13064 117110 PSH 0,3 ;
06 13065 106470 EJSR CAMP ; CHECK FOR MMPU/MMPUI
07
08 13067 163210 POP 3,0 ;
09 13070 02072 LDA 0,MAPT1 ;RESTORE C(76000)
10 13071 041000 STA 0,0,2 ;
11 13072 162070 GMSK: ELEF 0,177,0 ;
12 000177
13 13074 062003 MORMAP: DOB 0,MMPU ;
14 13075 041000 STA 0,0,2 ;
15 13076 025000 LDA 1,0,2 ;
16 13077 122415 SUB# 1,0,SNR ;
17 13100 000403 JMP -* ;
18 13101 100110 SKI 1,0 ;
19 13102 000772 JMP MORMAP ;
20
21 13103 040152 STA 0,HIG1K ;
22 13104 124510 NEWMAP: XDR 1,1 ;
23 13105 065003 GDA 1,MMPU ;PRESET "B" FIRST
24 13106 022165 LDA 0,0,EGGS;*
25 13107 101004 MOV 0,0,SZR ;IS IT =0?
26 13110 000403 JMP -* ;**NO,THIS IS DTOS MODE
27 13111 022234 LDA 0,CICATS;*
28 13112 101000 MOV 0,0,SZR ;*IS CAT SW SET?
29 13113 020962 LDA 0,MENTOP;YES,SAVE TOP 1K FOR CAT
30 13114 101500 MOV# 0,0 ;
31 13115 101220 MOVZR 0,0 ;
32 13116 101220 MOVZR 0,0 ;
33 13117 143770 ANDI 37,0 ;
34 000037
35 13121 040153 STA 0,DTOS1K ;
36 13122 062677 IORST ;
37 13123 102400 SUB 0,0 ;
38 13124 040154 LDA 2,MENTOP;EXIT WITH C(AC2)=C(MENTOP)+1
39 13125 050062 INC 2,2 ;
40 13126 151400 JMP 0,3 ;
41 13127 001400

```

```

0238 ESPCL
01
02
03
04
05
06
07
08 13130 000000
09 13131 012677
10 13132 000400
11 13133 077777
12 13134 020774 EMZ:
13 13135 040152 EMZ1:
14 13136 034774 EMZ1:
15 13137 101400
16 13140 116435
17 13141 000423
18 13142 062003
19 13143 032370
20 13144 154000
21 13145 056766
22 13146 056765
23 13147 052764
24 13150 156405
25 13151 000413
26 13152 040152
27
28 13153 117110
29 13154 172070
30 078000
31 13156 155000
32 13157 166070
33 002000
34 13161 133710
35 13162 163210
36
37
38 13163 000753
39 13164 020152 EMZ2:
40 13165 101120
41 13166 103770
42 000377
43 13170 042741
44 13171 163210
45 13172 020670
46 13173 041000
47 13174 000710

```

```

MMPUI PHY MEMORY SIZER MASK CONTROL SUBROUTINE
ASSUMES CONTIGUOUS PHYSICAL MEMORY ALLOCATION
MODIFIES LOC. MAPM+1, AND FLUSH PHYSICAL ERCC
MEMORY PRIOR TO MODE 3 ENTRY.
0
MAPM+1 ;
000
77777
LDA 0,EMZ-4 ;
STA 0,HIG1K ;
LDA 3,EMZ-2 ;400
IMC 0,0 ;
SURZ# 0,3,SNR ;
JMP EMZ2 ;
00B 0,MMPU ;
LDA 2,EMZ-1 ;
COM 2,3 ;
STA 3,EMZ-1 ;
LDA 3,EMZ-1 ;
STA 2,EMZ-1 ;
SUM 2,3,SNR ;
JMP EMZ2 ;DONE
STA 0,HIG1K ;
PSH 0,3 ;*FLUSH ERCC MODE 2
ELEF 2,76000,0,*LAST 1K
MOV 2,3 ;*SRC = DEST READ/WRITE
ELEF 1,2000,0,*# WORDS IN 1K BLOCK
PLM ;*DOIT ERROR IN ERCC HERE TO
POP 3,0 ;*INDICATES IORST FAILED TO
;*ENTER ERCC MODE 2 (10)
JMP EMZ1 ;
LDA 0,HIG1K ;
MOVZL 0,0 ;
TONI 377,0 ;*MASK
STA 0,EMZ-3 ;
POP 3,0 ;
LDA 0,MAPT1 ;
STA 0,0,2 ;RESTORE C(76000)
JMP NEWMAP ;

```



```

0241 ESPCL
01 13304 124530
02
03 13305 174430
04 13306 015156
05 13307 024154
06
07 13310 124330
08 13311 024155
09
10 13312 124330
11
12 13313 174430
13 13314 015206
14 13315 020153
15 13316 101005
16 13317 000406
17
18 13320 174430
19 13321 015216
20 13322 024153
21
22 13323 124330
23
24 13324 124530
25 13325 034201 MAPDW:
26 13326 175004
27 13327 024006
28 13330 010241
29 13331 000401
30 13332 106470
31
32 13334 002401
33 13335 000000 MAPDX: 0
34 13336 014620 DMMD:
35 13337 014641

```

```

XOP 1,1,5      ;*OUTPUT CR/LF
PRINT MS10
XOP 3,3,4      ;*PRINT MESSAGE MS10
MS10
LDA 1,MPBGN
DECP
XOP 1,1,3      ;*PRINT (AC1) DECIMAL ZERO(S) SUPPRESSED
LDA 1,MPEND
DECP
XOP 1,1,3      ;*PRINT (AC1) DECIMAL ZERO(S) SUPPRESSED
PRINT MS11
XOP 3,3,4      ;*PRINT MESSAGE MS11
MS11
LDA 0,DTOS1K;IS DTOS1K = 0 ?
MOV 0,0,SNR ;NO,GO TO PRINT
JMP MAPDW ;YES,EXIT
PRINT MS12
XOP 3,3,4      ;*PRINT MESSAGE MS12
MS12
LDA 1,DTOS1K
DECP
XOP 1,1,3      ;*PRINT (AC1) DECIMAL ZERO(S) SUPPRESSED
CRLEPR
XOP 1,1,5      ;*OUTPUT CR/LF
LDA 3,MDFLG
MOV 3,3,SZR
JMP MAPDX
ISZ MDFLG
JMP *+1
EJSR DMMD
JMP MAPDX
0
DMMD
DMMD:

```

```

0242 ESPCL
01
02 ; SCL HANDLER ROUTINE
03
04 13340 101000 SCLCL: MOV 0,0
05 13341 107710 ;RETURN
06
07 ; PFLTA, PROTECTION FAULT HANDLER
08
09
10 13342 060403 PFLTA: DIA 0,MMPU ;C(AC0)=CURRENT USER STATUS
11 13343 066403 ;C(AC1)=CURRENT MAP STATUS
12 13344 150510 XDR 2,2
13 13345 174510 XDR 3,3
14 13346 070402 DIA 2,ERCC ;C(AC2)=MEM FLT ADDR.
15 13347 075402 DIB 3,ERCC ;C(AC3)=MEM FLT CODE.
16
17 13350 100030 EVR270: XOP 0,0,0 ;ERROR CALL
18 ;SEE RESTRICTION BELOW" !!!
19
20 13351 073077 H\F041: HALTA 2 ;***** FATAL ERROR HALT *****
21 13352 000777 JMP *-1 ;*NO GO FATAL ERROR
22
23 ;ERRNUM SIGNIFICANCE
24 ; MULTILEVEL INDIRECT PROTECTION (MMPU/MMPUI -
25 ; STATUS BIT<12>) IS THE ONLY PROTECTION ENABLED IN MAP
26 ; MODE EXECUTION. THIS FAULT SHOULD NOT OCCUR (!!!) AND
27 ; REPRESENTS A FATAL HARDWARE FAILURE.
28 ; ON SYSTEMS WITH ERCC THE PROGRAM RUNS WITH ERROR
29 ; CHECKING ON. THIS MAY RESULT IN A PROTECTION FAULT.
30 ; THIS ALSO IS A FATAL HARDWARE FAILURE.
31 ; IN EITHER CASE IT IS SUGGESTED THE USER RUN THE
32 ; BASIC ECLIPSE DIAGNOSTICS.
33
34 ; READ MAP STATUS
35 ; USER STATUS
36 ; DIA AC,MAP 018 AC,MAP DIA AC,ERCC DIR AC,ERCC
37 ; (AC0) (AC1) (AC2) (AC3)
38
39 ;FOR FURTHER INFORMATION SEE:
40 ; *PROGRAMMERS REFERENCE MANUAL, ECLIPSE LINE COMPUTERS"
41 ; (ORDER #: 015-000024-04)
42 ; *PAGE: 3-40 3-40 4-10 4-10
43
44 ;RESTRICTION:
45 ;-----
46 ;
47 ; IF RUNNING "AUTO ACCEPT" MODE (SHREG<15>=1) UNDER DTOS,
48 ; THE PROGRAM WILL ABORT.
49
50
51
52
53
54
55
56

```

```

THIS FAILURE IS CONSIDERED TO BE CATASTROPHIC IN
NATURE AND THERE IS A GOOD POSSIBILITY THAT THE
ERROR MESSAGE PRINTED MAY BE INCORRECT.

```

```

0243  ESPL
01
02
03
04 13353 000000
05 13354 000000
06 13355 054443 PWRLO:
*07
08 13356 174430 PRINT MESSAGE HEADER
09 13357 014724 XOP 3,1,3,4
10 13360 024216 LDA 1,TESTN ;*TEST# (TALLY)
11
12 13361 124130 XOP 1,1,1 ;**PRINT (AC1) OCTAL ZERO(S) SUPPRESS
13 13362 105210 POP 0,0 ;POP PC + CRY
14 13363 124510 XOP 1,1
MOVZL# 0,0,SZC
15 13364 101132 ADI 1,1
16 13365 104010 ANDI 77777,0
17 13366 143770
18 13370 077777
19 13370 044763 STA 1,PWRLO-2
20
OCTZ
21 13371 124130 XOP 1,1,1 ;**PRINT (AC1) OCTAL ZERO(S) SUPPRESS
22 13372 040762 STA 0,PWRLO-1
23 13373 165210 POP 3,0 ;POP AC0 - AC3
24 13374 104710 XCH 0,1
25
OCTP
26 13375 124230 XOP 1,1,2 ;**PRINT (AC1) OCTAL
XCH 0,1
27 13376 104710
28
OCTP
29 13377 124230 XOP 1,1,2 ;**PRINT (AC1) OCTAL
XCH 1,2
30 13400 130710
31
OCTP
32 13401 124230 XOP 1,1,2 ;**PRINT (AC1) OCTAL
XCH 1,3
33 13402 134710
34
OCTP
35 13403 124230 XOP 1,1,2 ;**PRINT (AC1) OCTAL
LDA 1,PWRLO-1
36 13404 024750 LDA 0,RELOC
37 13405 020214 LDA 0,1
38 13406 106400 SUB
39
OCTZ
40 13407 124130 XOP 1,1,1 ;**PRINT (AC1) OCTAL ZERO(S) SUPPRESS
LDA 1,PWRLO-1 ;*GET LOGICAL PC
41 13410 024744
42
OCTZ
43 13411 124130 XOP 1,1,1 ;**PRINT (AC1) OCTAL ZERO(S) SUPPRESS
EJSR PHYP ;*GET PHYSICAL PC
44 13412 106470
45 000410
46
CRLFPR
47 13414 124530 XOP 1,1,5 ;*OUTPUT CR/LF
EJSR PFLR ;PRINT FLOATING POINT REGISTERS
48
CRLFPR
49
SUBZL 0,0
50
MOVUL 0,0
51
STA 0,PRNSW ;SET PRINT SWITCH
52
EJSR PHBYT ;PRINT STRING DATA
53
EJSR MAPDTA ;YES,PRINT MAP IF MAPPED
54 13415 106470
55 077634
56 13417 002401 JMP
57 13420 000000 PWRLEX: 0

```

```

0244  FSPL
01
02
03
04
05
06
07
08 13421 054416 STK0: STA 3,STKR
09 13422 176070 ELEF 3,STAKB,0
10 001010
11 13424 000403 JMP STK2
12
13 13425 054412 STK1: STA 3,STKR
14 13426 034127 LDA 3,STKB ;
15
16 13427 054040 STK2: STA 3,SP ;SP = C(STKB)
17 13430 054041 STA 3,FP ;FP = C(STKB)
18 13431 177770 ADDI 20,*X6,3 ; STACK AREA
19
20 13433 054042 STA 3,SL ;SL = C(STKB) + 20,*X6
21 13434 034130 LDA 3,STKF
22 13435 054043 STA 3,SF ;SF = C(STKF)
23 13436 002401 JMP @STKR ;EXIT
24
25 13437 000000 STKR: 0

```

```

0244  FSPL
01
02
03
04
05
06
07
08 13421 054416 STK0: STA 3,STKR
09 13422 176070 ELEF 3,STAKB,0
10 001010
11 13424 000403 JMP STK2
12
13 13425 054412 STK1: STA 3,STKR
14 13426 034127 LDA 3,STKB ;
15
16 13427 054040 STK2: STA 3,SP ;SP = C(STKB)
17 13430 054041 STA 3,FP ;FP = C(STKB)
18 13431 177770 ADDI 20,*X6,3 ; STACK AREA
19
20 13433 054042 STA 3,SL ;SL = C(STKB) + 20,*X6
21 13434 034130 LDA 3,STKF
22 13435 054043 STA 3,SF ;SF = C(STKF)
23 13436 002401 JMP @STKR ;EXIT
24
25 13437 000000 STKR: 0

```



```

0245 ESPCL      :STACK FAULT ROUTINE
01
02
03 13440 000000
04 13441 000000
05 13442 040777 STAKF:
06 13443 020042 STA 0,-1
07 13444 040774 LDA 0,SL
08 13445 163770 ADUI 20,-0
09
10 13447 040042 STA 0,SL
11
12 13450 124530 CRLFPR
13 XOP 1,1,S      :*OUTPUT CR/LF
14 PRINT MS2
15 XOP 3,3,4    :*PRINT MESSAGE MS2
16 MS2
17 LDA 0,SP
18 ANDI 77360,0
19
20 MOVZ# 0,0,SZR
21 JMP +3
22 PRINT MS2X
23 XOP 3,3,4    :*PRINT MESSAGE MS2X
24 MS2X
25 CRLFPR
26 XOP 1,1,5    :*OUTPUT CR/LF
27 PRINT MS6
28 XOP 3,3,4    :*PRINT MESSAGE MS6
29 MS6
30 LDA 1,SP
31 OCTZ
32 XOP 1,1,1    :*PRINT (AC1) OCTAL ZERO(S) SUPPRESS
33 LDA 1,FP
34 OCTZ
35 XOP 1,1,1    :*PRINT (AC1) OCTAL ZERO(S) SUPPRESS
36 LDA 1,STAKF-2
37 OCTZ
38 XOP 1,1,1    :*PRINT (AC1) OCTAL ZERO(S) SUPPRESS
39 LDA 1,8F
40 OCTZ
41 XOP 1,1,1    :*PRINT (AC1) OCTAL ZERO(S) SUPPRESS
42 LDA 1,TESTN **
43 OCTZ
44 XOP 1,1,1    :*PRINT (AC1) OCTAL ZERO(S) SUPPRESS
45 LDA 0,STAKF-1
46 PRMRLO
47 EJSR
48 PWRLO
49
50 13500 106470
51 077654
52 13502 077077 HAF042: HALTA 3
53 077654
54 13503 000777 JMP -1
55
56 ***** FATAL ERROR HALT *****
57 **FATAL STACK FAULT NO GO HALT
58 **SUGGEST YOU RUN BASIC DIAGNOSTICS
59

```

```

0246 ESPCL
01
02
03
04 13504 177760
05 13505 000000
06 13506 054777 COBIT:
07 13507 034775 STA 3,-1
08 13510 101122 MOVZL
09 13511 125400 INC 1,1
10 13512 175404 TNC 3,3,SZR
11 13513 040775 JMP -3
12 13514 062711 JMP
13 13515 000000
14 13516 054777 IUGEN: STA 3,-1
15 13517 022235 LDA 0,BISWREG:**
16 13520 143770 ANDI 40001,0 :PROCEED SWT <1> OR <1S>
17
18 13522 101015 MOVZ# 0,0,SNR :TEST = 0
19 13523 01400 JMP 0,5 :EXIT
20 13524 024203 LDA 1,PASS :TEST
21 13525 125014 MOVZ# 1,1,SZR :NOT = 0
22 13526 014900 JMP 0,3 :EXIT
23 13527 132470 ELDA 2,MARK-1;
24
25 13531 024375 LDA 1,10 :PROGRAM ERROR ID WORD
26 13532 021000 LDA 0,0,2
27 13533 047753 JSK COBIT
28 13534 021001 LDA 0,1,2
29 13535 047751 JSK COBIT
30 13536 021002 LDA 0,2,2
31 13537 047747 JSK COBIT
32 13540 031003 LDA 0,3,2
33 13541 047745 JSK COBIT
34 13542 021004 LDA 0,4,2
35 13543 101120 MOVZL
36 13544 100510 XOR 0,0
37 13545 101100 MOVZL
38 13546 107020 ADDZ 0,1
39 13547 044375 STA 1,10 :UPDATE
40 13550 002745 JMP @IUGEN-1:RETURN

```

```

:ERROR COUNT BITS (COB) SUBROUTINE
-16.
0
COBIT:
MOVZL
INC
TNC
JMP
JMP
STA
LDA
ANDI
MOVZ#
JMP
LDA
LDA
MOVZ#
JMP
ELDA
LDA
LDA
JSK
LDA
JSK
LDA
JSK
LDA
LDA
MOVZL
XOR
MOVZL
ADDZ
STA
JMP

```

```

0247 ESPCL
01 13551 000000
02 13552 152470
03 13553 040234
04 000234
05 13554 152470
06 000233
07 13556 102111
08 13557 000402
09 13560 014211
10 13561 000401
11 13562 156434
12 13563 000413
13
14
15 13564 004732
16 13565 034210
17 13566 175014
18 13567 000555
19
20 13570 174430
21 13571 014707
22 13572 024203
23 13573 125800
24
25 13574 124330
26 13575 000427
27
28 13576 014211
29 13577 000401
30 13600 124470
31 000205
32 13602 125605
33 13603 000404
34 13604 116870
35 000201
36 13606 107710
37 13607 024576
38 13610 044576
39
40 13611 174430
41 13612 014674
42 13613 131000
43 13614 024374
44
45 13615 124130
46 13616 147000
47 13617 044374
48
49 13620 124530
50 13621 100510
51 13622 040210
52 13623 000403
53 13624 102020
54 13625 040210
55
56 13626 174430
57 13627 014724
58 13630 030557
59 13631 021000
60 13632 025001

:ERROR PROCESS CODE.
ERR: ESTA 2,MARK-1
ESTA 2,MARK
ISZ ITRC
JMP *+2
DSZ ITRC
JMP *+1
SURZ# 2,3,5ZR
TRACE
JMP

JSR IDGEN %COUNT BITS FOR I.D.
LDA 3,ITRER
MOV# 3,3,5ZR
JMP ERR1
PRINT ERPASS
XOP 3,3,4
:PRINT MESSAGE ERPASS
ERPASS
LDA 1,PASS
INC 1,1
DECP
XOP 1,1,3
:PRINT (AC1) DECIMAL ZERO(S) SUPPRESSED
JMP ERR0
DSZ ITRC
ELOA 1,MARK-2
MOV 1,1,5NR
JMP *+4
EUSZ MARK-2
POP8
LDA 1,TONST
STA 1,MARK-2
PRINT TRACX
XOP 3,3,4
:PRINT MESSAGE TRACK
MOV 1,2
LDA 1,NSCAN
OCTZ
XOP 1,1,1
:PRINT (AC1) OCTAL ZERO(S) SUPPRESS
ADD 2,1
STA 1,NSCAN
CRLFPR
XOP 1,1,5
:OUTPUT CR/LF
XOR 0,0
STA 0,ITRER
JMP ERR0+2
AOCZ 0,0
STA 0,ITRER
PRINT HEADER
XOP 3,3,4
:PRINT MESSAGE HEADER
HEADER
LDA 2,MARK-1
LDA 0,0,2
LDA 1,1,2

```

```

0248 ESPCL
01 13633 035003
LDA 3,3,2
LDA 2,2,2
STA 0,AC0
STA 1,AC1
STA 2,AC2
STA 3,AC3
07
08
09 13641 000401
10 13642 000401
11 13643 000401
12 13644 042543
13 13645 110710
14 13646 030541
15 13647 045001
16 13650 041002
17 13651 055003
18 13652 024216
19
20 13653 124130
21 13654 025004
22 13655 125120
23 13656 124510
24 13657 125100
25 13660 044233
26
27 13661 124330
28 13662 025000
29
30 13663 124230
31 13664 025001
32
33 13665 124230
34 13666 025002
35
36 13667 124230
37 13670 025003
38
39 13671 124230
40 13672 025004
41 13673 147770
42 077777
43 13675 046654
44 13676 020214
45 13677 106000
46 13700 125132
47 13701 024650
48
49 13702 124130
50 13703 024646
51 13704 104110
52
53 13705 124130
54 13706 106470
55 060114
56
57 13710 124530
58 13711 020225
59 13712 143770
60 000017

:INSERT USER DEBUG CODE HERE:
*+1 :USER
*+1 :INSTRUCTIONS
*+1 :GO HERE.
*+1 :MARK-1
*+1 :MARK-1
*+1 :MARK-1
*+1 :TESTN **
OCTZ
XOP 1,1,1
:PRINT (AC1) OCTAL ZERO(S) SUPPRESS
XOP 1,1,2
LDA 1,4,2
MOVZL
XOR 1,1
MOV 1,1
STA 1,CRY
DECP
XOP 1,1,3
:PRINT (AC1) DECIMAL ZERO(S) SUPPRESSED
LDA 1,0,2
OCTP
XOP 1,1,2
:PRINT (AC1) OCTAL
LDA 1,1,2
OCTP
XOP 1,1,2
:PRINT (AC1) OCTAL
LDA 1,2,2
OCTP
XOP 1,1,2
:PRINT (AC1) OCTAL
LDA 1,3,2
OCTP
XOP 1,1,2
:PRINT (AC1) OCTAL
LDA 1,4,2
AND1 77777,1
STA 1,ERR-1
LDA 0,RELOC
ADC 0,1
MOVZL#
LDA 1,1,5ZC
:LEGAL ?
LDA 1,ERR-1
:*(NO) USE LOGICAL
OCTZ
XOP 1,1,1
:PRINT (AC1) OCTAL ZERO(S) SUPPRESS
LDA 1,ERR-1
SRI 1,1
OCTZ
XOP 1,1,1
:PRINT (AC1) OCTAL ZERO(S) SUPPRESS
EJSR PHYP
CRLFPR
XOP 1,1,5
:OUTPUT CR/LF
LDA 0,PRNSW
AND1 17,0

```





```

0253 ESPCL
01 14223 104710
02 14224 143770
03 14224 040001
04 14226 101015
05 14227 009414
06 14230 024375
07 14231 125005
08 14232 000410
09
10 14233 174430
11 14234 014765
12
13 14235 174430
14 14236 014662
15
16 14237 124230
17 14240 126000
18 14241 044421
19
20 14242 124530
21 14243 034165
22 14244 021400
23 14245 101005
24 14246 000410
25 14247 015403
26 14250 000406
27 14251 062677
28 14252 021403
29 14253 035404
30 14254 041776
31 14255 001400
32
33 14256 100510
34 14257 040225
35
36 14260 006061
37 14261 012355
38 14262 000000

```

XCH 0,1 ;  
ANDI 40001,0 ;SWT (1) AND SWT (15) = 1  
MOV# 0,0,SNR ;  
JMP PSCX2 ;BYPASS  
LDA 1,1 ;\*  
MOV 1,1,SNR ;\*  
JMP PSCXK ;\*  
PRINT PROG ;OUTPUT  
XOP 3,3,4 ;\*PRINT MESSAGE PROG  
PRG HIO ;DISPLAY  
XOP 3,3,0 ;\*PRINT MESSAGE HIO  
HIO  
OCTP ;I/O WORD  
XOP 1,1,2 ;\*\*PRINT (AC1) OCTAL  
ADC 1,1 ;  
STA 1,FRRIO ;  
PSCXK: XOP 1,1,5 ;\*OUTPUT CR/LF  
LDA 0,0,3 ;  
LDA 0,0,3 ;  
MOV 0,0,SNR  
JMP \*+8.  
DSZ 3,3  
JMP \*+6  
IORST  
LDA 0,3,3  
LDA 3,4,3  
STA 0,-2,3  
JMP 0,3  
;END OF PROGRAM TEST CYCLE.  
DUMOR: XOP 0,0 ;\*  
STA 0,PRNSW ;\*  
CALL REL ;\*CALL RELOCATION  
JSR @ICAL ;CALL SUBROUTINE  
REL ;REL  
;SUPPRESS = -1  
0

```

0254 ESPCL
01
02
03
04 14263 013552 ;ERROR DISPLAY CALL
05 14264 012157 ;PRINT OCTAL ZERO(S) SUPPRESS
06 14265 012171 ;PRINT OCTAL
07 14266 012161 ;PRINT DECIMAL
08 14267 012233 ;PRINT ASCII TEXT
09 14270 012236 ;PRINT ASCII CR/LF
10 14271 013732 ;FATAL ERROR MONITOR MODE CALL
11 14272 000164 FALSE
12 14273 000164 FALSE
13 14274 000164 FALSE
14 14275 000164 FALSE
15 14276 000164 FALSE
16 14277 000164 FALSE
17 14300 000164 FALSE
18 14301 000164 FALSE
19 14302 000164 FALSE
20 14303 000164 FALSE
21 14304 000164 FALSE
22 14305 000164 FALSE
23 14306 000164 FALSE
24 14307 000164 FALSE
25 14310 000164 FALSE
26 14311 000164 FALSE
27 14312 000164 FALSE
28 14313 000164 FALSE
29 14314 000164 FALSE
30 14315 000164 FALSE
31 14316 000164 FALSE
32 14317 000164 FALSE
33 14320 000164 FALSE
34 14321 000164 FALSE
35 14322 000164 FALSE
36 14323 000164 FALSE

```

0255 ESPCL

```
01 14324 000566 : MAPSD : MMPU/MMPUI MAP STATE DUMP UTILITY
02 : MAPSD : MAP STATE UTILITY DUMP MMPU/MMPUI
03 : STAND ALONE ENTRY
04 JMP DMM7
05 ENTRY DMM7
06 14325 000000 : 1-10
07 14326 000000 : 1-7
08 14327 000000 : 1-6
09 14330 000000 : 1-5
10 14331 000000 : 1-4
11 14332 000400 : 400
12 14333 015333 : SWREG 1-2
13 14334 001600 : 1600 1-1
14 14335 117110 : DMM0: PSH 0,3
15 PRINT HOIC :DIA:
16 14336 174430 : XOP 3,3,4 :PRINT MESSAGE HOIC
17 14337 014607 : HOIC
18 14340 066403 : DIC 1,MMPU :RESULTS
19 OCTP
20 14341 124230 : XOP 1,1,2 **:PRINT (AC1) OCTAL
21 PRINT HOIA :DIA:
22 14342 174430 : XOP 3,3,4 :PRINT MESSAGE HOIA
23 14343 014614 : HOIA
24 14344 064403 : DIA 1,MMPU :RESULTS
25 OCTP
26 14345 124230 : XOP 1,1,2 **:PRINT (AC1) OCTAL
27 CRLFPR ICRLF **:OUTPUT CR/LF
28 14346 124530 : XOP 1,1,5
29 READS 0
30 14350 101113 : MOVL# 0,0,SNC
31 14351 022762 : LDA 0,DMMO-2:SWREG
32 14352 143770 : ANDI 200,0 :SMT 8 = 0
33 14353 002500 : MOV# 0,0,SNR
34 14354 101615 : JMP DMM7
35 14355 005343 : LDA 0,DMMO-1:PMT BITS (6-8)
36 14356 020756 : DOC 0,MMPU :LATE BLK
37 14357 063003 : DIA 1,MMPU :RESULTS
38 14360 064403 : AND 0,1 :MASK
39 14361 107400 : SUB# 0,1,SZR :RESULTS?
40 14362 106414 : JMP **:3 :MMPU
41 14363 000403 : :BYPASS
42 SUBZL 3,3 :MMPUI = 1
43 14364 176520 : JMP **:2
44 14365 000402 : XOP 3,5 :MMPUI = 0
45 14366 174510 : STA 3,DMMO-5 :MMPF RECORD
46 14367 054741 :
47
48 14370 020742 : LDA 0,DMMO-3
49 14371 175004 : MOV 3,3,SZR :TEST
50 14372 100510 : XOR 0,0 :MMPUI = 0
51 14373 040736 : STA 0,DMMO-4 :MMPUI = 400
52 14374 022737 : LDA 2,DMMO-2 :SWREG (USER)
53 14375 050732 : STA 2,DMMO-6 :RECORD
54 14376 100510 : XOR 0,0 :ICLEAR
55 14377 042734 : STA 0,DMMO-2 :SWREG
56 14400 034730 : LDA 3,DMMO-5 :MMPUI = 0 MMPUI = 1
57 14401 162470 : ELEF 0,DMMA :HEADER
58 14402 000203 :
59 14403 117000 : ADD 0,3
60 14404 021400 : LDA 0,0,3 :FETCH
```

0256 ESPCL

```
01 14405 040402 : STA 0,0,2 :HEADER
02 : SETUP
03 PRINT 0 :HEADER
04 XOP 3,3,4 **:PRINT MESSAGE 0
05 0
06 14406 174430 : LDA 0,DMMO-4 :MMPUI
07 14407 000000 : STA 0,DMMO-10 :400 = MMPUI
08 14410 020721 : XOR 0,0
09 14411 040714 : STA 0,DMMO-7 :RECORD
10 14413 100510 : LDA 1,DMMO-7
11 14414 020712 : MOVZS 1,1
12 14415 125520 : MOVZR 1,1
13 14417 125520 : OCTP
14
15 14420 124230 : XOP 1,1,2 **:PRINT (AC1) OCTAL
16 14421 024705 : LDA 1,DMMO-7
17 14422 030703 : LDA 2,DMMO-10
18 14423 133000 : ADD 1,2
19 14424 073003 : DOC 3,MMPU :LATE BLK
20 14425 076403 : DIC 2,MMPU :RESULTS
21 14426 157770 : ANDI 10177,3 :MMPUI
22 14430 020700 : LDA 0,DMMO-5 :MMPF
23 14431 101004 : MOV 0,0,SZR :TEST
24 14432 000403 : JMP **:3 :MMPUI BYPASS
25 14433 157770 : ANDI 777,3 :MMPUI
26
27 14435 164710 : XCH 3,1
28 14436 117110 : PSH 0,3
29 OCTP
30 14437 124230 : XOP 1,1,2 **:PRINT (AC1) OCTAL
31 14440 163210 : POP 3,0
32 14441 024664 : LDA 1,DMMO-10
33 14442 150510 : XOR 2,2
34 14443 113770 : IORI 1400,2 :MMPUI
35 14445 101004 : MOV 0,0,SZR
36 14446 030666 : LDA 2,DMMO-1 :MMPUI = 1600
37 14447 132414 : SUB# 1,2,SZR
38 14448 000417 : JMP DMM8
39 14450 000417 : :MMPUI
40 14451 020660 : LDA 0,DMMO-4 :400 0
41 14452 040653 : STA 0,DMMO-10
42 14453 124530 : CRLFPR ICRLF
43 14454 024652 : XOP 1,1,5 :OUTPUT CR/LF
44 14455 172070 : LDA 1,DMMO-7
45 14456 076000 : ELEF 2,76000,0
46 14457 132415 : SUB# 1,2,SNR :TEST
47 14460 000434 : JMP DMM8 :DONE
48 14461 024645 : LDA 1,DMMO-7
49 14462 172070 : ELEF 2,2,DMMO-6
50 14463 002000 : ADD 2,1 :ADVANCE LOGICAL
51 14464 147000 : STA 1,DMMO-7 :RECORD
52 14465 046461 : JMP DMM2
53 14466 000726 : LDA 2,DMMO-5 :MMPF
54 14467 030641 : MOV 2,2,SNR :TEST
55 14470 151005 : JMP DMM6 :?????
56 14471 000415 : ELEF 2,1400,0 :MMPUI HERE
57 14472 172070 :
58
59
60
```



```

0359 ESPCL
01 14620 005215 DMW: .TXTE !<15><12>MAP DATA<15><12>
02 040515
03 120120
04 040504
05 040724
06 005215
07 14626 040714 LAST MPMU DGA WORD: I
08 152123
09 046640
10 050115
11 120125
12 147504
13 120101
14 147727
15 042322
16 120072
17 000000
18 14641 005215 OMNY: .TXTE !<15><12>MAP DATA<15><12>
19 040515
20 120120
21 040504
22 040724
23 005215
24 14647 040714 LAST MPMU LNP WORD: I
25 152123
26 046640
27 050115
28 130525
29 146240
30 050115
31 153640
32 151317
33 035104
34 000240
35 14662 005215 HID: .TXTE !<15><12><15>ERROR CODE ID = I
36 142615
37 151322
38 151317
39 141640
40 042317
41 120305
42 042311
43 136640
44 000240

```

```

0260 FSPCL
01
02
03 14674 005215 TRAX: .TXTE !<15><12>TRACE:# I
04 151324
05 141501
06 035305
07 120243
08 000000
09 14702 005215 PASMES: .TXTE !<15><12>PASS I
10 040520
11 051523
12 120240
13 000000
14 14707 005215 ERPASS: .TXTE !<15><12>ERR IN PASS I
15 151305
16 120322
17 047311
18 050240
19 051501
20 120123
21 008240
22 14717 005215 PERCEV: .TXTE !<15><12>% FAIL=I
23 120245
24 040706
25 146311
26 000275
27 14724 005215 HEADER: .TXTE !<15><12><15>
28 14724 152215 TEST# CRY AC0 AC1 AC2 AC3 LISTING LOGICAL
29 051705
30 121724
31 141411
32 054722
33 004640
34 040640
35 030303
36 004640
37 040640
38 130703
39 004640
40 040640
41 131303
42 004640
43 040640
44 031703
45 146011
46 051711
47 140724
48 043516
49 146240
50 043717
51 141711
52 146101
53 14756 050240 PHYSICAL<15><12><15>I
54 054510
55 144523
56 040703
57 106714
58 106412
59 000000
60 14765 005215 PRUG: .TXTE !<15><12><15>ESPCLX REV. 03 12/28/76I

```



```

0261 ESPCL
01 142615
02 050125
03 146303
04 154305
05 120240
06 120240
07 142722
08 027126
09 030240
10 004463
11 131261
12 131257
13 127670
14 033267
15 000000
16 15005 005215 MESIZ: .TXTE I<15><12>NO MPU<15><12>
17 147514
18 046640
19 050115
20 106525
21 15012 146012 LAST LGL ADDR=I
22 051501
23 120324
24 043714
25 120314
26 042101
27 151104
28 000275
29 15022 005215 KCRLF: .TXTE I<15><12>I
30 000000
31 15024 142523 SETSW: .TXTE !SET SWITCHES AND PRESS CONTINUEI
32 120324
33 153523
34 152311
35 044303
36 051705
37 040640
38 042116
39 050240
40 142722
41 051523
42 141640
43 047317
44 144724
45 052516
46 000305
47 15044 051305 MS1: .TXTE !ERR CORR.I
48 120322
49 147705
50 151322
51 000056
52 15051 042101 MS3: .TXTE !ADDR: I
53 151104
54 120072
55 000000
56 15055 147703 MS4: .TXTE !CODE: I
57 142504
58 120072
59 000000
60 15061 047311 MS7: .TXTE !INT: I

0262 ESPCL
01 035324
02 000240
03 15064 152123 MS2: .TXTE !STACK FAULTI
04 141501
05 120113
06 040706
07 146125
08 000324
09 15072 052442 MS2X: .TXTE !"UNDERFLOW"<15><12><15>I
10 042116
11 151305
12 146306
13 153717
14 106442
15 106412
16 000000
17 15102 005215 MS4: .TXTE I<15><12><15>
18 15103 051615 SP FP SL SF TEST#<15><12><15>I
19 004520
20 050306
21 051411
22 004714
23 143123
24 152011
25 051705
26 121724
27 005215
28 000215
29 15116 045661 MS8: .TXTE !1K MODULE: I
30 046640
31 042317
32 146125
33 035305
34 000240
35 15124 046515 MS9X: .TXTE !MPPI PRESENT<15><12>I
36 052520
37 120261
38 151120
39 051705
40 047305
41 106724
42 000012
43 15134 046515 MS9: .TXTE !MPPI PRESENT<15><12>I
44 052520
45 050240
46 142722
47 142523
48 152116
49 005215
50 000000
51 15144 147724 MS9Y: .TXTE !TOTAL # OF 1K'S = I
52 040724
53 120314
54 120243
55 143317
56 130640
57 023513
58 120123
59 130275
60 000000

```

```

0263 ESPCL
01 15156 040515 MS10: .TXTE !MAPPING (IN TERMS OF !K'S) <15><12>
02 050120
03 047311
04 120107
05 144450
06 120116
07 142724
08 046722
09 120123
10 143317
11 130640
12 023513
13 124523
14 106640
15 15174 004412
16 151306
17 046717
18 152011
19 106717
20 15201 050012 PHYSI. !
21 054510
22 144523
23 004456
24 000000
25 15206 005215 MS11: .TXTE !<15><12>
26 15207 147714 LOGICL 0 31<15><12>!
27 144507
28 144303
29 030011
30 031411
31 106661
32 000012
33 15216 154305 MS12: .TXTE !EXCEPT 0TOSIK = PHYSICAL !
34 142703
35 152120
36 042240
37 147724
38 130523
39 120113
40 120275
41 044120
42 051531
43 141711
44 146101
45 000240
46 15233 17777 MSPFL: .TXTE !<177><177><177><15><12>POWER FAIL INTERRUPT!
47 106777
48 050012
49 153717
50 151305
51 144240
52 144501
53 120314
54 047311
55 142724
56 151322
57 050125
58 000524
59 15250 005215 ABORT: .TXTE !<15><12><15>
60 004615

```

```

0264 ESPCL
01 050011
02 147722
03 151107
04 046501
05 040640
06 147502
07 152322
08 047311
09 106507
10 15243 120012
11 147714
12 053303
13 147640
14 120306
15 146011
16 043717
17 141711
18 146101
19 050011
20 054510
21 144523
22 040703
23 120314
24 142724
25 152123
26 106643
27 15304 120012
28 040706
29 040724
30 120314
31 141520
32 136640
33 000011
34 15313 005215 SWIS: .TXTE !<15><12><15>SMT'S SELECTED = !
35 051615
36 152327
37 051447
38 051640
39 146305
40 141705
41 142724
42 120104
43 120275
44 000000

```

LOGICAL PHYSICAL TEST#<15><12>

LUC: OF

FATAL PC = :

PROGRAM ABORTING<15><12>

0265 ESPCL  
 01 15326 00000 EGGS: 0  
 02 15327 00000 DEVI: 0  
 03 15328 00000 CATSW: 0  
 04 15330 00000 PRUM: 0  
 05 15331 00000 RTRN: 0  
 06 15332 00000 SHREG: 0  
 07 15333 00000 DIRT: 0  
 08 15334 051705 DIRT: .TXTE /ESPCLEX 03/ ;REV. 03  
 09 15335 141520  
 10 142714  
 11 120330  
 12 030240  
 13 000063  
 14 15342 000000  
 15 15343 00200  
 16 15344 175772  
 17 15345 000000  
 18 15346 000000  
 19 15347 000000  
 20 15350 000000  
 21 15351 000000  
 22 15352 015352 PRCEND: .  
 23  
 24  
 25  
 26  
 27  
 28  
 29  
 30  
 31  
 32  
 33  
 34  
 35  
 36  
 37  
 38  
 39  
 40  
 41  
 42  
 43  
 44  
 45  
 46  
 47  
 48  
 49  
 50  
 51  
 52  
 53  
 54  
 55  
 56  
 57  
 58  
 59  
 60

01 AUTO SWITCH NON ZERO FOR DTOS  
 02 DEVICE CODE PASSED FROM DTOS  
 03 CAT SWITCH NON ZERO IF CAT LOADED  
 04 PASS COUNT DETOS RUNS \*N\* PASSES  
 05 RETURN LOCATION (MEMTOP-1)  
 06 DTOS SWITCH REGISTER ENTRY  
 07  
 08  
 09  
 10  
 11  
 12  
 13  
 14  
 15  
 16  
 17  
 18  
 19  
 20  
 21  
 22  
 23  
 24  
 25  
 26  
 27  
 28  
 29  
 30  
 31  
 32  
 33  
 34  
 35  
 36  
 37  
 38  
 39  
 40  
 41  
 42  
 43  
 44  
 45  
 46  
 47  
 48  
 49  
 50  
 51  
 52  
 53  
 54  
 55  
 56  
 57  
 58  
 59  
 60

0266 ESPCL  
 01 15666 015666 FINIX: .  
 02  
 03

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

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

0265 ESPCL  
 01 15326 00000 EGGS: 0  
 02 15327 00000 DEVI: 0  
 03 15328 00000 CATSW: 0  
 04 15330 00000 PRUM: 0  
 05 15331 00000 RTRN: 0  
 06 15332 00000 SHREG: 0  
 07 15333 00000 DIRT: 0  
 08 15334 051705 DIRT: .TXTE /ESPCLEX 03/ ;REV. 03  
 09 15335 141520  
 10 142714  
 11 120330  
 12 030240  
 13 000063  
 14 15342 000000  
 15 15343 00200  
 16 15344 175772  
 17 15345 000000  
 18 15346 000000  
 19 15347 000000  
 20 15350 000000  
 21 15351 000000  
 22 15352 015352 PRCEND: .  
 23  
 24  
 25  
 26  
 27  
 28  
 29  
 30  
 31  
 32  
 33  
 34  
 35  
 36  
 37  
 38  
 39  
 40  
 41  
 42  
 43  
 44  
 45  
 46  
 47  
 48  
 49  
 50  
 51  
 52  
 53  
 54  
 55  
 56  
 57  
 58  
 59  
 60

0266 ESPCL  
 01 15666 015666 FINIX: .  
 02  
 03

01  
 02  
 03

START  
 .END

RELOCATABLE UPPER BUFFER AREA USED BY ESPCLX

FINI.: .TXTE /COPYRIGHT (C) OGC, 1975, 1976

ALL RIGHTS RESERVED/

\*LOCATION: XOP 0,3,6 ;\*FATAL ERROR CALL

THRU

\*LOCATION: XOP 0,3,6 ;\*FATAL ERROR CALL





0271 ESPCL

02010 007134 175/28 175/36  
 02020 007225 176/40 176/46  
 02020 007206 176/28 176/36  
 02030 007277 177/40 177/46  
 02030 007260 177/28 177/36  
 E001 001605 66/26  
 E002 001630 67/23  
 E003 001661 68/27  
 E004 001710 69/26  
 E005 001741 70/25  
 E006 002015 71/48  
 E007 002034 72/19  
 E010 002053 72/40  
 E011 002072 73/19  
 E012 002111 73/40  
 E013 002144 74/31  
 E014 002175 75/20  
 E015 002215 76/20  
 E016 002241 77/23  
 E017 002273 78/27  
 E020 002323 79/24  
 E021 002400 80/51  
 E022 002407 80/59  
 E023 002417 81/08  
 E024 002440 82/19  
 E025 002445 82/25  
 E026 002452 82/31  
 E027 002457 82/37  
 E030 002507 83/30  
 E031 002513 83/35  
 E032 002516 83/39  
 E033 002521 83/43  
 E034 002523 83/46  
 E035 002530 83/52  
 E036 002547 84/21  
 E037 002562 85/14  
 E040 002570 85/21  
 E041 002634 87/26  
 E042 002652 88/17  
 E043 002675 89/22  
 E044 002720 90/22  
 E045 002741 91/21  
 E046 002762 92/21  
 E047 003007 93/29  
 E050 003032 94/26  
 E051 003047 95/21  
 E052 003117 96/33  
 E053 003157 97/33  
 E054 003206 98/26  
 E055 003227 99/27  
 E056 003231 99/30  
 E057 003233 99/33  
 E060 003235 99/36  
 E061 003256 100/22  
 E062 003260 100/25  
 E063 003262 100/28  
 E064 003264 100/31  
 E065 003302 101/18

0272 ESPCL

E066 003327 102/25  
 E067 003350 103/21  
 E070 003371 104/21  
 E071 003424 105/36  
 E072 003467 106/47  
 E073 003534 107/45  
 E074 003602 108/47  
 E075 003650 109/46  
 E076 003672 110/29  
 E077 003674 110/32  
 E100 003676 110/35  
 E101 003700 110/38  
 E102 003703 110/42  
 E103 003724 111/23  
 E104 003747 112/25  
 E105 003773 113/27  
 E106 004017 114/27  
 E1064 000147 50/10  
 E107 004057 115/48  
 E110 004123 116/50  
 E111 004172 117/49  
 E112 004242 118/51  
 E113 004312 119/50  
 E114 004347 120/37  
 E115 004400 121/26  
 E116 004402 121/29  
 E117 004432 122/29  
 E120 004442 122/38  
 E121 004471 123/23  
 E122 004512 124/24  
 E123 004514 124/27  
 E124 004516 124/50  
 E125 004541 125/27  
 E126 004564 126/27  
 E127 004612 127/30  
 E130 004622 128/13  
 E131 004625 129/17  
 E132 004630 128/21  
 E133 004633 128/25  
 E134 004636 128/29  
 E135 004641 128/33  
 E136 004644 128/37  
 E137 004647 128/41  
 E140 004652 128/45  
 E141 004655 128/49  
 E142 004660 128/53  
 E143 004663 129/01  
 E144 004666 129/05  
 E145 004671 129/09  
 E146 004674 129/13  
 E147 004677 130/30  
 E150 004725 131/30  
 E151 004733 131/30  
 E152 005015 132/49  
 E153 005072 134/02  
 E154 005152 135/59  
 E155 005232 137/60  
 E156 005312 139/60

222/15

0273 ESPCL

E157 005327 141/23  
 E160 005331 141/26  
 E161 005333 141/29  
 E162 005335 141/32  
 E163 005331 142/16  
 E164 005374 143/27  
 E165 005422 144/29  
 E166 005450 145/29  
 E167 005476 146/29  
 E170 005500 147/48  
 E171 005615 149/02  
 E172 005675 150/39  
 E173 005755 152/60  
 E174 006035 154/60  
 E175 006060 156/30  
 E176 006062 156/33  
 E177 006064 156/36  
 E200 006111 157/28  
 E201 006101 158/30  
 E202 006171 159/31  
 E203 006221 160/31  
 E204 006244 161/24  
 E205 006250 161/29  
 E206 006254 161/34  
 E207 006314 162/41  
 E210 006363 163/52  
 E211 006435 164/53  
 E212 006507 165/54  
 E213 006561 166/54  
 E214 006604 167/50  
 E215 006606 167/53  
 E216 006610 167/56  
 E217 006635 168/28  
 E220 006665 169/50  
 E221 006715 170/31  
 E222 006745 171/31  
 E223 006770 172/24  
 E224 006774 172/29  
 E225 007001 172/35  
 E226 007041 173/40  
 E227 007110 174/52  
 E230 007162 175/53  
 E231 007234 176/54  
 E232 007306 177/54  
 E233 007355 178/55  
 E234 007357 180/52  
 E235 007427 180/52  
 E236 007431 180/55  
 E237 007501 181/52  
 E240 007561 182/57  
 E241 007642 184/58  
 E242 007723 186/57  
 E243 007777 188/52  
 E244 010056 189/56  
 E245 010140 191/59  
 E246 010222 193/58  
 E247 010321 196/18  
 E250 010423 198/19

0274 ESPCL

E251 010525 200/20  
 E252 010627 202/20  
 E253 010747 204/35  
 E254 011073 206/36  
 E255 011217 208/37  
 E256 011343 210/37  
 E257 011417 211/51  
 E260 011476 212/56  
 E261 011567 215/09  
 E262 011610 216/17  
 E263 011634 217/17  
 E264 011663 218/20  
 E265 011712 219/20  
 E266 012514 230/04  
 E267 012613 232/10  
 E270 013350 242/17  
 E4000 000146 50/09  
 EDSF1 020376 37/02 222/18  
 180/02 181/03 182/02 184/02 186/02  
 188/02 189/02 191/02 193/02 195/05 197/04 199/04  
 201/04 203/05 205/04 207/04 209/04 211/02 212/02  
 214/02  
 EDSF2 020712 MC 38/02 195/44 197/43 199/43 201/43 203/44 205/43  
 207/43 209/43  
 EOSP3 021153 MC 39/02 195/03 197/02 199/02 201/02  
 EOSP4 021160 MC 39/09 203/03 205/02 207/02 209/02  
 EGG8 015326 48/06 50/00 265/02  
 E16N1 012071 221/36 221/47  
 E18Z1 020060 MC 35/02 156/06 167/06  
 E18Z2 020131 MC 34/02 157/02 158/02 160/02 162/02 164/02 166/02 168/02 169/02  
 170/02 171/02  
 E18Z3 020250 MC 35/02 162/02 173/02  
 E18Z4 020410 MC 36/02 163/03 164/02 165/02 166/02 174/03 175/02  
 176/02 177/02  
 EJM1 017355 MC 28/02 125/02 126/02 143/02  
 EJM2 017415 MC 29/02 127/02 144/01  
 EJM3 017467 MC 30/02 130/02 131/02  
 EJM4 017546 MC 31/02 132/07 147/07 145/01 146/01  
 EJM5 017677 MC 32/02 133/07 135/02 137/02 139/02 148/07 150/02  
 152/02 154/02  
 ELD41 016633 MC 23/02 99/06 100/01  
 ELD42 016636 MC 23/07 108/02 109/01  
 ELEF1 016347 MC 19/02 87/10 88/01  
 ELEF2 016406 MC 20/02 89/01 90/01  
 ELEF3 016453 MC 21/02 99/07 100/02  
 ELEF4 016532 MC 22/02 96/02 97/02  
 EMZ 013134 54/38 238/12 238/14 238/19 238/21 238/22 238/23  
 238/43  
 EMZ1 013136 238/14 238/36  
 EMZ2 013164 238/17 238/25 238/39  
 ENTIN 000120 49/34 59/03 65/09 67/04 68/03 69/04 70/03  
 71/04 72/04 72/25 73/04 73/05 74/02 75/03  
 76/04 77/06 78/11 79/10 80/20 82/11 83/08  
 84/07 85/07 86/10 87/12 88/03 89/03 90/03  
 91/03 92/03 93/07 94/07 95/03 96/04 97/04  
 98/07 99/09 100/04 101/03 102/03 103/03 104/03  
 105/08 106/07 107/03 108/04 109/03 110/04 111/04  
 112/04 113/05 114/05 115/12 116/08 117/04 118/05  
 119/04 120/08 121/04 122/05 123/05 124/12 125/09

0276 ESPCL

0275 ESPCL

|        |        |        |        |        |        |        |        |        |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 126/09 | 127/09 | 128/08 | 130/09 | 131/09 | 132/14 | 133/14 | 118/30 | 119/07 | 119/11 | 119/18 | 119/29 | 120/14 | 122/08 |
| 135/08 | 137/09 | 147/09 | 148/14 | 142/07 | 132/09 | 149/08 | 123/08 | 124/15 | 125/13 | 126/13 | 127/13 | 130/13 | 131/13 |
| 145/08 | 146/08 | 149/14 | 150/09 | 150/09 | 152/09 | 154/09 | 132/25 | 133/18 | 133/25 | 133/25 | 133/38 | 135/13 | 135/20 |
| 156/09 | 157/05 | 158/05 | 159/05 | 160/05 | 162/09 | 162/05 | 135/33 | 137/13 | 137/20 | 137/33 | 139/13 | 139/20 | 139/33 |
| 163/06 | 164/05 | 165/05 | 166/05 | 167/05 | 168/05 | 169/05 | 141/15 | 143/13 | 144/12 | 145/12 | 146/12 | 147/18 | 147/25 |
| 170/05 | 171/05 | 172/04 | 173/05 | 174/06 | 175/05 | 176/05 | 149/18 | 148/25 | 148/38 | 150/13 | 150/20 | 150/33 | 152/13 |
| 177/05 | 178/13 | 180/09 | 181/10 | 182/09 | 184/09 | 186/09 | 152/20 | 152/25 | 154/13 | 154/20 | 154/33 | 156/12 | 156/22 |
| 188/09 | 189/09 | 191/09 | 193/09 | 195/12 | 197/11 | 199/11 | 157/08 | 157/12 | 158/08 | 158/12 | 159/08 | 159/12 | 160/22 |
| 201/11 | 203/12 | 205/11 | 207/11 | 209/11 | 211/09 | 212/09 | 160/12 | 161/08 | 162/08 | 162/12 | 163/09 | 163/09 | 163/13 |
| 214/09 | 216/04 | 217/04 | 218/04 | 219/04 | 219/04 | 219/04 | 163/22 | 163/33 | 164/08 | 164/12 | 164/21 | 164/30 | 165/08 |
| 49/12  | 49/33  | 49/49  | 65/09  | 66/28  | 67/04  | 67/25  | 165/12 | 165/21 | 166/08 | 166/12 | 166/21 | 166/30 | 166/30 |
| 68/03  | 68/29  | 72/04  | 72/25  | 70/03  | 70/05  | 71/04  | 167/13 | 167/22 | 168/08 | 168/12 | 169/08 | 169/12 | 170/08 |
| 71/50  | 72/04  | 74/02  | 74/33  | 75/03  | 75/04  | 75/21  | 170/12 | 171/08 | 171/12 | 172/08 | 173/08 | 173/12 | 173/21 |
| 73/25  | 73/42  | 77/06  | 77/25  | 78/29  | 79/10  | 79/26  | 174/09 | 174/13 | 174/22 | 174/31 | 175/08 | 175/12 | 175/21 |
| 76/22  | 77/06  | 77/25  | 78/11  | 78/29  | 79/10  | 79/26  | 175/30 | 176/08 | 176/12 | 176/20 | 177/08 | 177/12 | 177/21 |
| 80/24  | 81/10  | 82/11  | 82/40  | 83/08  | 83/54  | 84/07  | 177/21 | 177/30 | 178/21 | 178/28 | 178/39 | 178/47 | 180/17 |
| 84/24  | 85/07  | 85/23  | 86/10  | 86/34  | 87/12  | 87/28  | 180/24 | 180/35 | 180/44 | 181/18 | 181/25 | 181/36 | 182/17 |
| 88/03  | 88/19  | 89/03  | 89/24  | 90/03  | 90/24  | 91/03  | 182/24 | 182/35 | 184/17 | 184/24 | 184/35 | 186/17 | 186/24 |
| 91/23  | 92/03  | 92/23  | 93/07  | 93/31  | 94/07  | 94/28  | 186/35 | 188/17 | 188/24 | 188/35 | 189/17 | 189/24 | 189/35 |
| 95/03  | 95/23  | 96/04  | 96/35  | 97/04  | 97/35  | 98/07  | 191/17 | 191/24 | 191/35 | 193/17 | 193/24 | 193/35 | 195/20 |
| 97/03  | 97/23  | 98/04  | 98/35  | 99/04  | 99/35  | 99/07  | 195/27 | 195/38 | 195/60 | 197/19 | 197/26 | 197/37 | 197/59 |
| 98/28  | 99/09  | 99/38  | 100/04 | 100/33 | 101/03 | 101/20 | 199/19 | 199/26 | 199/37 | 199/59 | 201/19 | 201/26 | 201/37 |
| 102/03 | 102/27 | 103/05 | 103/25 | 104/03 | 104/23 | 105/08 | 201/59 | 203/20 | 203/38 | 203/60 | 204/14 | 205/19 | 205/37 |
| 105/38 | 106/07 | 106/49 | 107/03 | 107/47 | 108/04 | 108/49 | 205/26 | 205/37 | 205/59 | 206/13 | 207/19 | 207/26 | 207/37 |
| 109/03 | 109/48 | 110/09 | 110/44 | 111/04 | 111/25 | 112/04 | 207/59 | 208/13 | 209/19 | 209/26 | 209/37 | 209/59 | 210/13 |
| 112/27 | 113/05 | 113/29 | 114/05 | 114/29 | 115/12 | 115/50 | 211/17 | 211/24 | 211/35 | 212/17 | 212/24 | 212/35 | 214/17 |
| 116/08 | 116/52 | 117/04 | 117/51 | 118/05 | 118/53 | 119/04 | 52/03  | 52/37  | 49/32  | 49/44  | 49/45  | 52/57  | 54/40  |
| 119/52 | 120/08 | 120/41 | 121/04 | 121/31 | 122/05 | 122/40 | 243/48 | 243/50 | 244/01 | 244/03 | 250/16 | 265/21 | 265/22 |
| 123/05 | 123/25 | 124/12 | 124/32 | 125/09 | 125/29 | 130/32 | 63/12  | 63/36  | 63/49  | 64/24  | 248/50 | 254/04 |        |
| 126/29 | 127/09 | 127/32 | 128/08 | 129/15 | 130/09 | 130/32 | 247/21 | 248/14 | 248/47 | 248/50 | 254/04 |        |        |
| 131/09 | 131/32 | 132/11 | 132/51 | 133/14 | 134/04 | 135/09 | 247/03 | 248/43 | 248/73 |        |        |        |        |
| 136/01 | 137/09 | 138/02 | 139/09 | 140/02 | 141/12 | 141/34 | 247/26 | 247/52 | 247/53 |        |        |        |        |
| 142/07 | 142/18 | 143/09 | 143/29 | 144/08 | 144/31 | 145/08 | 249/02 | 250/14 | 249/31 |        |        |        |        |
| 145/31 | 146/08 | 146/31 | 147/14 | 147/50 | 148/14 | 149/04 | 249/59 | 249/40 | 249/54 |        |        |        |        |
| 150/09 | 151/01 | 152/09 | 153/02 | 154/09 | 155/02 | 156/09 | 249/59 | 250/03 | 250/04 |        |        |        |        |
| 156/38 | 157/05 | 157/30 | 158/05 | 158/32 | 159/05 | 159/33 | 2/01   | 2/01   | 2/01   |        |        |        |        |
| 160/05 | 160/33 | 161/04 | 161/36 | 162/05 | 162/43 | 163/06 | 51/31  | 66/25  | 61/22  | 68/26  | 69/25  | 70/42  | 71/47  |
| 163/54 | 164/05 | 164/55 | 165/05 | 165/56 | 166/05 | 166/56 | 72/18  | 72/39  | 73/18  | 73/39  | 74/30  | 75/19  | 76/19  |
| 167/09 | 167/38 | 168/05 | 168/30 | 169/05 | 169/32 | 170/05 | 77/22  | 78/26  | 79/23  | 80/50  | 80/56  | 81/07  | 82/18  |
| 170/33 | 171/05 | 171/33 | 172/04 | 172/37 | 173/05 | 173/42 | 82/34  | 82/30  | 82/36  | 83/29  | 83/34  | 83/38  | 83/42  |
| 174/06 | 174/54 | 175/05 | 176/55 | 176/55 | 176/56 | 177/05 | 83/45  | 83/51  | 84/20  | 85/13  | 85/20  | 87/25  | 88/16  |
| 177/56 | 178/13 | 178/60 | 180/09 | 180/57 | 181/10 | 181/56 | 89/21  | 90/21  | 91/20  | 92/20  | 93/28  | 94/25  | 95/20  |
| 182/09 | 183/01 | 184/09 | 185/03 | 186/09 | 187/02 | 188/09 | 96/32  | 97/32  | 98/25  | 99/26  | 99/38  | 99/35  | 99/35  |
| 188/56 | 189/09 | 189/60 | 191/09 | 192/04 | 193/09 | 194/03 | 100/21 | 100/24 | 100/27 | 100/30 | 101/17 | 102/24 | 103/20 |
| 195/12 | 196/23 | 197/11 | 198/24 | 199/11 | 200/25 | 201/11 | 104/20 | 105/35 | 106/46 | 107/44 | 108/46 | 109/45 | 110/28 |
| 202/25 | 203/12 | 204/41 | 205/11 | 206/42 | 207/11 | 208/43 | 110/31 | 110/34 | 110/37 | 110/41 | 111/22 | 112/24 | 113/26 |
| 209/11 | 210/43 | 211/09 | 211/55 | 212/09 | 212/60 | 214/09 | 114/26 | 115/47 | 116/49 | 117/48 | 118/50 | 119/49 | 120/36 |
| 215/14 | 216/04 | 216/19 | 217/04 | 217/04 | 218/60 | 218/72 | 121/25 | 121/28 | 122/28 | 122/37 | 123/22 | 124/23 | 124/26 |
| 219/04 | 219/22 | 235/24 | 71/07  | 72/07  | 72/28  | 73/07  | 124/29 | 125/26 | 126/26 | 127/29 | 128/12 | 128/16 | 128/20 |
| 49/36  | 65/12  | 67/07  | 71/07  | 72/07  | 72/28  | 73/07  | 128/24 | 128/28 | 128/36 | 128/44 | 128/44 | 128/44 | 128/48 |
| 73/28  | 75/06  | 76/07  | 77/09  | 78/14  | 79/13  | 80/23  | 128/52 | 128/56 | 128/60 | 129/12 | 129/12 | 130/29 | 130/29 |
| 87/15  | 88/06  | 89/06  | 91/06  | 91/06  | 91/09  | 92/06  | 131/29 | 132/48 | 134/01 | 135/58 | 137/59 | 139/59 | 141/22 |
| 92/09  | 93/11  | 94/10  | 95/06  | 96/07  | 97/07  | 99/13  | 141/25 | 143/28 | 141/31 | 142/15 | 143/26 | 144/28 | 145/28 |
| 100/08 | 101/07 | 102/07 | 103/07 | 104/07 | 105/17 | 106/11 |        |        |        |        |        |        |        |
| 106/18 | 106/26 | 107/07 | 107/14 | 107/22 | 108/08 | 108/15 |        |        |        |        |        |        |        |
| 108/23 | 109/07 | 109/14 | 109/22 | 110/13 | 110/19 | 111/08 |        |        |        |        |        |        |        |
| 111/15 | 112/06 | 112/18 | 113/09 | 113/15 | 114/09 | 114/15 |        |        |        |        |        |        |        |
| 115/16 | 115/23 | 115/32 | 116/12 | 116/19 | 116/27 | 116/43 |        |        |        |        |        |        |        |
| 117/08 | 117/15 | 117/23 | 117/42 | 118/08 | 118/12 | 118/19 |        |        |        |        |        |        |        |

ENTRX 000371  
 ENVRT 000000  
 ERAOD 001456  
 ERPAS 014707  
 ERR0 013552  
 ERR1 013624  
 ERR2 014013  
 ERR3 013770  
 ERR4 014002  
 ERR5 013717  
 ERBB 001066  
 ERRET 000212  
 ERROR 000453  
 MC  
 MC

ENTRA 000121

ENTLO 000117



## 0277 ESPCL

|               |        |        |        |        |        |        |        |
|---------------|--------|--------|--------|--------|--------|--------|--------|
| ERRR 000373   | 146/28 | 147/47 | 149/01 | 150/58 | 152/59 | 154/59 | 156/29 |
| ESP01 002303  | 156/32 | 156/35 | 157/27 | 158/29 | 159/30 | 160/30 | 161/23 |
| ESP02 002430  | 161/28 | 161/33 | 162/40 | 163/51 | 164/52 | 165/53 | 166/53 |
| ESP03 002462  | 167/59 | 167/59 | 167/35 | 168/27 | 169/29 | 170/30 | 171/30 |
| ESP04 002532  | 172/23 | 172/28 | 172/34 | 173/39 | 174/51 | 175/52 | 176/53 |
| ESP05 002552  | 177/53 | 178/54 | 178/57 | 180/51 | 180/54 | 181/51 | 182/56 |
| ESP06 002572  | 184/57 | 186/56 | 188/51 | 189/55 | 191/58 | 193/57 | 196/17 |
| ESP2 002461   | 198/18 | 200/19 | 202/19 | 204/34 | 206/35 | 208/36 | 210/36 |
| ESP2X 002455  | 211/50 | 212/55 | 215/08 | 216/16 | 217/16 | 218/19 | 219/19 |
| ESP3 002511   | 230/03 | 232/09 | 242/16 | 260/02 |        |        |        |
| ESP5 002564   | 52/59  | 78/10  | 79/09  | 220/56 | 221/03 | 221/23 | 227/31 |
| ESP5A 002560  | 80/19  | 249/56 |        |        |        |        |        |
| ESP6 002614   | 82/01  |        |        |        |        |        |        |
| ESP6A 002610  | 84/01  |        |        |        |        |        |        |
| ESP7 002642   | 85/07  |        |        |        |        |        |        |
| ESP8 002652   | 86/05  |        |        |        |        |        |        |
| ESP9 002662   | 87/09  |        |        |        |        |        |        |
| ESP10 002672  | 88/11  |        |        |        |        |        |        |
| ESP11 002682  | 89/19  |        |        |        |        |        |        |
| ESP12 002692  | 90/19  |        |        |        |        |        |        |
| ESP13 002702  | 91/06  |        |        |        |        |        |        |
| ESP14 002712  | 92/06  |        |        |        |        |        |        |
| ESP15 002722  | 93/06  |        |        |        |        |        |        |
| ESP16 002732  | 94/06  |        |        |        |        |        |        |
| ESP17 002742  | 95/06  |        |        |        |        |        |        |
| ESP18 002752  | 96/06  |        |        |        |        |        |        |
| ESP19 002762  | 97/06  |        |        |        |        |        |        |
| ESP20 002772  | 98/06  |        |        |        |        |        |        |
| ESP21 002782  | 99/06  |        |        |        |        |        |        |
| ESP22 002792  | 100/06 |        |        |        |        |        |        |
| ESP23 002802  | 101/06 |        |        |        |        |        |        |
| ESP24 002812  | 102/06 |        |        |        |        |        |        |
| ESP25 002822  | 103/06 |        |        |        |        |        |        |
| ESP26 002832  | 104/06 |        |        |        |        |        |        |
| ESP27 002842  | 105/06 |        |        |        |        |        |        |
| ESP28 002852  | 106/06 |        |        |        |        |        |        |
| ESP29 002862  | 107/06 |        |        |        |        |        |        |
| ESP30 002872  | 108/06 |        |        |        |        |        |        |
| ESP31 002882  | 109/06 |        |        |        |        |        |        |
| ESP32 002892  | 110/06 |        |        |        |        |        |        |
| ESP33 002902  | 111/06 |        |        |        |        |        |        |
| ESP34 002912  | 112/06 |        |        |        |        |        |        |
| ESP35 002922  | 113/06 |        |        |        |        |        |        |
| ESP36 002932  | 114/06 |        |        |        |        |        |        |
| ESP37 002942  | 115/06 |        |        |        |        |        |        |
| ESP38 002952  | 116/06 |        |        |        |        |        |        |
| ESP39 002962  | 117/06 |        |        |        |        |        |        |
| ESP40 002972  | 118/06 |        |        |        |        |        |        |
| ESP41 002982  | 119/06 |        |        |        |        |        |        |
| ESP42 002992  | 120/06 |        |        |        |        |        |        |
| ESP43 003002  | 121/06 |        |        |        |        |        |        |
| ESP44 003012  | 122/06 |        |        |        |        |        |        |
| ESP45 003022  | 123/06 |        |        |        |        |        |        |
| ESP46 003032  | 124/06 |        |        |        |        |        |        |
| ESP47 003042  | 125/06 |        |        |        |        |        |        |
| ESP48 003052  | 126/06 |        |        |        |        |        |        |
| ESP49 003062  | 127/06 |        |        |        |        |        |        |
| ESP50 003072  | 128/06 |        |        |        |        |        |        |
| ESP51 003082  | 129/06 |        |        |        |        |        |        |
| ESP52 003092  | 130/06 |        |        |        |        |        |        |
| ESP53 003102  | 131/06 |        |        |        |        |        |        |
| ESP54 003112  | 132/06 |        |        |        |        |        |        |
| ESP55 003122  | 133/06 |        |        |        |        |        |        |
| ESP56 003132  | 134/06 |        |        |        |        |        |        |
| ESP57 003142  | 135/06 |        |        |        |        |        |        |
| ESP58 003152  | 136/06 |        |        |        |        |        |        |
| ESP59 003162  | 137/06 |        |        |        |        |        |        |
| ESP60 003172  | 138/06 |        |        |        |        |        |        |
| ESP61 003182  | 139/06 |        |        |        |        |        |        |
| ESP62 003192  | 140/06 |        |        |        |        |        |        |
| ESP63 003202  | 141/06 |        |        |        |        |        |        |
| ESP64 003212  | 142/06 |        |        |        |        |        |        |
| ESP65 003222  | 143/06 |        |        |        |        |        |        |
| ESP66 003232  | 144/06 |        |        |        |        |        |        |
| ESP67 003242  | 145/06 |        |        |        |        |        |        |
| ESP68 003252  | 146/06 |        |        |        |        |        |        |
| ESP69 003262  | 147/06 |        |        |        |        |        |        |
| ESP70 003272  | 148/06 |        |        |        |        |        |        |
| ESP71 003282  | 149/06 |        |        |        |        |        |        |
| ESP72 003292  | 150/06 |        |        |        |        |        |        |
| ESP73 003302  | 151/06 |        |        |        |        |        |        |
| ESP74 003312  | 152/06 |        |        |        |        |        |        |
| ESP75 003322  | 153/06 |        |        |        |        |        |        |
| ESP76 003332  | 154/06 |        |        |        |        |        |        |
| ESP77 003342  | 155/06 |        |        |        |        |        |        |
| ESP78 003352  | 156/06 |        |        |        |        |        |        |
| ESP79 003362  | 157/06 |        |        |        |        |        |        |
| ESP80 003372  | 158/06 |        |        |        |        |        |        |
| ESP81 003382  | 159/06 |        |        |        |        |        |        |
| ESP82 003392  | 160/06 |        |        |        |        |        |        |
| ESP83 003402  | 161/06 |        |        |        |        |        |        |
| ESP84 003412  | 162/06 |        |        |        |        |        |        |
| ESP85 003422  | 163/06 |        |        |        |        |        |        |
| ESP86 003432  | 164/06 |        |        |        |        |        |        |
| ESP87 003442  | 165/06 |        |        |        |        |        |        |
| ESP88 003452  | 166/06 |        |        |        |        |        |        |
| ESP89 003462  | 167/06 |        |        |        |        |        |        |
| ESP90 003472  | 168/06 |        |        |        |        |        |        |
| ESP91 003482  | 169/06 |        |        |        |        |        |        |
| ESP92 003492  | 170/06 |        |        |        |        |        |        |
| ESP93 003502  | 171/06 |        |        |        |        |        |        |
| ESP94 003512  | 172/06 |        |        |        |        |        |        |
| ESP95 003522  | 173/06 |        |        |        |        |        |        |
| ESP96 003532  | 174/06 |        |        |        |        |        |        |
| ESP97 003542  | 175/06 |        |        |        |        |        |        |
| ESP98 003552  | 176/06 |        |        |        |        |        |        |
| ESP99 003562  | 177/06 |        |        |        |        |        |        |
| ESP100 003572 | 178/06 |        |        |        |        |        |        |
| ESP101 003582 | 179/06 |        |        |        |        |        |        |
| ESP102 003592 | 180/06 |        |        |        |        |        |        |
| ESP103 003602 | 181/06 |        |        |        |        |        |        |
| ESP104 003612 | 182/06 |        |        |        |        |        |        |
| ESP105 003622 | 183/06 |        |        |        |        |        |        |
| ESP106 003632 | 184/06 |        |        |        |        |        |        |
| ESP107 003642 | 185/06 |        |        |        |        |        |        |
| ESP108 003652 | 186/06 |        |        |        |        |        |        |
| ESP109 003662 | 187/06 |        |        |        |        |        |        |
| ESP110 003672 | 188/06 |        |        |        |        |        |        |
| ESP111 003682 | 189/06 |        |        |        |        |        |        |
| ESP112 003692 | 190/06 |        |        |        |        |        |        |
| ESP113 003702 | 191/06 |        |        |        |        |        |        |
| ESP114 003712 | 192/06 |        |        |        |        |        |        |
| ESP115 003722 | 193/06 |        |        |        |        |        |        |
| ESP116 003732 | 194/06 |        |        |        |        |        |        |
| ESP117 003742 | 195/06 |        |        |        |        |        |        |
| ESP118 003752 | 196/06 |        |        |        |        |        |        |
| ESP119 003762 | 197/06 |        |        |        |        |        |        |
| ESP120 003772 | 198/06 |        |        |        |        |        |        |
| ESP121 003782 | 199/06 |        |        |        |        |        |        |
| ESP122 003792 | 200/06 |        |        |        |        |        |        |
| ESP123 003802 | 201/06 |        |        |        |        |        |        |
| ESP124 003812 | 202/06 |        |        |        |        |        |        |
| ESP125 003822 | 203/06 |        |        |        |        |        |        |
| ESP126 003832 | 204/06 |        |        |        |        |        |        |
| ESP127 003842 | 205/06 |        |        |        |        |        |        |
| ESP128 003852 | 206/06 |        |        |        |        |        |        |
| ESP129 003862 | 207/06 |        |        |        |        |        |        |
| ESP130 003872 | 208/06 |        |        |        |        |        |        |
| ESP131 003882 | 209/06 |        |        |        |        |        |        |
| ESP132 003892 | 210/06 |        |        |        |        |        |        |
| ESP133 003902 | 211/06 |        |        |        |        |        |        |
| ESP134 003912 | 212/06 |        |        |        |        |        |        |
| ESP135 003922 | 213/06 |        |        |        |        |        |        |
| ESP136 003932 | 214/06 |        |        |        |        |        |        |
| ESP137 003942 | 215/06 |        |        |        |        |        |        |
| ESP138 003952 | 216/06 |        |        |        |        |        |        |
| ESP139 003962 | 217/06 |        |        |        |        |        |        |
| ESP140 003972 | 218/06 |        |        |        |        |        |        |
| ESP141 003982 | 219/06 |        |        |        |        |        |        |
| ESP142 003992 | 220/06 |        |        |        |        |        |        |
| ESP143 004002 | 221/06 |        |        |        |        |        |        |
| ESP144 004012 | 222/06 |        |        |        |        |        |        |
| ESP145 004022 | 223/06 |        |        |        |        |        |        |
| ESP146 004032 | 224/06 |        |        |        |        |        |        |
| ESP147 004042 | 225/06 |        |        |        |        |        |        |
| ESP148 004052 | 226/06 |        |        |        |        |        |        |
| ESP149 004062 | 227/06 |        |        |        |        |        |        |
| ESP150 004072 | 228/06 |        |        |        |        |        |        |
| ESP151 004082 | 229/06 |        |        |        |        |        |        |
| ESP152 004092 | 230/06 |        |        |        |        |        |        |
| ESP153 004102 | 231/06 |        |        |        |        |        |        |
| ESP154 004112 | 232/06 |        |        |        |        |        |        |
| ESP155 004122 | 233/06 |        |        |        |        |        |        |
| ESP156 004132 | 234/06 |        |        |        |        |        |        |
| ESP157 004142 | 235/06 |        |        |        |        |        |        |
| ESP158 004152 | 236/06 |        |        |        |        |        |        |
| ESP159 004162 | 237/06 |        |        |        |        |        |        |
| ESP160 004172 | 238/06 |        |        |        |        |        |        |
| ESP161 004182 | 239/06 |        |        |        |        |        |        |
| ESP162 004192 | 240/06 |        |        |        |        |        |        |
| ESP163 004202 | 241/06 |        |        |        |        |        |        |
| ESP164 004212 | 242/06 |        |        |        |        |        |        |
| ESP165 004222 | 243/06 |        |        |        |        |        |        |
| ESP166 004232 | 244/06 |        |        |        |        |        |        |
| ESP167 004242 | 245/06 |        |        |        |        |        |        |
| ESP168 004252 | 246/06 |        |        |        |        |        |        |
| ESP169 004262 | 247/06 |        |        |        |        |        |        |
| ESP170 004272 | 248/06 |        |        |        |        |        |        |
| ESP171 004282 | 249/06 |        |        |        |        |        |        |
| ESP172 004292 | 250/06 |        |        |        |        |        |        |
| ESP173 004302 | 251/06 |        |        |        |        |        |        |
| ESP174 004312 | 252/06 |        |        |        |        |        |        |
| ESP175 004322 | 253/06 |        |        |        |        |        |        |
| ESP176 004332 | 254/06 |        |        |        |        |        |        |
| ESP177 004342 | 255/06 |        |        |        |        |        |        |
| ESP178 004352 | 256/06 |        |        |        |        |        |        |
| ESP179 004362 | 257/06 |        |        |        |        |        |        |
| ESP180 004372 | 258/06 |        |        |        |        |        |        |
| ESP181 004382 | 259/06 |        |        |        |        |        |        |
| ESP182 004392 | 260/06 |        |        |        |        |        |        |
| ESP183 004402 | 261/06 |        |        |        |        |        |        |
| ESP184 004412 | 262/06 |        |        |        |        |        |        |
| ESP185 004422 | 263/06 |        |        |        |        |        |        |
| ESP186 004432 | 264/06 |        |        |        |        |        |        |
| ESP187 004442 | 265/06 |        |        |        |        |        |        |
| ESP188 004452 | 266/06 |        |        |        |        |        |        |
| ESP189 004462 | 267/06 |        |        |        |        |        |        |
| ESP190 004472 | 268/06 |        |        |        |        |        |        |
| ESP191 004482 | 269/06 |        |        |        |        |        |        |
| ESP192 004492 | 270/06 |        |        |        |        |        |        |
| ESP193 004502 | 271/06 |        |        |        |        |        |        |
| ESP194 004512 | 272/06 |        |        |        |        |        |        |
| ESP195 004522 | 273/06 |        |        |        |        |        |        |
| ESP196 004532 | 274/06 |        |        |        |        |        |        |
| ESP197 004542 | 275/06 |        |        |        |        |        |        |
| ESP198 004552 | 276/06 |        |        |        |        |        |        |
| ESP199 004562 | 277/06 |        |        |        |        |        |        |
| ESP200 004572 | 278/06 |        |        |        |        |        |        |
| ESP201 004582 | 279/06 |        |        |        |        |        |        |
| ESP202 004592 | 280/06 |        |        |        |        |        |        |
| ESP203 004602 | 281/06 |        |        |        |        |        |        |
| ESP204 004612 | 282/06 |        |        |        |        |        |        |
| ESP205 004622 | 283/06 |        |        |        |        |        |        |
| ESP206 004632 | 284/06 |        |        |        |        |        |        |
| ESP207 0      |        |        |        |        |        |        |        |

0279 ESPCL

HEARE 014724  
 HERE 000170  
 HERIT 000171  
 HFILL 000126 MC  
 HIBFA 000736

243/09  
 51/03  
 49/41  
 2/01  
 109/20  
 117/21  
 133/23  
 145/10  
 152/31  
 163/29  
 169/10  
 176/19  
 49/48  
 108/25  
 116/22  
 118/22  
 131/16  
 139/16  
 150/23  
 162/24  
 166/24  
 175/24  
 182/12  
 198/02  
 206/16  
 49/25  
 249/49  
 50/14  
 238/39

247/57  
 260/27  
 55/26  
 112/06  
 119/27  
 135/11  
 147/16  
 154/11  
 164/19  
 170/10  
 176/28  
 62/33  
 109/17  
 117/25  
 117/56  
 132/21  
 144/15  
 150/36  
 163/25  
 166/33  
 176/24  
 186/12  
 200/02  
 208/02  
 49/39  
 253/14  
 221/33  
 240/19

107/20  
 113/07  
 125/11  
 137/11  
 148/10  
 150/11  
 161/06  
 165/19  
 174/20  
 177/28  
 106/21  
 111/11  
 112/11  
 119/32  
 135/23  
 147/28  
 158/15  
 164/24  
 164/33  
 173/24  
 177/33  
 189/12  
 204/15  
 210/16  
 265/28  
 77/10  
 108/17  
 115/25  
 120/30  
 132/10  
 143/27  
 154/15  
 164/23  
 169/14  
 170/14  
 182/25  
 188/25  
 199/54  
 201/13  
 205/27  
 209/38  
 234/21

108/21  
 116/17  
 131/11  
 139/11  
 144/10  
 150/18  
 162/19  
 166/28  
 175/19  
 108/18  
 113/12  
 125/16  
 137/16  
 148/41  
 160/15  
 164/33  
 165/24  
 174/25  
 177/33  
 196/03  
 205/14  
 214/56  
 238/13  
 77/30  
 108/25  
 117/10  
 125/15  
 135/35  
 148/40  
 160/14  
 165/23  
 174/23  
 186/11  
 189/25  
 197/13  
 197/27  
 203/14  
 205/38  
 210/10  
 205/53  
 206/10

71/04  
 76/04  
 79/07  
 82/16  
 82/32  
 88/03  
 96/04  
 95/03  
 102/03  
 103/03  
 110/09  
 69/04  
 74/02  
 78/35  
 82/16  
 82/28  
 87/12  
 94/07  
 101/05  
 107/03  
 68/03  
 73/25  
 78/11  
 80/20  
 82/32  
 86/10  
 86/07  
 92/03  
 99/09  
 100/04  
 107/03  
 67/04  
 73/25  
 78/08  
 80/20  
 82/32  
 209/13  
 214/46  
 197/53  
 210/10  
 214/47

0280 ESPCL

111/04  
 118/05  
 125/09  
 133/14  
 144/08  
 154/09  
 162/05  
 169/05  
 176/05  
 186/09  
 195/11  
 218/09  
 48/20  
 51/55  
 51/53  
 51/53  
 237/27  
 53/01  
 249/13  
 246/14  
 88/14  
 48/23  
 49/20  
 221/24  
 221/22  
 52/57  
 2/01  
 63/14  
 2/01  
 47/11  
 42/01  
 2/01  
 58/30  
 158/17  
 160/26  
 162/36  
 165/09  
 167/20  
 170/09  
 172/21  
 174/18  
 174/47  
 176/49  
 159/16  
 162/18  
 164/18  
 165/25  
 171/16  
 174/17  
 175/50  
 177/26  
 165/25  
 174/28  
 176/34  
 177/27  
 177/25  
 48/19  
 51/54

112/04  
 119/04  
 126/09  
 135/09  
 145/08  
 156/09  
 163/06  
 170/05  
 177/05  
 188/09  
 201/11  
 203/11  
 203/12  
 216/04  
 49/16  
 60/07  
 59/49  
 62/13  
 249/41  
 246/40  
 252/26  
 221/32  
 221/25  
 221/25  
 221/04  
 55/28  
 64/39  
 62/45  
 62/46  
 63/02  
 62/45  
 156/17  
 158/25  
 161/13  
 163/10  
 163/18  
 165/17  
 168/09  
 170/17  
 172/35  
 174/47  
 177/09  
 157/18  
 159/18  
 162/26  
 164/18  
 166/16  
 169/27  
 171/18  
 171/18  
 174/19  
 176/16  
 177/51  
 163/28  
 165/34  
 174/28  
 174/35  
 177/27  
 55/29  
 60/47  
 115/12  
 122/05  
 130/09  
 141/12  
 148/14  
 158/05  
 164/05  
 173/05  
 181/10  
 193/09  
 207/11  
 218/04  
 114/05  
 121/04  
 128/08  
 139/09  
 147/14  
 158/05  
 165/05  
 172/04  
 180/09  
 191/09  
 205/11  
 217/04  
 229/31  
 227/36  
 61/01  
 61/12  
 232/52  
 232/54  
 252/55  
 247/15  
 221/10  
 221/46  
 221/30  
 221/04  
 81/18  
 86/38  
 157/17  
 159/26  
 161/31  
 162/09  
 164/17  
 166/49  
 169/17  
 169/25  
 171/17  
 173/35  
 174/10  
 176/17  
 158/18  
 160/18  
 163/17  
 165/18  
 166/26  
 168/51  
 170/18  
 173/18  
 175/18  
 176/51  
 177/16  
 157/09  
 159/17  
 161/25  
 163/47  
 166/09  
 166/17  
 168/23  
 171/09  
 173/09  
 175/17  
 177/49  
 157/25  
 160/16  
 163/17  
 164/50  
 166/26  
 169/27  
 170/16  
 173/16  
 174/49  
 176/18  
 176/26  
 163/35  
 166/25  
 175/27  
 175/34  
 177/34  
 86/39  
 220/14  
 220/35  
 220/42  
 222/10

ICAT 000061  
 ICAT 000236  
 ICATS 000234  
 ID 000375  
 IOGEN 013516  
 IH 000001  
 IMAPI 000064  
 INIT 012014  
 INITY 012070  
 INITX 012050  
 INITZ 012016 MC  
 INSTK 000776  
 INTBA 001467 MC  
 INTRM 001005  
 INTRP 001355  
 INTRU 021237 MC  
 INTRX 000000  
 IR 000000  
 ISLCO 000264  
 ISLC1 000265  
 ISLC2 000266  
 ISTK1 000060  
 ISMRE 000235

HIBUF 015353  
 HID 014662  
 HIGIK 000152  
 HIGRF 000124

247/57  
 260/27  
 55/26  
 112/06  
 119/27  
 135/11  
 147/16  
 154/11  
 164/19  
 170/10  
 176/28  
 62/33  
 109/17  
 117/25  
 117/56  
 132/21  
 144/15  
 150/36  
 163/25  
 166/33  
 176/24  
 186/12  
 200/02  
 208/02  
 49/39  
 253/14  
 221/33  
 240/19

107/20  
 113/07  
 125/11  
 137/11  
 148/10  
 150/11  
 161/06  
 165/19  
 174/20  
 177/28  
 106/21  
 111/11  
 112/11  
 119/32  
 135/23  
 147/28  
 158/15  
 164/24  
 164/33  
 173/24  
 177/33  
 189/12  
 204/15  
 210/16  
 265/28  
 77/10  
 108/17  
 115/25  
 120/30  
 132/10  
 143/27  
 154/15  
 164/23  
 169/14  
 170/14  
 182/25  
 188/25  
 199/54  
 201/13  
 205/27  
 209/38  
 234/21

108/21  
 116/17  
 131/11  
 139/11  
 144/10  
 150/18  
 162/19  
 166/28  
 175/19  
 108/18  
 113/12  
 125/16  
 137/16  
 148/41  
 160/15  
 164/33  
 165/24  
 174/25  
 177/33  
 196/03  
 205/14  
 214/56  
 238/13  
 77/30  
 108/25  
 117/10  
 125/15  
 135/35  
 148/40  
 160/14  
 165/23  
 174/23  
 186/11  
 189/25  
 197/13  
 197/27  
 203/14  
 205/38  
 210/10  
 205/53  
 206/10

71/04  
 76/04  
 79/07  
 82/16  
 82/32  
 88/03  
 96/04  
 95/03  
 102/03  
 103/03  
 110/09  
 69/04  
 74/02  
 78/35  
 82/16  
 82/28  
 87/12  
 94/07  
 101/05  
 107/03  
 68/03  
 73/25  
 78/11  
 80/20  
 82/32  
 86/10  
 86/07  
 92/03  
 99/09  
 100/04  
 107/03  
 67/04  
 73/25  
 78/08  
 80/20  
 82/32  
 209/13  
 214/46  
 197/53  
 210/10  
 214/47

0281 ESPCL

ISZA0 006037  
 ISZ80 006066  
 ISZ81 006113  
 ISZ82 006143  
 ISZ83 006173  
 ISZ84 006223  
 ISZC2 006256  
 ISZD0 006316  
 ISZD1 006365  
 ISZD2 006437  
 ISZD3 006511  
 ISZSW 000003

246/15 252/01 252/58  
 156/06  
 157/01  
 158/01  
 159/01  
 160/01  
 161/02  
 162/01  
 163/02  
 164/01  
 165/01  
 166/01  
 167/01  
 168/01  
 169/01  
 170/01  
 171/01  
 172/01  
 173/01  
 174/01  
 175/01  
 176/01  
 177/01  
 178/01  
 179/01  
 180/01  
 181/01  
 182/01  
 183/01  
 184/01  
 185/01  
 186/01  
 187/01  
 188/01  
 189/01  
 190/01  
 191/01  
 192/01  
 193/01  
 194/01  
 195/01  
 196/01  
 197/01  
 198/01  
 199/01  
 200/01  
 201/01  
 202/01  
 203/01  
 204/01  
 205/01  
 206/01  
 207/01  
 208/01  
 209/01  
 210/01  
 211/01  
 212/01  
 213/01  
 214/01  
 215/01  
 216/01  
 217/01  
 218/01  
 219/01  
 220/01  
 221/01  
 222/01  
 223/01  
 224/01  
 225/01  
 226/01  
 227/01  
 228/01  
 229/01  
 230/01  
 231/01  
 232/01  
 233/01  
 234/01  
 235/01  
 236/01  
 237/01  
 238/01  
 239/01  
 240/01  
 241/01  
 242/01  
 243/01  
 244/01  
 245/01  
 246/01  
 247/01  
 248/01  
 249/01  
 250/01  
 251/01  
 252/01

157/19 158/19 159/01 159/19  
 160/19 162/28 163/02 163/41  
 164/42 165/01 166/01 166/40  
 168/19 169/19 170/01 170/19  
 171/19 173/28 173/30 174/02  
 174/41 175/01  
 176/40 177/01 177/40  
 178/40 179/01 180/01  
 181/15 182/16 183/16  
 184/16 185/16 186/16  
 187/16 188/16 189/16  
 190/16 191/16 192/16  
 193/16 194/16 195/16  
 196/16 197/16 198/16  
 199/16 200/16 201/16  
 202/16 203/16 204/16  
 205/16 206/16 207/16  
 208/16 209/16 210/16  
 211/16 212/16 213/16  
 214/16 215/16 216/16  
 217/16 218/16 219/16  
 220/16 221/16 222/16  
 223/16 224/16 225/16  
 226/16 227/16 228/16  
 229/16 230/16 231/16  
 232/16 233/16 234/16  
 235/16 236/16 237/16  
 238/16 239/16 240/16  
 241/16 242/16 243/16  
 244/16 245/16 246/16  
 247/16 248/16 249/16  
 250/16 251/16 252/16

JMLC4 000263  
 JMPA0 004476  
 JMP80 004520  
 JMP81 004543  
 JMP82 004566  
 JMP83 004614  
 JMP84 004701  
 JMP85 004727  
 JMP86 004755  
 JMP87 005017  
 JMP88 005074  
 JMP89 005154  
 JMP90 005234  
 JMSWC 000003

135/50 137/17 137/24 137/30 137/40 137/50 139/17  
 139/24 139/30 139/40 139/50 147/22 147/29 147/32  
 147/39 148/22 148/29 148/35 148/45 148/55 150/17  
 150/24 150/30 150/40 150/50 152/17 152/24 152/30  
 152/40 152/50 154/17 154/24 154/30 154/40 154/50  
 152/27 134/32 133/42 133/48 135/27 135/37 135/43  
 137/27 137/37 137/43 139/27 139/37 139/43  
 148/42 148/48 150/27 150/37 150/43 152/27  
 152/43 154/27 154/37 154/43  
 124/06  
 125/02  
 126/02  
 127/02  
 128/02  
 130/02  
 131/02  
 133/06  
 135/01  
 137/01  
 139/01  
 132/06  
 137/01  
 148/06  
 154/01  
 125/20  
 126/24  
 127/23  
 130/20  
 131/27  
 132/36  
 132/40  
 132/52  
 133/31  
 133/33  
 133/44  
 133/47  
 135/47  
 135/51  
 135/56  
 135/18  
 135/26  
 135/31  
 137/47  
 137/51  
 137/57  
 137/18  
 137/26  
 137/31  
 137/39  
 139/42  
 139/47  
 139/51  
 139/57  
 139/18  
 139/26  
 139/39  
 139/42  
 142/12  
 142/26  
 143/26  
 144/26  
 145/19  
 145/26  
 146/19  
 146/26  
 147/36  
 147/40  
 147/45  
 147/53  
 147/57  
 148/52  
 148/56  
 148/59  
 148/23  
 148/31  
 148/44  
 148/47  
 150/47  
 150/51  
 150/56  
 150/18  
 150/26  
 150/31  
 150/39  
 150/42

0282 ESPCL

JMEX 013230  
 JMFH 013212  
 JMFIL 013175  
 JMLC0 000257

246/15 252/01 252/58  
 156/06  
 157/01  
 158/01  
 159/01  
 160/01  
 161/02  
 162/01  
 163/02  
 164/01  
 165/01  
 166/01  
 167/01  
 168/01  
 169/01  
 170/01  
 171/01  
 172/01  
 173/01  
 174/01  
 175/01  
 176/01  
 177/01  
 178/01  
 179/01  
 180/01  
 181/01  
 182/01  
 183/01  
 184/01  
 185/01  
 186/01  
 187/01  
 188/01  
 189/01  
 190/01  
 191/01  
 192/01  
 193/01  
 194/01  
 195/01  
 196/01  
 197/01  
 198/01  
 199/01  
 200/01  
 201/01  
 202/01  
 203/01  
 204/01  
 205/01  
 206/01  
 207/01  
 208/01  
 209/01  
 210/01  
 211/01  
 212/01  
 213/01  
 214/01  
 215/01  
 216/01  
 217/01  
 218/01  
 219/01  
 220/01  
 221/01  
 222/01  
 223/01  
 224/01  
 225/01  
 226/01  
 227/01  
 228/01  
 229/01  
 230/01  
 231/01  
 232/01  
 233/01  
 234/01  
 235/01  
 236/01  
 237/01  
 238/01  
 239/01  
 240/01  
 241/01  
 242/01  
 243/01  
 244/01  
 245/01  
 246/01  
 247/01  
 248/01  
 249/01  
 250/01  
 251/01  
 252/01

157/19 158/19 159/01 159/19  
 160/19 162/28 163/02 163/41  
 164/42 165/01 166/01 166/40  
 168/19 169/19 170/01 170/19  
 171/19 173/28 173/30 174/02  
 174/41 175/01  
 176/40 177/01 177/40  
 178/40 179/01 180/01  
 181/15 182/16 183/16  
 184/16 185/16 186/16  
 187/16 188/16 189/16  
 190/16 191/16 192/16  
 193/16 194/16 195/16  
 196/16 197/16 198/16  
 199/16 200/16 201/16  
 202/16 203/16 204/16  
 205/16 206/16 207/16  
 208/16 209/16 210/16  
 211/16 212/16 213/16  
 214/16 215/16 216/16  
 217/16 218/16 219/16  
 220/16 221/16 222/16  
 223/16 224/16 225/16  
 226/16 227/16 228/16  
 229/16 230/16 231/16  
 232/16 233/16 234/16  
 235/16 236/16 237/16  
 238/16 239/16 240/16  
 241/16 242/16 243/16  
 244/16 245/16 246/16  
 247/16 248/16 249/16  
 250/16 251/16 252/16

JMLC1 000260  
 JMLC2 000261

135/50 137/17 137/24 137/30 137/40 137/50 139/17  
 139/24 139/30 139/40 139/50 147/22 147/29 147/32  
 147/39 148/22 148/29 148/35 148/45 148/55 150/17  
 150/24 150/30 150/40 150/50 152/17 152/24 152/30  
 152/40 152/50 154/17 154/24 154/30 154/40 154/50  
 152/27 134/32 133/42 133/48 135/27 135/37 135/43  
 137/27 137/37 137/43 139/27 139/37 139/43  
 148/42 148/48 150/27 150/37 150/43 152/27  
 152/43 154/27 154/37 154/43  
 124/06  
 125/02  
 126/02  
 127/02  
 128/02  
 130/02  
 131/02  
 133/06  
 135/01  
 137/01  
 139/01  
 132/06  
 137/01  
 148/06  
 154/01  
 125/20  
 126/24  
 127/23  
 130/20  
 131/27  
 132/36  
 132/40  
 132/52  
 133/31  
 133/33  
 133/44  
 133/47  
 135/47  
 135/51  
 135/56  
 135/18  
 135/26  
 135/31  
 137/47  
 137/51  
 137/57  
 137/18  
 137/26  
 137/31  
 137/39  
 139/42  
 139/47  
 139/51  
 139/57  
 139/18  
 139/26  
 139/39  
 139/42  
 142/12  
 142/26  
 143/26  
 144/26  
 145/19  
 145/26  
 146/19  
 146/26  
 147/36  
 147/40  
 147/45  
 147/53  
 147/57  
 148/52  
 148/56  
 148/59  
 148/23  
 148/31  
 148/44  
 148/47  
 150/47  
 150/51  
 150/56  
 150/18  
 150/26  
 150/31  
 150/39  
 150/42

0263 ESPCL

JR02 005753 152/47 152/51 152/57  
 JR020 005710 152/18 152/26  
 JR021 005723 152/31 152/39 152/42  
 JR03 006033 154/47 154/51 154/57  
 JR030 005770 154/18 154/26  
 JR031 006003 154/31 154/39 154/42  
 JSRA0 005314 141/06  
 JSRA1 005337 142/01  
 JSR80 005353 143/02  
 JSR81 005376 144/01  
 JSR82 005424 145/01  
 JSR83 005452 146/01  
 JSR84 005500 147/06  
 JSR80 005542 148/06  
 JSR81 005617 150/01  
 JSR82 005677 152/01  
 JSR83 005757 154/01  
 KRFLF 015022 261/29  
 KINST 000240 51/57  
 L001 001606 66/28  
 L002 001631 67/25  
 L003 001662 68/29  
 L004 001711 84/28  
 L005 001742 70/45  
 L006 002016 71/50  
 L007 002035 72/21  
 L010 002054 72/42  
 L011 002073 73/21  
 L012 002112 73/42  
 L013 002147 74/33  
 L014 002176 75/22  
 L015 002216 76/22  
 L016 002242 77/25  
 L017 002274 78/29  
 L020 002324 79/26  
 L021 002420 81/10  
 L022 002461 82/40  
 L023 002531 83/54  
 L024 002551 84/24  
 L025 002571 85/23  
 L026 002614 86/34  
 L027 002635 87/28  
 L030 002653 88/19  
 L031 002676 89/24  
 L032 002721 90/24  
 L033 002742 91/23  
 L034 002763 92/23  
 L035 003010 93/31  
 L036 003033 94/28  
 L037 003060 95/23  
 L040 003120 96/35  
 L041 003160 97/35  
 L042 003207 98/28  
 L043 003236 99/38  
 L044 003265 100/33  
 L045 003303 101/20  
 L046 003330 102/27  
 L047 003351 103/23

0284 ESPCL

L050 003372 104/23  
 L051 003425 105/38  
 L052 003470 106/49  
 L053 003535 107/47  
 L054 003603 108/49  
 L055 003651 109/48  
 L056 003704 110/44  
 L057 003725 111/25  
 L060 003750 112/27  
 L061 003774 113/29  
 L062 004020 114/29  
 L063 004060 115/50  
 L064 004124 116/52  
 L065 004173 117/51  
 L066 004243 118/53  
 L067 004315 119/52  
 L070 004352 120/41  
 L071 004403 121/51  
 L072 004443 122/40  
 L073 004472 123/25  
 L074 004517 124/32  
 L075 004542 125/29  
 L076 004565 126/29  
 L077 004613 127/32  
 L100 004700 129/15  
 L101 004726 130/32  
 L102 004754 131/32  
 L103 005016 132/51  
 L104 005073 134/04  
 L105 005153 136/01  
 L106 005233 138/02  
 L107 005313 140/02  
 L110 005336 141/34  
 L111 005352 142/18  
 L112 005375 143/29  
 L113 005425 144/31  
 L114 005451 145/31  
 L115 005477 146/31  
 L116 005541 147/50  
 L117 005616 149/04  
 L120 005676 151/01  
 L121 005756 153/02  
 L122 006036 155/02  
 L123 006065 156/38  
 L124 006112 157/30  
 L125 006142 158/32  
 L126 006172 159/33  
 L127 006222 160/33  
 L130 006255 161/36  
 L131 006315 162/43  
 L132 006364 163/54  
 L133 006436 164/55  
 L134 006510 165/56  
 L135 006562 166/56  
 L136 006611 167/38  
 L137 006634 168/30  
 L140 006666 169/32  
 L141 006716 170/33











0293 ESPCL

RNDAT 000750 MC 2/01  
 RNDOT 012317 197/17 199/57 201/17 201/57 203/18  
 RNDXX 012342 203/58 204/14 205/17 206/13 207/11 207/57  
 RNDYY 012335 208/13 209/17 209/57 210/13 211/15 212/15 214/15  
 214/51  
 RNDZZ 012317 2/01  
 RNDAA 012332 48/25 84/09 226/05  
 RNDAB 012335 226/24  
 RNDAC 012335 226/12  
 RNDAD 012272 49/23 99/37 225/20  
 RNDAM 012315 225/27 225/39  
 RNDAN 012310 225/34 225/40  
 RNDAP 012310 48/25  
 RNDQA 000666 MC 2/01  
 RNDQB 033321 2/01  
 RNDQC 015332 265/06  
 RNDQD 000616 54/00  
 RNDQE 003745 112/13 112/16 112/20  
 RNDQF 004030 MC 115/21 115/29  
 RNDQG 032220 2/01 54/40  
 RNDQH 004076 116/25 116/33  
 RNDQI 004166 117/30 117/44  
 RNDQJ 004142 117/21 117/29  
 RNDQK 004205 118/17 118/25  
 RNDQL 004255 119/16 119/24  
 RNDQM 004321 120/15 120/26  
 RNDQN 004382 120/31 120/39  
 RNDQO 004405 122/09 122/11 122/13 122/17 122/19 122/21 122/52  
 122/42  
 RNDQP 004474 123/10 123/27  
 RNDQR 000002 46/15  
 RNDQS 013300 47/13 62/25 242/04  
 RNDQT 015024 60/40 261/31  
 RNDQU 000416 MC 2/01 65/08 67/03 68/02 69/03 70/02 71/03  
 72/03 73/03 73/24 74/01 75/02 76/03  
 77/05 78/11 79/10 80/19 82/10 83/07 84/06  
 85/06 86/09 87/11 88/02 89/02 90/02 91/02  
 92/02 93/06 94/06 95/02 96/03 97/03 98/06  
 99/08 100/03 101/02 102/02 103/02 104/02 105/07  
 106/06 107/02 108/03 109/02 110/08 111/03 112/03  
 113/04 114/04 115/11 116/07 117/03 118/04 119/03  
 120/07 121/03 122/04 123/04 124/11 125/08 126/08  
 127/08 128/07 130/08 131/06 132/13 133/13 135/08  
 137/08 139/08 141/11 142/06 143/08 144/07 145/07  
 146/07 147/13 148/13 150/08 152/08 154/08 156/08  
 157/04 158/04 159/04 160/04 161/03 162/04 163/05  
 164/04 165/04 166/04 167/08 168/04 169/04 170/04  
 171/04 172/03 173/04 174/05 175/04 176/04 177/04  
 178/12 180/08 181/09 182/08 184/08 185/08 188/08  
 189/08 191/08 193/08 195/11 197/10 199/10 201/10  
 203/11 205/10 207/10 209/10 211/08 212/08 214/08  
 216/03 217/03 218/03 219/03  
 2/01  
 RNDQV 000504 MC 2/01  
 RNDQW 000083 46/25 59/23 80/16 244/22 245/38  
 RNDQX 000043 48/03  
 RNDQY 013016 59/52 60/58 236/05 236/16 244/20 245/10  
 SL 000042 46/24 59/20 80/14 86/16 86/23 56/52 56/59  
 SP 000040 46/22 46/60 56/06 56/16 56/23 244/16 245/16  
 59/16 80/10 80/28 80/54 86/13 244/16 245/16  
 245/29 249/60

0294 ESPCL

STAA0 003652 110/06  
 STAB0 003705 111/02  
 STAB1 003726 112/02  
 STAB2 003751 113/02  
 STAB3 003775 114/02  
 STAC2 004021 115/08  
 STAD0 004061 116/06  
 STAD1 004125 117/02  
 STAD2 004174 118/02  
 STAD3 004244 119/01  
 STAE0 004314 120/04  
 STAE1 004353 121/02  
 STAE2 004404 122/03  
 STAF5 004451 123/05  
 STAK8 001010 47/59 47/60 48/01 49/29 49/42 58/05 59/41  
 244/09  
 STAKF 013442 49/50 49/53 245/05 245/07 245/35 245/44  
 START 001306 60/57 62/04 266/03  
 STB9 002304 79/05 79/31  
 STK0 013421 227/27 231/05 234/34 244/08  
 STK1 013425 48/19 244/13  
 STK2 013427 244/11 244/16  
 STK3 000127 49/42 56/07  
 STK4 000130 49/43 80/15 244/21 244/14  
 STKR 013437 244/08 244/13 244/25 244/25 111/19 112/12 112/22  
 STLC0 000253 52/20 110/20 110/39 111/12 111/17 114/24 115/20  
 113/13 113/17 113/24 114/13 114/17 116/40 116/47  
 115/27 115/30 115/34 116/16 116/31 118/23 118/38  
 117/12 117/27 117/39 117/46 118/16 118/23 118/38  
 118/48 119/15 119/22 119/37 119/47 120/11 120/25  
 120/28 120/38  
 STLC1 000254 52/21 111/16 111/20 116/24 116/37 117/20 117/36  
 118/27 118/35 119/26 119/34 119/44 120/14 120/17  
 STSV0 000255 52/22 118/09 118/45 119/08 119/44 120/14 120/17  
 120/24  
 SWMS 001231 60/14 61/13  
 SWMS0 001241 60/21 60/26  
 SWMS1 001245 60/25 60/32  
 SWMS2 001262 60/37 60/49  
 SWMS3 015333 51/54 265/12 265/07  
 SWMS4 015313 60/51 264/34  
 SWMS5 002222 77/01 77/05  
 SWMS6 002231 77/14  
 SWMS7 002244 77/03 77/27  
 SWMS8 002113 74/01  
 SWMS9 002132 74/17  
 SWMSA 002145 74/28  
 SWMSB 002146 74/25 74/30  
 SWMSC 002147 74/22 74/32  
 T001 001472 65/09  
 T002 001607 67/04  
 T003 001632 68/03  
 T004 001663 69/04  
 T005 001712 70/03  
 T006 001743 71/04  
 T007 002017 72/04  
 T010 002036 72/25  
 T011 002055 73/04

0295 ESPCL

T012 002074 73/25  
 T013 002113 74/02  
 T014 002156 75/03  
 T015 002177 76/04  
 T016 002222 77/06  
 T017 002255 78/11  
 T020 002307 79/10  
 T021 002343 80/20  
 T022 002432 82/11  
 T023 002462 83/08  
 T024 002532 84/07  
 T025 002554 85/07  
 T026 002574 86/10  
 T027 002620 87/12  
 T030 002656 88/05  
 T031 002654 89/03  
 T032 002677 90/03  
 T033 002722 91/03  
 T034 002743 92/03  
 T035 002764 93/07  
 T036 003011 94/07  
 T037 003037 95/03  
 T040 003064 96/04  
 T041 003124 97/04  
 T042 003164 98/07  
 T043 003210 99/09  
 T044 003237 100/04  
 T045 003266 101/03  
 T046 003304 102/03  
 T047 003331 103/03  
 T050 003352 104/03  
 T051 003373 105/08  
 T052 003426 106/07  
 T053 003471 107/03  
 T054 003536 108/04  
 T055 003604 109/03  
 T056 003652 110/09  
 T057 003705 111/04  
 T060 003726 112/04  
 T061 003751 113/05  
 T062 003775 114/05  
 T063 004021 115/12  
 T064 004061 116/08  
 T065 004125 117/04  
 T066 004174 118/05  
 T067 004244 119/04  
 T070 004314 120/08  
 T071 004353 121/04  
 T072 004404 122/05  
 T073 004431 123/05  
 T074 004500 124/12  
 T075 004522 125/09  
 T076 004545 126/09  
 T077 004570 127/09  
 T100 004616 128/08  
 T101 004703 130/09  
 T102 004731 131/09  
 T103 004757 132/14

0296 ESPCL

T104 005021 133/14  
 T105 005076 135/09  
 T106 005156 137/04  
 T107 005236 139/09  
 T110 005316 141/12  
 T111 005341 142/07  
 T112 005355 143/09  
 T113 005400 144/08  
 T114 005426 145/08  
 T115 005454 146/08  
 T116 005502 147/14  
 T117 005544 148/14  
 T120 005621 150/09  
 T121 005701 152/09  
 T122 005761 154/09  
 T123 006037 156/09  
 T124 006066 157/05  
 T125 006113 158/05  
 T126 006145 159/05  
 T127 006173 160/05  
 T130 006223 161/04  
 T131 006256 162/05  
 T132 006316 163/06  
 T133 006365 164/05  
 T134 006437 165/05  
 T135 006511 166/05  
 T136 006563 167/09  
 T137 006612 168/05  
 T140 006657 169/05  
 T141 006667 170/05  
 T142 006717 171/05  
 T143 006747 172/04  
 T144 007003 173/05  
 T145 007043 174/06  
 T146 007112 175/05  
 T147 007164 176/05  
 T150 007236 177/05  
 T151 007312 178/13  
 T152 007363 180/09  
 T153 007435 181/10  
 T154 007507 182/09  
 T155 007567 184/09  
 T156 007651 186/09  
 T157 007732 188/09  
 T160 010005 189/09  
 T161 010064 191/09  
 T162 010147 193/09  
 T163 010231 195/12  
 T164 010330 197/11  
 T165 010432 199/11  
 T166 010534 201/11  
 T167 010636 203/12  
 T170 010757 205/11  
 T171 011103 207/11  
 T172 011227 209/11  
 T173 011355 211/09  
 T174 011425 212/09  
 T175 011504 214/09

0297 ESPCL

1176 011574  
 1177 011620  
 1200 011644  
 1201 011673  
 TBLSI 000302  
  
 182/22  
 198/27  
 195/25  
 201/29  
 209/24  
 214/27  
 247/37  
 52/41  
 76/16  
 79/04  
 51/35  
 247/12  
 247/41  
 48/22  
 102/16  
 251/12  
 251/13  
 251/14  
 251/15  
 251/16  
 251/17  
 251/18  
 251/19  
 50/19  
 2/01  
 49/02  
 46/19  
 46/17  
 2/01  
 47/35  
 47/46  
 47/53  
 50/34  
 224/06  
 232/33  
 232/37  
 243/01  
 243/01  
 243/01  
 243/01  
 XAC1 000005  
 XAC2 000006  
 XAC3 000007  
 XFERN 012614  
 XFERR 012515  
 XHLT0 000647  
 XHLT1 000657  
 XHLT2 000665  
 XHLT3 000674

0298 ESPCL

216/04  
 217/04  
 218/04  
 219/04  
 52/04  
 182/22  
 198/27  
 195/25  
 201/29  
 209/24  
 214/27  
 247/37  
 52/41  
 76/16  
 79/04  
 51/35  
 247/12  
 247/41  
 48/22  
 102/16  
 251/12  
 251/13  
 251/14  
 251/15  
 251/16  
 251/17  
 251/18  
 251/19  
 50/19  
 2/01  
 49/02  
 46/19  
 46/17  
 2/01  
 47/35  
 47/46  
 47/53  
 50/34  
 224/06  
 232/33  
 232/37  
 243/01  
 243/01  
 243/01  
 243/01  
 XAC1 000005  
 XAC2 000006  
 XAC3 000007  
 XFERN 012614  
 XFERR 012515  
 XHLT0 000647  
 XHLT1 000657  
 XHLT2 000665  
 XHLT3 000674

178/31  
 180/22  
 180/27  
 181/23  
 181/28  
 186/22  
 186/27  
 193/22  
 193/27  
 199/24  
 201/24  
 207/24  
 212/22  
 212/27  
 214/22  
  
 74/14  
 75/10  
 75/16  
 78/15  
 78/23  
 78/30  
 78/31  
 79/27  
 79/28  
 83/26  
 83/27  
 83/28  
 83/29  
 83/30  
 83/31  
 83/32  
 83/33  
 83/34  
 83/35  
 83/36  
 83/37  
 83/38  
 83/39  
 83/40  
 83/41  
 83/42  
 83/43  
 83/44  
 83/45  
 83/46  
 83/47  
 83/48  
 83/49  
 83/50  
 83/51  
 83/52  
 83/53  
 83/54  
 83/55  
 83/56  
 83/57  
 83/58  
 83/59  
 83/60  
 83/61  
 83/62  
 83/63  
 83/64  
 83/65  
 83/66  
 83/67  
 83/68  
 83/69  
 83/70  
 83/71  
 83/72  
 83/73  
 83/74  
 83/75  
 83/76  
 83/77  
 83/78  
 83/79  
 83/80  
 83/81  
 83/82  
 83/83  
 83/84  
 83/85  
 83/86  
 83/87  
 83/88  
 83/89  
 83/90  
 83/91  
 83/92  
 83/93  
 83/94  
 83/95  
 83/96  
 83/97  
 83/98  
 83/99  
 83/100

X3 000310  
 X4 000003  
 X5 000020  
 X6 000076  
 X9 000076  
 XAC1 000005  
 XAC2 000006  
 XAC3 000007  
 XFERN 012614  
 XFERR 012515  
 XHLT0 000647  
 XHLT1 000657  
 XHLT2 000665  
 XHLT3 000674

56/35  
 56/42  
 56/43  
 57/02  
 233/02  
 232/07  
 230/01  
 230/35  
 221/22  
 252/03  
 49/51  
 55/16  
 55/41  
 56/06  
 57/04  
 56/58  
 57/06  
 55/20  
 57/34  
 55/21  
 55/39  
 53/22  
 57/36  
 33/23  
 57/29  
 55/30  
 57/27  
 57/38  
 55/19  
 55/30  
 56/17  
 57/39  
 57/06  
 57/11  
 57/40  
 46/35  
 54/60  
 86/18  
 250/49  
 250/51  
 250/55  
 250/60  
 251/06  
 251/09  
  
 249/43  
 249/52  
 51/08  
 59/49  
 2/01  
 58/40  
 34/60  
 55/40  
 54/18  
 54/18  
 54/18  
 58/12  
 54/18  
 54/18  
 50/40  
 49/22  
 59/26  
 60/34  
 237/24  
 249/31  
 249/33  
 253/21  
 225/04  
 58/02  
 58/60

