

LISTING

096-000144-08

PROGRAM

EXERCISER

TAPE

095-000012-08

ABSTRACT

EXERCISER IS A MAINTENANCE PROGRAM DESIGNED TO TEST FOR RELIABLE OPERATION OF THE PROCESSOR INSTRUCTIONS AND THE PAPER TAPE EQUIPMENT. THE PROGRAM MAY EXERCISE THE TELETYPE READER/PUNCH, HIGH SPEED READER/PUNCH, THE REAL TIME CLOCK, AND THE NOVA INSTRUCTIONS. THE DEVICES TO BE USED ARE SELECTED BY CONSOLE SWITCHES AND ARE SERVICED VIA THE INTERRUPT SYSTEM.

0001 .MAIN

```

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

```

NAME: EXERCISER.SH PART NUMBER: 004-000011

DESCRIPTION: EXERCISER

REVISION HISTORY:

REV.	DATE
00	05/01/69
01	03/10/71
02	09/10/73
03	11/30/73
04	01/25/74
05	06/07/74
06	04/10/75
07	XX/XX/XX
08	06/11/76

COPYRIGHT (C) DATA GENERAL CORPORATION, 1969, 1970, 1971,
1973, 1974, 1975, 1976
ALL RIGHTS RESERVED.

A 0002 .MAIN

```

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

```

EXERCISER

11. ABSTRACT

EXERCISER IS A MAINTENANCE PROGRAM DESIGNED TO TEST FOR RELIABLE OPERATION OF THE PROCESSOR INSTRUCTIONS AND THE PAPER TAPE EQUIPMENT. THE PROGRAM MAY EXERCISE THE TELETYPE READER/PUNCH, HIGH SPEED READER/PUNCH, THE REAL TIME CLOCK, AND THE NOVA INSTRUCTIONS. THE DEVICES TO BE USED ARE SELECTED BY CONSOLE SWITCHS AND ARE SERVICED VIA THE INTERRUPT SYSTEM.

12. MACHINE REQUIREMENTS

12.1 STANDERD NOVA/ECLIPSE PROCESSOR

12.2 4K READ/WRITE MEMORY, MINIMUM

12.3 OPTIONAL EQUIPMENT

12.3.1 TELETYPE READER

12.3.2 TELETYPE PUNCH

12.3.3 HIGH SPEED READER

12.3.4 HIGH SPEED PUNCH

12.3.5 REAL TIME CLOCK

13. SWITCH SETTINGS

13.1 STARTING ADDRESS #00002

13.2 SWITCH 0(1) #ACTIVATE HIGH SPEED PUNCH

13.3 SWITCH 1(1) # " " READER

13.4 SWITCH 2(1) # " TELETYPE READER

13.5 SWITCH 3(1) # " " PUNCH

13.6 SWITCH 4(1) # " REAL TIME CLOCK

13.7 SWITCH 5(1) #ACTIVATE SECOND PTR

13.8 SWITCH 6(1) #ACTIVATE SECOND PTR

13.9 SWITCH 7(1) #ACTIVATE SECOND TTI

13.10 SWITCH 8(1) #ACTIVATE SECOND TTO

14. OPERATING PROCEEDURE

14.1 LOAD THE PROGRAM VIA THE BINARY LOADER

14.2 SET SWITCHES TO 00002

14.3 PRESS START

14.4 THE PROGRAM WILL RUN UNTILL MANUALLY STOPPED OR A ERROR IS DETECTED.

14.5 IN-OUT EQUIPMENT TESTING

14.5.1 BY SETTING SWITCHES 0-4 THE CORRESPONDIN DEVICE WILL BE ACTIVATED. THE SWITCHS AR READ AT THE COMPLETION OF EACH PASS OF T INSTRUCTION TEST. IF A CONSOLE SWITCH HA BEEN SET/CLEARED DURING THE INSTRUCTION TEST THE DEVICE WILL BE ACTIVATED/DEACTI AT THE END OF A PASS.

14.5.2 TO TEST THE HIGH SPEED READER AND PUNCH

14.5.2.1 RAISE THE READER LEVER

14.5.2.2 SET SWITCHES 0 AND 1

14.5.2.3 WHEN 5 TO 6 FEET OF TAPE HAVE BEEN PUNCHED PRESS STOP.

14.5.2.4 INSERT THE TAPE INTO THE READER

14.5.2.5 CLOSE THE LEVER

14.5.2.6 PRESS CONTINUE

14.5.2.7 THE TAPE WILL BE CHECKED

A 0003 .MAIN

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
14.5.3          TO TEST THE TELETYPE READER AND PUNCH
14.5.3.1        SET THE TELETYPE READER TO FREE
14.5.3.2        PRESS THE PUNCH ON BUTTON
14.5.3.3        SET SWITCHES 2 AND 3
14.5.3.4        WHEN SEVERAL INCHES OF TAPE
14.5.3.5        HAVE BEEN PUNCHED INSERT THE
14.5.3.6        LEADER... INTO THE READER,
14.5.3.7        SET THE READER SWITCH TO START
14.5.3.8        THE TAPE WILL BE CHECKED

15.  PROGRAM OUTPUT/ERROR DISCUSSION
15.1  WHEN A MALFUNCTION IS DETECTED THE PROGRAM
15.2  WILL HALT, CONSULT THE LISTING FOR THE
15.3  CAUSE OF ERROR.
15.4  UPON CONTINUING FROM A ERROR MOST ROUTINES
15.5  WILL ENTER A LOOP USING THE FAILING OPERANDS,
15.6  MOST ROUTINES ISSUE A "P" PULSE(474) FOR
15.7  SYNCING A SCOPE, IF POSSIBLE TRY TO OBSERVE
15.8  THE ERROR WITHOUT THE I/O EQUIPMENT RUNNING.

16.  PROGRAM DISCUSSION/THEORY OF OPERATION
16.1  THE EXERCISER PROGRAM IS DESIGNED TO PRODUCE
16.2  A INTERNAL MACHINE ENVIRONMENT AS SIMILAR
16.3  TO NORMAL OPERATING CONDITIONS AS POSSIBLE.
16.4  THE PROGRAM ISSUES A I/O RESET PULSE AND EXECUTE
16.5  THE INSTRUCTION TEST, THE INSTRUCTION TEST USES
16.6  A DATA BUFFER WHOS START AND STOP ADDRESS ARE
16.7  LOADED AS C(BUFF) AND C(FIN), AT THE END OF EACH
16.8  PASS THE BUFFER START AND STOP PARAMETERS ARE
16.9  ADJUSTED TO THE MAXIMUM MEMORY SIZE, THE
16.10 INSTRUCTIONS WILL BE TESTED WITH THE SYSTEM
16.11 MEMORY SIZE ON SUCESSIVE PASSES, THE PROGRAM
16.12 WILL NOT USE THE LAST 200 LOCATIONS OF THE
16.13 HIGHEST MEMORY MODULE (LOCATION OF LOADERS).
16.14 AT THE END OF EACH PASS THE CONSOLE SWITCHS
16.15 ARE READ, IF SWITCHS 0-8 ARE ALL ZERO THE
16.16 PROGRAM WILL RETURN TO CYCLE THE INSTRUCTION
16.17 TEST, IF A SWITCH HAS BEEN SET THE CORRESPOND-
16.18 ING DEVICE WILL BE ACTIVATED, AND THE INTERRUPT
16.19 SYSTEM TURNED ON, AFTER THE SWITCHS HAVE BEEN
16.20 INTEROGATED AND THE MEMORY SIZE DETERMINED THE
16.21 PROGRAM RETURNS TO THE INSTRUCTION TEST FOR
16.22 ANOTHER PASS.
16.23 THE DEVICES ARE SERVICED VIA THE INTERRUPT
16.24 SYSTEM, THE "INTA" INSTRUCTION IS USED TO
16.25 DISPATCH TO THE ROUTINE REQUESTING SERVICE,
16.26 IF THE INTA READS BACK A IMPROPER DEVICE CODE
16.27 A HALT WILL OCCUR, WHEN DEVICES
16.28 ARE ACTIVE A HALT OCCURING WITH "ION" LIGHT OFF
16.29 INDICATES A ERROR AT THE INTERRUPT SERVICE LEVEL

```

A 0004 .MAIN

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
16.4  THE REAL TIME CLOCK IS ACTIVATED BY
16.5  SWITCH 4, THE CLOCK IS SET TO INTERRUPT
16.6  AT A 1000 HERTZ RATE, WHEN A CLOCK INTERRUPT
16.7  OCCURS THE BUSY AND DONE FLAGS ARE TESTED
16.8  BEFORE AND AFTER THE CLOCK IS RESTARTED.
16.9  THE TELETYPE PUNCH IS ACTIVATED BY
16.10 SWITCH 3, THE PUNCH WILL PUNCH SEVERAL
16.11 INCHES OF LEADER FOLLOWED BY NON
16.12 ZERO,EVEN PARITY,RANDOM CHARACTORS,
16.13 THE PUNCHING WILL CONTINLE UNTILL
16.14 SWITCH 3 IS DEPRESED, AT WHICH TIME
16.15 TRAILER WILL BE PUNCHED.
16.16 THE TELETYPE READER IS ACTIVATED BY
16.17 SWITCH 2, THE READER WILL ACCEPT TAPE
16.18 PUNCHED EITHER ON THE TELETYPE OR HIGH
16.19 SPEED PUNCH, THE READER WILL INITIALIZE
16.20 ITS RANDOM NUMBER GENERATOR ON A ZERO CHAR-
16.21 ACTOR, IT IS NECESSARY TO LOAD THE READER IN
16.22 THE LEADER SECTION OF THE TAPE.
16.23 THE HIGH SPEED READER IS ACTIVATED BY SWITCH
16.24 1, THE READER WILL READ AND CHECK TAPES
16.25 PRODUCED ON EITHER THE TELETYPE OR HIGH
16.26 SPEED PUNCH, IF THE PUNCH IS ACTIVE
16.27 WHEN TAPES ARE BEING READ THE READER WILL BE
16.28 SYNCHRONIZED WITH THE PUNCH SUCH THAT THE TAPE
16.29 BEING PUNCHED MAY BE LOADED INTO THE READER,
16.30 IF THE HIGH SPEED PUNCH IS NOT ACTIVE A PRE-
16.31 PUNCHED TAPE WILL BE READ AT HIGH SPEED.
16.32 THE HIGH SPEED PUNCH IS ACTIVATED BY SWITCH 0,
16.33 THE PUNCH WILL PUNCH 256 LINES OF LEADER
16.34 FOLLOWED BY NON ZERO,EVEN PARITY,RANDOM
16.35 CHARACTORS, PUNCHING WILL CONTINUE UNTILL
16.36 SWITCH 0 IS DEPRESED, AT WHICH TIME TRAILER
16.37 WILL BE PUNCHED.

17.  PESTRICTIONS/MISC
17.1  BOTH THE TELETYPE AND HIGH SPEED READER MUST
17.2  BE IN THE LEADER SECTION OF THE TAPE WHEN
17.3  THEY ARE STARTED.
17.4  SWITCHES ARE READ ONLY AT THE END OF A PASS,
17.5  THEREFOR, THERE IS A DELAY BETWEEN THE SWITCH
17.6  ACTION AND THE DEVICE RESPONSE.
17.7  DERUGING IS DIFFICULT IN THE ENVIRONMENT
17.8  CREATED BY ACTIVE I/O EQUIPMENT, WHENEVER
17.9  POSSIABLE USE THE DIAGNOSTIC FOR THE
17.10 PARTICULAR I/O DEVICE FAILING.
17.11 PRESSING A TELETYPE KEY MAY CAUSE A ERROR,
17.12 READING A CHARACTER OF ZEROS WILL
17.13 INITIALIZE THE RANDOM NUMBER TO 53 OCTAL.
17.14 THIS IS A IMPORTANT NUMBER IF PAPER TAPE
17.15 READERS ARE FAILING BY READING ALL ZEROS!

```

```

A 0005 .MAIN
01
02      000000      .LOC 0
03 00000 003501    DIRT
04 00001 002711    INTR
05 00002 002003    JMP 0.+1
06 00003 000000    REG
07 00004 003077    HALT
08 00005 000004    JMP .-1
09      000045      .LOC 45
10 00045 000045    EGGS
11 00046 000000    EGGS: 0
12 00047 000000    0
13 00050 000000    0
14 00051 000000    0
15 00052 000000    0
16 00053 000000    0
17
18
19 00054 001005    ITAB: LSR
20 00055 001145    TYO
21 00056 001044    HSR1
22 00057 000744    PUN
23 00060 001201    CLOCK
24 00061 000004    4
25 00062 000004    4
26 00063 000004    4
27 00064 001332    .LSR
28 00065 001366    .TYO
29 00066 001311    .HSR1
30 00067 001206    .PUN
31
32 00070 000000    LSR0: 0
33 00071 000000    LSRCH: 0
34 00072 000000    KRAN: 0
35 00073 000000    HSR0: 0
36 00074 000000    HSRCH: 0
37 00075 000000    READER: 0
38 00076 000000    RHAN: 0
39 00077 000000    PHAV: 0
40 00100 000000    DOT: 0
41 00101 000377    LT: 377
42 00102 000000    PUNCH: 0
43 00103 000377    RPDIF: 377
44 00104 000000    C170: 0
45 00105 000000    ACTION: 0
46 00106 135525    C13552: 135525
47 00107 000000    RER: 0
48 00110 000177    C177: 177
49 00111 000010    C10: 10
50 00112 001121    CXRAND: XRAND
51 00113 000000    ISAV0: 0
52 00114 000000    ISAV1: 0
53 00115 000000    ISAV2: 0
54 00116 000000    ISAV3: 0
55 00117 000000    ISAVC: 0
56 00120 000000    TRAN: 0
57 00121 000000    TYPE: 0

```

```

DIRT BLOCK PNTR FOR DTOS
INTERRUPT FROM PLT,COR,DIS
JEGGS BLK PNTR FOR DTOS
DTOS AUTO MODE SW
DEVICE CODE
ICAT SWITCH
# OF PASSES
DTOS RETURN ADDR
ISWITCH REGISTER
JKEYBOARD INTERRUPT
ITTO
IREADER
IPUNCH
JKEYBOARD CHAR BUFFER
J" "
J" RANDOM
JREADER CHAR BUFFER
J" "
J" CHAR COUNTER
J" RANDOM
JPUNCH RANDOM
JOUT OF TAPE FLAG
JLEADER TRAILER
JPUNCH CHAR COUNT
JTAPE LENGTH BETWEEN
JACTIVE DEVICES

```

```

A 0006 .MAIN
01
02 00122 000000    TEM: 0
03 00123 177773    CM5: -5
04 00124 000004    C4: 4
05 00125 007777    MSIZE: 7777
06 00126 000040    C40: 40
07 00127 003000    RUFF: PRGEN0
08 00130 004500    FIN: PRGEN0+1000
09 00131 004300    FIN200: PRGEN0+1000-200
10 00132 003700    RUF200: PRGEN0+200
11 00133 000003    C3: 3
12 00134 000005    C5: 5
13 00135 000377    C377: 377
14 00136 000000    CF00: 0
15 00137 177600    C170X: 177600
16 00140 001422    MAIN: STAN
17 00141 007000    C3000: 7000
18 00142 001000    C1000: 1000
19 00143 000200    C200: 200
20 00144 000000    STP: 0
21 00145 000000    STT: 0
22 00146 000000    TIN: 0
23 00147 000000    TIMEY: 0
24
25
26 00150 000037    C37: 37
27 00151 000000    .STP: 0
28 00152 000000    .RRAN: 0
29 00153 000000    .PRAN: 0
30 00154 000000    .PUNCH: 0
31 00155 000000    .HEAD: 0
32 00156 000000    .MSRH: 0
33 00157 000000    .MSRCH: 0
34 00160 000000    .TIMEY: 0
35 00161 000000    .TIN: 0
36 00162 000000    .LSRH: 0
37 00163 000000    .LSRCH: 0
38 00164 000000    .KRAN: 0
39 00165 000000    .TYPE: 0
40 00166 000000    .TRAN: 0
41 00167 000000    .STT: 0
42 00170 000733    DISMIS: EN0IT
43 00171 001422    ICAT: STAN
44 00172 000000    C400: 400
45 00173 000200    C200: 200
46 00174 000200    C2002: 2002
47
48      000200      .LOC 200
49
50      J STANDARD START AT 200
51
52
53 00200 002201    DTOSH: JMP 0.+1
54 00201 000556    MEMRZ
55      000203      .LOC 203
56 00203 000000    PASS: 0
57
58      000053 .PTP=53
59      000052 .PTH=52

```

JPASS COUNT

```

0007 .MAIN
01 000050 .TTI=50
02 000051 .TTO=51

```

```

A 0000 .MAIN
01
02 000500 .LOC 500
03
04 00500 102400 REG: SJR 0,0 ISTARTS HERE
05 00501 040203 STA 0,PASS ICLEAR PASS COUNT
06 00502 040105 STA 0,ACTION
07 00503 002140 JMP 0MAIN IPROCESSOR TEST
08
09 00504 030105 REG: LDA 2,ACTION
10 00505 064477 READS 1
11 00506 150000 COM 2,2
12 00507 133502 ANDL 1,2,SZC
13 00510 004522 JSR PTPSI ISTART PUNCH
14 00511 151102 MOVL 2,2,SZC
15 00512 004531 JSR PTRSU ISTART HEADER
16 00513 151102 MOVL 2,2,SZC
17 00514 004535 JSR TTISU ISTART TTI READER
18 00515 151102 MOVL 2,2,SZC
19 00516 004505 JSR TTOSU ISTART TTO OUTPUT
20 00517 151102 MOVL 2,2,SZC
21 00520 004500 JSR RTCSU ISTART REAL TIME CLOCK
22 00521 151102 MOVL 2,2,SZC ISECOND PUNCH
23 00522 004552 JSR ,TPSI
24 00523 151102 MOVL 2,2,SZC ISECOND READER
25 00524 004561 JSR ,TRSU
26 00525 151102 MOVL 2,2,SZC ISECOND TTI
27 00526 004532 JSR ,TISU
28 00527 151102 MOVL 2,2,SZC ISECOND TTO
29 00530 004535 JSR ,TOSU
30
31 00531 030105 LDA 2,ACTION
32 00532 044105 STA 1,ACTION
33 00533 124000 COM 1,1
34 00534 133502 ANDL 1,2,SZC
35 00535 004500 JSR PTPSD ISTOP PUNCH
36 00536 151102 MOVL 2,2,SZC
37 00537 004502 JSR PTRSD
38 00540 151102 MOVL 2,2,SZC
39 00541 004506 JSR TTISD ISTOP TTI READER
40 00542 151102 MOVL 2,2,SZC
41 00543 004403 JSR TTOSD ISTOP TTO
42 00544 151102 MOVL 2,2,SZC
43 00545 060214 NIOC RTC ISTOP CLOCK
44 00546 151102 MOVL 2,2,SZC
45 00547 004530 JSR ,TPSD ISTOP SECOND PTP
46 00550 151102 MOVL 2,2,SZC
47 00551 004532 JSR ,TRSD ISTOP SECOND PTR
48 00552 151102 MOVL 2,2,SZC
49 00553 004503 JSR ,TISD ISTOP SECOND TTI
50 00554 151102 MOVL 2,2,SZC
51 00555 004513 JSR ,TOSD ISTOP SECOND TTO

```

A 0004 ,MAIN

```

01
02 00556 024141 MEMSZ: LDA 1,C3000
03 00557 020142 LDA 0,C1000
04 00560 107000 ADD 0,1
05 00561 044125 STA 1,MSIZE
06 00562 125112 MOVL# 1,1,SZC
07 00563 000405 JMP ,+5
08 00564 036125 LDA 3,0MSIZE
09 00565 046125 STA 1,0MSIZE
10 00566 032125 LDA 2,0MSIZE
11 00567 056125 STA 3,0MSIZE
12 00570 132415 SUB# 1,2,SNR
13 00571 000766 JMP MEMSZ+1
14 00572 014125 DSZ MSIZE
15 00573 000401 JMP ,+1
16 00574 020173 LDA 0,C200
17 00575 034050 LDA 3,EGGS+2
18 00576 175004 MOV 3,3,SZR
19 00577 020174 LDA 0,C2000
20 00600 024125 LDA 1,MSIZE
21 00601 106400 SUB 0,1
22 00602 020172 LDA 0,C400
23 00603 123000 ADD 1,0
24 00604 101400 INC 0,0
25 00605 175004 MOV 3,3,SZR
26 00606 040171 STA 0,ICAT
27 00607 020143 LDA 0,C200
28 00610 044130 STA 1,FIN
29 00611 106400 SUB 0,1
30 00612 044131 STA 1,FIN200
31 00613 020137 LDA 0,C174X
32 00614 024105 LDA 1,ACTION
33 00615 107404 AND 0,1,SZR
34 00616 060177 NINS CPU
35 00617 002140 JMP 0MAIN

```

ISIZE THE MEMORY

MEMORY MASK

IF SWITCHES SET TURN
ON INTERRUPT ENABLE.

A 0010 ,MAIN

```

01
02
03 00620 020133 RTCSU: LDA 0,C3
04 00621 061114 DCAS 0,RTC
05 00622 001400 JMP 0,3
06
07 00623 102400 TTOSU: SUB 0,0
08 00624 061111 DCAS 0,TT0
09 00625 102401 SUB 0,0,SKP
10 00626 102000 TTOSD: ADC 0,0
11 00627 040121 STA 0,TYPE
12 00630 040145 STA 0,STT
13 00631 001400 JMP 0,3
14 00632 102400 PTPSU: SUB 0,0
15 00633 061113 DCAS 0,PTP
16 00634 102401 SUB 0,0,SKP
17 00635 102000 PTPSD: ADC 0,0
18 00636 040102 STA 0,PUNCH
19 00637 040144 STA 0,STP
20 00640 001400 JMP 0,3
21
22 00641 060212 PTRSD: NIOC PTR
23 00642 101011 MOV# 0,0,SKP
24 00643 060112 PTRSH: NINS PTR
25 00644 102400 SUB 0,0
26 00645 040075 STA 0,READER
27 00646 001400 JMP 0,3
28
29 00647 010147 TTISD: ISZ TINEY
30 00650 001400 JMP 0,3
31 00651 060110 TTISU: NICS TTI
32 00652 102400 SUB 0,0
33 00653 040146 STA 0,TIN
34 00654 040147 STA 0,TINEY
35 00655 001400 JMP 0,3

```

START REAL TIME CLOCK

TELETYPE PUNCH

HIGH SPEED PUNCH

READER

```

A 0011 .MAIN
01
02 00656 010100 .TISD: IS7 .TINEY
03 00657 001400 JMP 0,3
04 00660 000150 .TISU: NIOS .TTI
05 00661 102400 SUR 0,0
06 00662 040151 STA 0,,TIA
07 00663 040100 STA 0,,TINEY
08 00664 001400 JMP 0,3
09
10 00665 102400 .TOSU: SUB 0,0
11 00666 061151 OQAS 0,,TTP
12 00667 102401 SUB 0,0,SKP
13 00670 102000 .TOSD: ADC 0,0
14 00671 040105 STA 0,,TYPE
15 00672 040167 STA 0,,STT
16 00673 001400 JMP 0,3
17
18 00674 102400 .TPSU: SUB 0,0
19 00675 061153 OQAS 0,,PTP
20 00676 102401 SUB 0,0,SKP
21 00677 102000 .TPSD: ADC 0,0
22 00700 040154 STA 0,,PUNCH
23 00701 040151 STA 0,,STP
24 00702 001400 JMP 0,3
25
26 00703 060252 .TRSD: NIOC .PTR
27 00704 101011 MOVN 0,0,SKP
28 00705 060152 .TRSU: NIOS .PTR
29 00706 102400 SUR 0,0
30 00707 040155 STA 0,,READER
31 00710 001400 JMP 0,3
32

```

ISECOND TTI

ISECOND TTD

ISECOND PTP

ISECOND PTR

```

A 0012 .MAIN
01 00711 040113 INTR: STA 0,ISAV0
02 00712 040114 STA 1,ISAV1
03 00713 050115 STA 2,ISAV2
04 00714 050116 STA 3,ISAV3
05 00715 175000 MOVR 3,3
06 00716 050117 STA 3,ISAVC
07 00717 061477 INTA 0
08 00720 030150 LDA 2,C37
09 00721 113520 ANDZL 0,2,
10 00722 024126 LDA 1,C40
11 00723 107020 ANDZR 0,1
12 00724 133220 ADDZR 1,2
13 00725 024111 LDA 1,C10
14 00726 034415 LDA 3,C24
15 00727 172433 SURZW 3,2,SNC
16 00730 132423 SURZ 1,2,SNC
17 00731 063077 HALT
18 00732 003054 JMP 0IAB,2
19
20 00733 020117 ENDIT: LDA 0,ISAVC
21 00734 101100 MOVL 0,0
22 00735 020113 LDA 0,ISAV0
23 00736 024114 LDA 1,ISAV1
24 00737 030115 LDA 2,ISAV2
25 00740 030116 LDA 3,ISAV3
26 00741 060177 INTEN
27 00742 002000 JMP 00
28
29 00743 000224 C24: 24
30 00744 000112 PUN: JSR 0CXRAND
31 00745 000077 PRAN
32 00746 024101 LDA 1,LT
33 00747 030102 LDA 0,PUNCH
34 00750 132032 ANDZW 1,2,SZC
35 00751 000425 JMP PUN1
36 00752 020106 LDA 0,C135525
37 00753 040077 STA 0,PRAN
38 00754 024110 LDA 1,C177
39 00755 034144 LDA 3,STP
40 00756 133405 AND 1,2,SNR
41 00757 175005 MOV 3,3,SNR
42 00760 000415 JMP PUNX
43 00761 000403 JMP .+3
44 00762 000000 0
45 00763 000000 0
46
47 00764 060213 NIOC PTP
48 00765 063412 SKPSN PTR
49 00766 063712 SKPDZ PTR
50 00767 082170 JMP 0DISMIS
51 00770 020105 LDA 0,ACTION
52 00771 101100 MOVL 0,0
53 00772 101102 MOVL 0,0,SZC
54 00773 060112 NIOS PTR
55 00774 002170 JMP 0DISMIS
56 00775 102400 PUNX: SUB 0,0
57

```

INTERRUPT DISPATCH
ISAVE INTERRUPTS

UNKNOWN INTERRUPT?
ISERVICE THE DEVICE

IRESTORE,EXIT INTERRUPT

IPUNCH INTERRUPT

ILEADER/TRAILER
ICHECK

IINIT RANDOM ON L/T

ISTOP PUNCHING

IRESTART READER

IPUNCH LEADER/TRAILER

A 0013 .MAIN

```

01
02
03 00776 010102 PUN1: ISZ PUNCH      ICHAR COUNTER
04 00777 101001      MOV R,0,SKP
05 01000 010102      ISZ PUNCH
06 01001 063413      SKPBN PTP
07 01002 063613      SKPBN PTP
08 01003 063077      HALT
09 01004 061113      DOAS R,PTP
10 01005 063513      SKPDZ PTP
11 01006 063713      SKPDZ PTP
12 01007 063077      HALT
13
14 01010 020102      LDA R,PUNCH
15 01011 020075      LDA 1,READER
16 01012 030103      LDA 2,RPDIF
17 01013 100400      SUB R,1
18 01014 133404      AND 1,2,SZR
19 01015 002170      JMP #DISMISS
20
21
22
23 01016 063412      SKPBN PTR
24 01017 063712      SKPDZ PTR
25 01020 002170      JMP #DISMISS
26 01021 020105      LDA R,ACTION
27 01022 101100      MOVL R,R
28 01023 101112      MOVL R,R,SZC
29 01024 060112      MINS PTR
30 01025 002170      JMP #DISMISS
31
32 01026 064512 HSR1: DIAS 1,PTR
33 01027 020073      LDA R,HSR1
34 01030 044073      STA 1,HSR1
35 01031 040074      STA R,HSRCH
36 01032 030075      LDA 2,READER
37 01033 101004      MOV R,R,SZR
38 01034 151235      MOVZR R,2,2,SNK
39 01035 000425      JMP HSR1
40 01036 000112      JSR #CXRAND
41 01037 000076      RKAN
42 01040 020074      LDA 1,HSRCH
43 01041 100414      SURR R,1,SZR
44 01042 063077      HALT
45 01043 002170      JMP #DISMISS
46
47

```

A 0014 .MAIN

```

01
02 01044 020105 HSR1: LDA R,ACTION
03 01045 012075      ISZ READER
04 01046 101001      MOV R,R,SKP
05 01047 010075      ISZ READER
06 01050 101103      MOVL R,R,SNK
07 01051 000755      JMP HSR
08 01052 020132      LDA R,PUNCH
09 01053 020075      LDA 1,READER
10 01054 030103      LDA 2,RPDIF
11 01055 100400      SUB R,1
12 01056 133404      AND 1,2,SZR
13 01057 000747      JMP HSR
14 01060 064612      DIAS 1,PTR
15 01061 000746      JMP HSR+1
16 01062 020106 HSR3: LDA R,C135525
17 01063 040076      STA R,RRAN
18 01064 002170      JMP #DISMISS
19
20
21
22
23 01065 020147 LSR1: LDA R,TINEY
24 01066 101005      MOV R,R,SNR
25 01067 000423      JMP R,3
26 01070 060210      NIOC TTI
27 01071 002170      JMP #DISMISS
28 01072 064512      DIAS 1,TTI
29 01073 063614      SKPBN TTI
30 01074 063410      SKPBN TTI
31 01075 063077      HALT
32 01076 030146      LDA 2,TIN
33 01077 010146      ISZ TIN
34 01100 101001      MOV R,R,SKP
35 01101 010146      ISZ TIN
36 01102 020070      LDA R,LSRH
37 01103 044070      STA 1,LSRH
38 01104 040071      STA R,LSRCH
39 01105 151004      MOV R,2,2,SZR
40 01106 101005      MOV R,R,SNR
41 01107 000407      JMP LSR1
42 01110 000112      JSR #CXRAND
43 01111 000072      RKAN
44 01112 020071      LDA 1,LSRCH
45 01113 100414      SURR R,1,SZR
46 01114 063077      HALT
47 01115 002170      JMP #DISMISS
48 01116 020106 LSR1: LDA R,C135525
49 01117 040072      STA R,KRAN
50 01120 002170      JMP #DISMISS
51
52
53
54

```

ITAPE LENGTH

IDON'T LET READER READ
TO MUCH TAPE

IREAD AND STOP

IRANDOM SETUP

IKEYBOARD

ISTOP READER
IKEYBOARD INTERRUPT

IFLAG CHECK

ICCHAR BUFFER
ITHE CHARACTER TO USE

IZERO CHARACTER

ICHECK DATA

IC(1)=RAD,C(0)=GOOD

```

A 0015 .MAIN
01
02 01121 027400 XRAND: LDA 1,00,3 ;GENERATE A 8 BIT
03 01122 020110 LDA 0,C177 ;EVEN PARITY RANDOM
04 01123 131120 MOVZL 1,2 INUMBER.
05 01124 151120 MOVZL 2,2
06 01125 133000 ADD 1,2
07 01126 024400 LDA 1,+2
08 01127 147000 ADD 2,1
09 01130 121300 MOVS 0,P
10 01131 000401 JMP ,XRAM
11 01132 123725 ,XRAM: ANDZS 1,P,SNR
12 01133 000707 JMP XRAND+1
13 01134 047400 STA 1,00,3
14 01135 111300 MOVS 0,2
15 01136 120000 ADC 1,1
16 01137 107000 ADD 0,1
17 01140 123400 AND 1,0,SZP
18 01141 000775 JMP ,+3
19 01142 101200 MOVH 0,P
20 01143 143300 ADPS 2,0
21 01144 001401 JMP 1,3
22
23 01145 000112 TY01: JSR 0,CYRAND ;TELETYPE OUTPUT
24 01146 000120 TRAN
25 01147 024101 LDA 1,LT
26 01150 125220 MOVZL 1,1 ;LEADER/2
27 01151 030121 LDA 2,TYPE
28 01152 132032 ADCZM 1,2,SZC
29 01153 000413 JMP TY01
30 01154 020100 LDA 0,C135525 ;INIT RANDOM
31 01155 040120 STA 0,TRAN
32 01156 125220 MOVZL 1,1
33 01157 034145 LDA 3,STT
34 01160 133405 AND 1,2,SNR
35 01161 175005 MOV 3,3,SNR
36 01162 000403 JMP ,+3
37 01163 060211 NIOS TTY ;SHUT DOWN
38 01164 002170 JMP 0DISMISS
39 01165 102400 SUB 0,0
40
41 01166 010121 TY01: ISZ TYPE
42 01167 101001 MOV 0,0,SKP
43 01170 010121 ISZ TYPE
44 01171 063411 SKPBN TTY
45 01172 063611 SKPDN TTY
46 01173 063077 HALT ;FLAG CHECK
47 01174 061111 DOAS 0,TTY
48 01175 063511 SKPBZ TTY
49 01176 063711 SKPDZ TTY
50 01177 063077 HALT ;FLAG CHECK
51 01200 002170 JMP 0DISMISS
52
53 01201 063414 CLOCK: SKPBN RTC ;CLOCK INTERRUPT
54 01202 063614 SKPDN RTC
55 01203 063077 HALT
56 01204 060114 NIOS RTC
57 01205 002170 JMP 0DISMISS

```

```

A 0015 .MAIN
01
02 01206 000112 ,PUN: JSR 0,CYRAND ;PUNCH INTERRUPT
03 01207 000153 ,PRAN
04 01210 024101 LDA 1,LT ;LEADER/TRAILER
05 01211 030154 LDA 2,,PUNCH
06 01212 132032 ADCZM 1,2,SZC
07 01213 000423 JMP ,PUN1
08 01214 020100 LDA 0,C135525
09 01215 040153 STA 0,,PRAN
10 01216 024110 LDA 1,C177
11 01217 034151 LDA 3,,STP
12 01220 133405 AND 1,2,SNR
13 01221 175005 MOV 3,3,SNR
14 01222 000413 JMP ,PUNX
15
16 01223 000253 NIOS ,PTP ;STOP PUNCHING
17 01224 063452 SKPBN ,PTR
18 01225 063752 SKPDZ ,PTR
19 01226 002170 JMP 0DISMISS
20 01227 020105 LDA 0,ACTION
21 01230 101300 MOVS 0,P
22 01231 101200 MOVH 0,P
23 01232 101202 MOVH 0,P,SZC
24 01233 000152 NIOS ,PTR ;RESTART READER
25 01234 002170 JMP 0DISMISS
26 01235 102400 ,PUNX: SUB 0,P ;PUNCH LEADER/TRAILER
27
28 01236 010154 ,PUN1: ISZ ,PUNCH ;CHARACTOR COUNTER
29 01237 101001 MOV 0,0,SKP
30 01240 010154 ISZ ,PUNCH
31 01241 063453 SKPBN ,PTP
32 01242 063653 SKPDN ,PTP
33 01243 063077 HALT ;PUNCH FLAG CHECK
34 01244 061153 DOAS 0,,PTP
35 01245 063553 SKPBZ ,PTP
36 01246 063753 SKPDZ ,PTP
37 01247 063077 HALT
38
39 01250 020154 LDA 0,,PUNCH ;WHEN PUNCHING AND
40 01251 024155 LDA 1,,READER ;READING IT MAY
41 01252 030103 LDA 2,,PDIF ;BE NECESSARY TO
42 01253 106400 SUB 0,1 ;RESTART THE READER
43 01254 133404 AND 1,2,SZC
44 01255 002170 JMP 0DISMISS ;EXIT INTERRUPT
45
46 01256 063452 SKPBN ,PTR
47 01257 063752 SKPDZ ,PTR
48 01260 002170 JMP 0DISMISS
49 01261 020105 LDA 0,ACTION
50 01262 101100 MOVL 0,0
51 01263 101100 MOVL 0,0
52 01264 101100 MOVL 0,0
53 01265 101100 MOVL 0,0
54 01266 101100 MOVL 0,0
55 01267 101100 MOVL 0,0
56 01270 101112 MOVL 0,P,0,SZC
57 01271 060152 NIOS ,PTH ;RESTART READER
58 01272 002170 JMP 0DISMISS

```

```

A 0017 .MAIN
01
02 01273 004552 .MSR: DIAS 1,.PTR ISECOND READER INTERRUPT
03 01274 020156 LDA 0,.HSRR
04 01275 044156 STA 1,.HSRR
05 01276 040157 STA 0,.HSRR
06 01277 030155 LDA 2,.READER
07 01300 101004 MOV 0,0,SZR
08 01301 151235 MOVZRN 2,2,SZR
09 01302 000425 JMP .HSR3 I FIRST CHAR OR ZERO DATA
10 01303 006112 JSR .CXRAND I GEN RANDOM CHAR
11 01304 000152 .KRN
12 01305 024157 LDA 1,.HSRR I CHECK DATA
13 01306 100414 SUB# 0,1,SZR I READER/PUNCH
14 01307 063077 HALT I SECOND DEVICES, C(0)=GOOD
15 01310 002170 JMP .DISMIS I C(1)=BAD
16
17 01311 020105 .HSR1: LDA 0,ACTION
18 01312 010155 ISZ .READER
19 01313 101001 MOV 0,0,SKP
20 01314 010155 ISZ .READER
21 01315 101103 MOVL 0,0,SNC
22 01316 000755 JMP .MSH
23 01317 020154 LDA 0,.PUNCH
24 01320 024155 LDA 1,.READER
25 01321 030103 LDA 2,.RDIF
26 01322 100400 SUB 0,1 I DONT LET READER READ TO MUCH
27 01323 133404 AND 1,2,SZR
28 01324 000747 JMP .MSR
29 01325 064652 DIAC 1,.PTR
30 01326 000746 JMP .MSH+1
31 01327 020106 .HSR3: LDA 0,C135525
32 01330 040152 STA 0,.KRN
33 01331 002170 JMP .DISMIS
34
35 01332 020100 .LSH: LDA 0,.TINEY I SECOND KEYBOARD
36 01333 101005 MOV 0,0,SZR
37 01334 000403 JMP .+3
38 01335 060250 NIOL .TTI
39 01336 002170 JMP .DISMIS I STOP SECOND READER
40 01337 064550 DIAS 1,.TTI
41 01340 063650 SKPDN .TTI
42 01341 063450 SKPDN .TTI
43 01342 063077 HALT
44 01343 030161 LDA 2,.TIN
45 01344 012161 ISZ .TIN
46 01345 101001 MOV 0,0,SKP
47 01346 010161 ISZ .TIN
48 01347 020162 LDA 0,.LSHR
49 01350 044162 STA 1,.LSRR I CHAR BUFFER
50 01351 040163 STA 0,.LSRR I THE CHARACTER TO USE
51 01352 151004 MOV 2,2,SZR
52 01353 101005 MOV 0,0,SNR
53 01354 000407 JMP .LSR1
54 01355 006112 JSR .CXRAND
55 01356 000164 .KRN
56

```

```

A 0018 .MAIN
01
02 01357 024163 LDA 1,.LSWCH
03 01360 100414 SUB# 0,1,SZR I SECOND TELETYPE
04 01361 063077 HALT I C(0)=GOOD, C(1)=BAD
05 01362 002170 JMP .DISMIS
06
07 01363 020100 .LSH1: LDA 0,C135525
08 01364 040164 STA 0,.KRN
09 01365 002170 JMP .DISMIS
10
11 01366 006112 .TYU: JSR .CXRAND I SECOND TTY (OUTPUT)
12 01367 000166 .TRAN
13 01370 024101 LDA 1,LT
14 01371 125220 MOVZRN 1,1
15 01372 030165 LDA 2,.TYPE
16 01373 132032 ADCZRN 1,2,SZC
17 01374 000413 JMP .TYU1
18 01375 020106 LDA 0,C135525
19 01376 040166 STA 0,.TRAN I INIT RANDOM
20 01377 125220 MOVZRN 1,1
21 01400 034167 LDA 3,.STT
22 01401 133405 AND 1,2,SNR
23 01402 175005 MOV 3,3,SNR
24 01403 000403 JMP .+3
25 01404 060251 NIOL .TTT I SHUT DOWN
26 01405 002170 JMP .DISMIS
27 01406 102400 SUB 0,0 I LEADER/TRAILER
28
29 01407 010165 .TYU1: ISZ .TYPE
30 01410 101001 MOV 0,0,SKP
31 01411 010165 ISZ .TYPE
32 01412 063451 SKPDN .TTT
33 01413 063651 SKPDN .TTT
34 01414 063077 HALT I FLAG CHECK
35 01415 061151 DIAS 0,.TTT
36 01416 063551 SKPHZ .TTT
37 01417 063751 SKPD7 .TTT
38 01420 063077 HALT I FLAG CHECK
39 01421 002170 JMP .DISMIS
40

```

A 0019 .MAIN

```

01
02      JA TEST OF STA,LOA
03      ILOAD C(1) FROM BUFFER, STORE C(1) IN BUFFER+1. THIS
04      IPROCEDURE
05      I WILL FILL THE BUFFER WITH THE VALUE OF THE FIRST WORD
06      IIN THE
07      IBUFFER. THE C(1) IS TESTED WHEN THE BUFFER HAS BEEN
08      IFILLED.
09      IBECAUSE THE LAST WORD IN THE BUFFER IS BASED ON THE
10      IPREVIOUS
11      IWORD ITS CORRECTNESS INSURES THAT THE ENTIRE BUFFER IS
12      ICORRECT.
13      IF THE C(1) IS IN ERROR (NOT ZERO) EXAMINE THE BUFFER
14
15 01422 030127 STA0:  LDA 2,BUFF      I FIRST LOCATION IN BUFFER
16 01423 034130      LDA 3,FIN        I LAST LOCATION IN BUFFER
17 01424 102470      SUB 0,P          IC(0)=0
18 01425 041000      STA 0,0,2        I VALUE OF FIRST BUFFER WORD
19
20 01426 025000 STA01: LDA 1,0,2      I WORD FROM BUFFER
21 01427 040001      STA 1,1,2        I PUT INTO BUFFER+1
22 01430 151400      INC 2,2          I MOVE TO NEXT BUFFER LOC
23 01431 150014      ADC 2,3,SZR      I TEST FOR END OF BUFFER
24 01432 000774      JMP ,=4         I MORE WORDS TO OUT PUT
25 01433 125014      MOV 1,1,SZR      I FULL BUFF! CHECK C(1)=0
26 01434 063077      HALT            I EXAMINE BUFFER FOR ERRORS
27 01435 101014      MOV 0,0,SZR      IC(0) WAS CLEARED AT STAR+2
28 01436 003077      HALT            I NOT OTHERWISE USED!
29
30      JA TEST OF STA,LOA
31      ITHIS TEST IS IDENTICAL TO THE PREVIOUS TEST EXCEPT THE
32      IFIRST
33      IWORD OF THE BUFFER IS SET TO ALL ONES (177777).
34
35 01437 030127 STA02: LDA 2,BUFF      I BUFF CONTAINS FIRST BUFFER LOC
36 01440 034130      LDA 3,FIN        I THE LAST LOCATION IN BUFFER
37 01441 102000      ADC 0,0          IC(0)=177777
38 01442 041000      STA 0,0,2        I VALUE OF FIRST BUFFER WORD
39
40 01443 025000 STA03: LDA 1,0,2      I WORD FROM BUFFER
41 01444 040001      STA 1,1,2        I PUT INTO BUFFER+1
42 01445 151400      INC 2,2          I ADVANCE TO NEXT LOCATION
43 01446 150014      ADC 2,3,SZR      I TEST FOR END OF BUFFER
44 01447 000774      JMP ,=4         I MORE TO FILL
45
46 01450 124014      COM 1,1,SZR      I CHECK C(1) FOR 177777.
47 01451 063077      HALT            I EXAMINE C(BUFFER) FOR ERROR
48 01452 100014      COM 0,0,SZR      IC(0) WAS MODIFIED DURING
49 01453 063077      HALT            I THIS TEST! ?
50

```

A 0020 .MAIN

```

01
02      IFILL AND CHECK BUFFER. A BUFFER OF ALTERNATING BITS
03
04
05 01454 030127 STA04: LDA 2,BUFF      I FIRST LOCATION OF THE BUFFER
06 01455 034130      LDA 3,FIN        I LAST LOCATION OF THE BUFFER
07 01456 020420      LDA 0,C02525
08 01457 104000      COM 0,1
09 01460 045000      STA 1,0,2        I BUFFER = 125252
10 01461 041001      STA 0,1,2        I BUFFER = 052525
11 01462 021000 STA05: LDA 0,0,2      I THIS PAIR OF WORDS
12 01463 151400      INC 2,2          I IS MOVED THROUGH THE
13 01464 150015      ADC 2,3,SZR      I BUFFER
14 01465 000774      JMP ,=4
15
16 01466 021000 STA06: LDA 0,0,2      I GET BUFFER'S LAST AND
17 01467 025377      LDA 1,-1,2      I LAST-1 WORDS
18 01470 123020      ADD 1,0
19 01471 100014      COM 0,0,SZR      I BUFFER SHOULD CONTAIN
20 01472 063077      HALT            I ALTERNATING WORDS
21 01473 101012      MOV 0,0,SZR      I EXAMINE BUFFER FOR ERRORS
22 01474 063077      HALT            I HOW DID CARRY CHANGE?
23 01475 101011      MOV 0,0,SKP      I SKIP OVER CONSTANT
24 01476 052525 C52525: 052525
25
26
27      IMAKE SURE THAT ALL LOCATIONS IN THE BUFFER EXIST.
28      ISTORE A ADDRESS PATTERN (EACH LOCATION CONTAINS ITS
29      IADDRESS).
30      IIF STA WERE ATTEMPTING TO STORE A 2400 IN LOCATION
31      I2400
32      IBUT INSTEAD STORED IN 2000, THE CONTENTS OF 2000 WOULD
33      ICONTAIN 2400. ONE SHOULD SUSPECT MA BIT 7 OR OTHER
34      IMEMORY ADDRESS DECODING.
35
36 01477 102400 STA10: SUB 0,0        IC(0) NOT USED THIS TEST
37 01500 034127      LDA 3,BUFF      I FIRST LOCATION OF BUFFER
38 01501 024130      LDA 1,FIN        I LAST LOCATION OF BUFFER
39
40 01502 055400 STA11: STA 3,0,3      I STORE THE ADDRESS AT ADDRESS
41 01503 031400      LDA 2,0,3      I LOAD C(2) WITH DATA JUST
42 01504 156414      SUB 2,3,SZR      I STORED. C(3)=DATA SENT,
43 01505 063077      HALT            IC(3)=ADDRESS,C(2)=RECEIVED
44 01506 175400      INC 3,3          I INC TO NEXT ADDR
45 01507 136414      SUB 1,3,SZR      I TEST FOR END OF BUFF
46 01510 000772      JMP STA11
47
48 01511 034127 STA12: LDA 3,BUFF      I CORE SHOULD BE FILLED WITH
49 01512 031400      LDA 2,0,3      I ADDRESS PATTERN. GET WORD
50 01513 156414      SUB 2,3,SZR      IC(2) SHOULD EQUAL C(3)
51 01514 063077      HALT            I LOOK FOR PATTERN OF ERRORS
52 01515 175400      INC 3,3          I GO TO NEXT ADDRESS
53 01516 136414      SUB 1,3,SZR      I TEST FOR END OF BUFFER
54 01517 000773      JMP STA12+1     I MORE TO GO
55
56 01520 101014 STA13: MOV 0,0,SZR    I CHANGE TO (JMP STA10) FOR LOOP
57 01521 063077      HALT            IC(0) NOT USED THIS TEST!
58

```

A 0021 .MAIN

```

01
02
03      !THIS IS A ADDRESS TEST IN WHICH EACH LOCATION IN THE
04      !BUFFER
05      !CONTAINS THE COMPLEMENT OF ITS ADDRESS. IF A FAILURE
06      !OCCURES
07      !EXAMINE THE BUFFER AND TRY TO FIND A PATTERN OF ERRORS
08      !LOA AND STA HAVE PREVIOUSLY WORKED WITH A SIMILAR
09      !PATTERN.
10
11 01522 034127 STA20: LDA 3,BUFF      !C(3)=FIRST BUFFER LOCATION
12 01523 024130      LDA 1,FIN        !C(1)=LAST BUFFER LOCATION
13
14 01524 100005 STA21: COM 3,0,SNR    !THE CONTENTS OF 3 SHOULD
15 01525 063077      HALT            !ALLWAYS PRODUCE NOT ZERO
16 01526 041400      STA 0,0,3        !STORE LOGICAL COMPLEMENT
17 01527 031400      LDA 2,0,3        !OF ADDRESS IN ADDRESS
18 01530 142414      SUR# 2,0,SZR    !C(2) SHOULD BE SAME AS
19 01531 063077      HALT            !C(0) WHICH WAS STORED.
20 01532 175400      INC 3,3          !INC TO NEXT ADDRESS
21 01533 136414      SUR# 1,3,SZR    !TEST FOR END OF BUFFER
22 01534 000770      JMP STA21       !NOT YET END OF BUFFER
23
24
25 01535 034127 STA22: LDA 3,BUFF      !RESET C(3) TO FIRST LOCATION
26 01536 031400      LDA 2,0,3        !IN THE BUFFER. WORD FROM
27 01537 140000      COM 2,0          !BUFFER SHOULD BE COMP OF
28 01540 116414      SUR# 1,3,SZR    !ADDRESS IN 3. C(2)=BUFFER WD
29 01541 063077      HALT            !C(0)=ITS COMP. C(3)=ADDRESS
30 01542 175400      INC 3,3          !INCREMENT TO NEXT ADDRESS
31 01543 136414      SUR# 1,3,SZR    !TEST FOR END OF BUFFER
32 01544 000772      JMP STA22+1     !MORE WORDS TO CHECK
33
34 01545 000401      JMP ,+1          !REPLACE WITH (JMP STA20) TO
35                                     !LOOP
36

```

A 0022 .MAIN

```

01
02      !A TEST OF JSR
03      !THIS PROGRAM WILL EXECUTE A JSR INSTRUCTION TO EVERY
04      !LOCATION
05      !IN THE BUFFER. EACH LOCATION IN THE BUFFER CONTAINS
06      !A(JSR 1,3)
07      !INSTRUCTION. THE JSR RETURN FROM THE BUFFER WILL BE TO
08      !THE
09      !CALLING (JSR TO BUFFER) +2. THE RETURN FROM THE BUFFER
10      !WILL
11      !ALSO STORE THE PC (PROGRAM COUNTER) OF THE BUFFER
12      !INSTRUCTION+1
13      !IN C(3). C(2) POINTS TO THE BUFFER LOCATION TO BE
14      !EXECUTED.
15      !C(3)-1 INDICATES WHICH LOCATION IN THE BUFFER WAS
16      !EXECUTED. THE
17      !C(3) SHOULD BE ONE GREATER THAN THE BUFFER POINTER IN
18      !C(2).
19
20 01546 102520 JSR10: SUB#L 0,0      !LOOP ON ERR SWITCH
21 01547 034127      LDA 3,BUFF      !C(3)=FIRST BUFFER LOCATION
22 01550 030421      LDA 2,CJSR10    !CONSTANT (JSR 1,3)
23 01551 024130      LDA 1,FIN        !C(1)=FINAL BUFFER LOCATION
24 01552 051400      STA 2,0,3        !FILL THE BUFFER
25 01553 175400      INC 3,3          !WITH JSR INSTRUCTIONS
26 01554 166414      SUR# 3,1,SZR    !TEST FOR END OF THE BUFFER
27 01555 020775      JMP ,+3         !MORE WORDS TO STORE
28
29 01556 030127 JSR11: LDA 2,BUFF      !C(2)=FIRST BUFFER LOCATION
30 01557 005000      JSR 0,2          !GO TO THE BUFFER! C(3)=
31 01560 063077      HALT            !RETURN ADDR. FAIL TO PC XFER
32 01561 156014      ADD# 2,3,SZR    !CHECK NEW C(3) FROM JSR IN
33                                     !BUFF
34 01562 063077      HALT            !C(3)SHOULD=C(2)+1. WENT WRONG
35 01563 156014      ADD# 2,3,SZR
36 01564 102400      SUB 0,0
37 01565 113000      ADD 0,2
38 01566 146414      SUR# 2,1,SZR    !NEXT BUFF LOC.TEST FOR END
39 01567 000770      JMP JSR11+1    !MORE LOCATIONS IN BUFF
40 01570 101011      MOV# 0,0,SKP    !SKIP OVER CONSTANTS
41 01571 025401 CJSR10: JSR 1,3
42

```

A 0023 .MAIN

```

01
02
03      JA TEST OF JMP
04      !THIS PROGRAM WILL EXECUTE A JMP TO EVERY LOCATION IN
05      !THE BUFFER.
06      !THE BUFFER CONTAINS A (JSR 1,3) INSTRUCTION IN EACH
07      !LOCATION.
08      !THE VALUE IN C(3) IS SET SUCH THAT THE JSR IN THE
09      !BUFFER WILL
10      !RETURN TO THE MAIN PROGRAM, THE PROGRAM THEN CHECKS
11      !THE PC
12      !PROGRAM COUNTER) STORED BY THE BUFFER'S JSR
13      !INSTRUCTION. THIS
14      !PC SHOULD BE ONE GREATER THAN THE LOCATION JUMPED TO.
15
16 01572 125520 JMP101 SURZL 1,1      !LOOP ON ERR SWIT
17 01573 020130 LDA 0,FIN          !FINAL ADDRESS OF BUFFER
18 01574 030127 LDA 2,BUFF        !FIRST ADDRESS OF BUFFER
19 01575 034413 LDA 3,CJMP11       !C(3)=A RETURN TO THE PROG
20 01576 001000 JMP 0,2          !GO TO A LOCATION IN BUFFER
21 01577 063077 HALT              !NO JMP? INDEX+1 IN BUFFER?
22 01600 156014 JMP11: ADC# 2,3,SZR !CHECK PC STORED BY BUFFER JSR
23 01601 063077 HALT              !C(2)=JMP ADDRESS,C(3)=JSR PC
24 01602 156014 ADC# 2,3,SZR
25 01603 126400 SUR 1,1
26 01604 133000 ADD 1,2
27 01605 142414 SUR# 2,0,SZR      !TEST FOR END OF THE BUFFER
28 01606 000767 JMP JMP10+3          !NOT YET END OF BUFFER
29 01607 101011 MOV# 0,0,SKP      !SKIP OVER THE CONSTANT
30 01610 001577 CJMP11: JMP11-1    !CONSTANT

```

A 0124 .MAIN

```

01
02      JA TEST OF POSITIVE DISPLACEMENT USING THE "LDA"
03      !INSTRUCTION
04      !THE BUFFER IS FILLED WITH A ADDRESS PATTERN
05      !C(ADDRESS)=ADDRESS).
06      !C(3) POINTS TO A LOCATION IN THE BUFFER,, A "LDA"
07      !INSTRUCTION
08      !WITH A POSITIVE DISPLACEMENT THEN REFFERANCES THE
09      !BUFFER VIA
10      !INDEX REGISTER 3. THE EFFECTIVE ADDRESS IS THE SUM OF
11      !THE INDEX
12      !REGISTER VALUE AND THE DISPLACEMENT. C(2)=EFFECTIVE
13      !ADDRESS
14      !OBTAINED,C(1)=CORRECT EFFECTIVE ADDRESS,C(3)=INDEXING
15      !VALUE
16      !C(POSX2)=FAILING INSTRUCTION--DISPLACEMENT IN BITS
17      !8-15
18
19 01611 034127 POSX: LDA 3,BUFF    !FIRST BUFFER LOCATION
20 01612 024130 LDA 1,FIN          !LAST BUFFER LOCATION
21 01613 055400 STA 3,0,3      !STORE A ADDRESS PATTERN
22 01614 175400 INC 3,3          !IN THE BUFFER
23 01615 136414 SUR# 1,3,SZR      !C(BUFFER)=ADDRESS OF BUFF
24 01616 000775 JMP POSX+2        !EXAMPLE: C(3015)=3015
25 01617 030426 LDA 2,CLOAPX      !CONSTANT (LDA 2,0,3) INST
26 01620 050411 STA 2,POSX2      !INITIALIZE THE INSTRUCTION
27
28 01621 020133 POSX1: LDA 0,C3     !ADD 3 TO THE INDEX
29                                !DISPLACEMENT=
30 01622 024407 LDA 1,POSX2        !MENT OF THE LDA INSTRUCTION
31 01623 107000 ADD 0,1
32 01624 044405 STA 1,POSX2
33 01625 020421 LDA 0,POSFIN      !END TEST FOR DISPLACEMENT
34 01626 106032 ANDZ# 0,1,SZC
35 01627 000420 JMP POSNO
36 01630 034127 LDA 3,BUFF
37 01631 000000 POSX2: 0
38
39 01632 024777 LDA 1,-1
40 01633 020135 LDA 0,C377
41 01634 107400 AND 0,1
42 01635 107000 ADD 3,1
43 01636 146414 SUB# 2,1,SZR
44 01637 063077 HALT
45 01640 020131 LDA 0,FIN200
46 01641 175400 INC 3,3
47 01642 116414 SUR# 0,3,SZR
48 01643 000766 JMP POSX2
49 01644 000755 JMP POSX1        !NEXT DISPLACEMENT
50
51 01645 031400 CLOAPX: LDA 2,0,3  !CONSTANT FIRST DISPLACEMENT
52 01646 031577 POSFIN: LDA 2,177,3 !CONSTANT FINAL "
53 01647 000401 POSNO: JMP ,+1     !REPLACE WITH JMP TO LOOP
54

```

```

02      J A TEST OF NEGATIVE DISPLACEMENT USING "LOA"
03      IF THE BUFFER IS FILLED WITH A ADDRESS PATTERN
04      IF C(ADDRESS)=ADDRESS), A "LOA" INSTRUCTION
05      WITH A NEGATIVE DISPLACEMENT THEN REFFERANCES
06      IF THE BUFFER VIA INDEX REGISTER 2, THE EFFECTIVE
07      ADDRESS IS THE VALUE OF THE INDEX REGISTER
08      MINUS THE DISPLACEMENT VALUE, C(3)=EFFECTIVE
09      ADDRESS OBTAINED,C(1)=CORRECT EFFECTIVE ADDRESS,
10      IC(2)=INDEXING VALUE,C(NEGX2)=FAILING INSTRUCTION
11      DISPLACEMENT IN BITS R-15.
12
13 01650 034127 NEGX: LDA 3,BUFF      IFIRST BUFFER LOCATION
14 01651 02413P      LDA 1,FIN      IFINAL BUFFER LOCATION
15 01652 05540A      STA 3,0,3      IFILL THE BUFFER WITH A
16 01653 1754AA      INC 3,3      IADDRESS PATTERN,FOR
17 01654 13641A      SUB# 1,3,57R IFXAMPLE: C(3417)=3417
18 01655 007775      JMP NEGX2+1 IC(4150)=4150
19 01656 030431      LDA 2,NEGFIN    IINITIALIZE THE LOA INST
20 01657 050412      STA 2,NEGX2    ITO (LDA 3,377,2)
21
22 01660 020133 NEGX1: LDA 0,C3      ISUBTRACT (3) FROM THE LDA'IS
23 01661 02441P      LDA 1,NEGX2    IDISPLACEMENT
24 01662 10640A      SUB 0,1      I
25 01663 04440A      STA 1,NEGX2    IPUT IT BACK IN CORE
26 01664 02042A      LDA 0,NEGXX    ITEST FOR A POSITIVE
27 01665 106033      ADDZ# 0,1,5NC   IDISPLACEMENT
28 01666 000423      JMP NEGND    IEND, GO TO NEXT TEST
29 01667 000401      JMP NEGX2-1
30
31 01670 030132      LDA 2,BUFF200    IFIRST BUFFER LOCATION+200
32 01671 00000A NEGX2: 0      IRECOMES: LDA 3,-N,2 11
33 01672 024777      LDA 1,-,1      IGET DISPLACEMENT BITS
34 01673 020135      LDA 0,C377     IASK THE INSTRUCTION PART
35 01674 10740P      AND 0,1
36 01675 10000A      COM 0,0
37 01676 10700P      ADD 0,1
38 01677 14700P      ADD 2,1
39 01700 13641A      SUB# 1,3,57R   IC(1)=THE EFFECTIVE ADDRESS
40 01701 063077      HALT          IC(2)=THE INDEXING VALUE
41 01702 02013P      LDA 0,FIN      IC(3)=THE ADDRESS OBTAINED
42 01703 15140A      INC 0,2      IJUST AS NEGX2 FAILED
43 01704 11261A      SUB# 0,2,57R   IUSE THIS DISPLACEMENT
44 01705 000764      JMP NEGX2     IWITH OTHER LOCATIONS IN THE
45 01706 020752      JMP NEGX1     IBUFFER
46                                     IEND BUFFER END
47                                     IEND, TRY NEW DISPLACEMENT
48 01707 035377 NEGFIN: LDA 3,-1,2   ICONSTANT: DISPLACEMENT=1
49 01710 03520A NEGXX:  LDA 3,-20,2  ICONSTANT: DISPLACEMENT OF -177
50 01711 000401 NEGND:  JMP .+1      IREPLACE WITH JMP TO LOOP TEST
51

```

```

01
02      JA TEST OF POSITIVE DISPLACEMENT USING THE "JMP".
03      IF THE BUFFER IS FILLED WITH (JSR 0,3) INSTRUCTIONS.
04      ITMUS WHEN A JSR IN THE BUFFER IS EXECUTED C(3) WILL
05      BE SET TO THE BUFFER LOCATION+1. A "JMP" INSTRUCTION
06      WITH A POSITIVE DISPLACEMENT GOES TO THE BUFFER VIA
07      INDEX REGISTER 2. THE JSR INSTRUCTION IN THE BUFFER
08      WILL RETURN TO THE PROGRAM SAVING ITS LOCATION +1 IN
09      IC(3). THE EFFECTIVE ADDRESS OF THE "JMP" IS COMPARED
10      WITH C(3). C(3)-1=LOCATION JUMPED TO. C(2)=INDEX
11      VALUE,C(1)=EFFECTIVE ADDRESS,C(POJSJ3)=FAILING INST-
12      RUTION,DISPLACEMENT IN BITS 8-15.
13
14 17112 034127 POSJ1: LDA 3,BUFF      IFIRST LOCATION IN BUFFER
15 17113 024134      LDA 1,FIN      IFINAL LOCATION IN BUFFER
16 17114 030435      LDA 2,POSJX     IFILL THE BUFFER WITH(JSR 0,3)
17 17115 051400      STA 2,P,3      INSTRUCTIONS
18 17116 175444      INC 3,3
19 17117 165414      SUM# 3,1,SZR    IFTEST FOR END OF BUFFER
20 17120 030774      JMP POSJ+2     IFMORE LOCATIONS TO FILL
21 17121 020427      LDA 0,POSJZ    IFINITIALIZE THE JUMP
22 17122 040416      STA 0,POSJ3    IFWITH POSITIVE DISPLACEMENT
23
24 17123 020134 POSJ1: LDA 0,C5      INCREASE POSITIVE DISPLACE
25 17124 024414      LDA 1,POSJ3    BY FIVE
26 17125 107000      ADD 0,1
27 17126 044412      STA 1,POSJ3    IFNEW "JMP" INSTRUCTION
28 17127 020403      LDA 0,POSJF    IFANY MORE + DISPLACEMENTS
29 17130 100030      ADC#4 0,1,SZC   IFGO
30 17131 020423      JMP POSJND     IFGO GO TO NEXT TEST
31
32 17132 030127 POSJ2: LDA 2,BUFF      IFIRST LOCATION IN BUFFER
33 17133 020130      LDA 0,FIN200    IFINAL LOCATION IN BUFFER
34 17134 151400      INC 2,2      IFPOINT TO NEXT BUFFER
35 17135 142415      SUM# 2,0,SNR    IFLOCATION. IS IT THE LAST?
36 17136 020765      JMP POSJ1     IFYES TRY NEW DISPLACEMENT
37 17137 034414      LDA 3,POSJY    IFSETUP C(3) FOR BUFFER RFT
38 17140 000000 POSJ3: R      IF"THE" JMP INSTRUCTION
39 17141 024777      LDA 1,0,-1
40 17142 020135      LDA 0,C377     IFCALCULATE THE EFFECTIVE
41 17143 107400      AND 0,1      ADDRESS OF THIS JUMP
42 17144 147000      ADD 2,1      IC(INDEX)+DISPLACEMENT
43 17145 130014      ADC# 1,3,S7H   IC(1)=EFFECTIVE ADDRESS,C(3)
44 17146 030777      HALT          IFJSR PC,C(2)=INDEX VALUE
45 17147 020764      JMP POSJ2+1    IFGO AGAIN AT NEXT LOC
46
47 17150 001000 POSJZ: JMP 0,2      IFCONSTANTS
48 17151 020600 POSJX: JSR 0,3      IF"
49 17152 001177 POSJF: JMP 177,2
50 17153 001741 POSJY: POSJ3+1
51 17154 000401 POSJND: JMP .+1     IFREPLACE WITH JMP TO LOOP
52

```

A 0027 .MAIN

```

01
02
03
04      ;A TEST OF NEGATIVE DISPLACEMENT USING "JSR"
05      ;THE BUFFER IS FILLED WITH (JSR 0,3) INSTRUCTIONS.
06      ;THE BUFFER IS ENTERED WITH A JSR VIA C(3) WITH A
07      ;NEGATIVE DISPLACEMENT IN BITS 8-15, EFFECTIVE
08      ;ADDRESS CALCULATION SHOULD SUBTRACT THE DIS-
09      ;PLACEMENT FROM THE VALUE IN INDEX REGISTER 3.
10      ;THE JSR IN THE BUFFER WILL RETURN TO THE PROGRAM
11      ;SAVING ITS LOCATION +1 IN C(3). COMPARISON OF
12      ;CALCULATED EFFECTIVE ADDRESS AND C(3) DETERMIN
13      ;IF THE BUFFER WAS ENTERED CORRECTLY.
14
15 01755 030432 NEGJ1: LDA 2,NEGJ1      ;C(2)=CONSTANT (JSR =1,2)
16 01756 050415      STA 2,NEGJ3      ;STORE "THE" JSR
17
18 01757 020133 NEGJ1: LDA 0,C3        ;DECREMENT THE VALUE OF
19 01760 024413      LDA 1,NEGJ3      ;THE DISPLACEMENT BY 3.
20 01761 106400      SUB 2,1         ;EACH DISPLACEMENT IS TEST-
21 01762 044411      STA 1,NEGJ3      ;ED WITH THE BUFFER
22 01763 020423      LDA 0,NEGJX     ;
23 01764 106052      ADDC 0,1,SZC     ;DISPLACEMENT TEST
24 01765 000423      JMP NEGJ0       ;GO TO NEXT TEST
25
26 01766 030132 NEGJ2: LDA 2,BUF200    ;FIRST ADDRESS+200 IN BUFFER
27 01767 020130      LDA 0,FIN       ;THIS ROUTINE SETS CARRY!!!
28 01770 151440      INCC 2,2        ;ITS TESTED LATER! INC BUFF
29 01771 142435      SUB2 2,0,SNW    ;ADDRESS AND TEST FOR END
30 01772 000765      JMP NEGJ1       ;END,GET NEW "JSR"
31
32 01773 000000 NEGJ3: 0              ;GO TO THE BUFFER HERE!
33 01774 024777      LDA 1,-1        ;OBTAIN THE EFFECTIVE
34 01775 020135      LDA 0,C377      ;ADDRESS IN C(1)
35 01776 107403      AND 0,1,SNW    ;THE CARRY SHOULD NOT HAVE
36 01777 063077      HALT           ;CHANGED! SEE ABOVE
37 02000 100000      COM 0,0        ;
38 02001 107000      ADD 2,1         ;
39 02002 147000      ADD 2,1         ;CHECK EFFECTIVE ADDRESS
40 02003 136014      ADDC 1,3,SZR    ;CHECK EFFECTIVE ADDRESS
41 02004 063077      HALT           ;WITH PC OF BUFFER JSR IN
42 02005 000762      JMP NEGJ2+1    ;C(3),C(2)=INDEX VALUE
43
44 02006 005200 NEGJX: JSR -200,2     ;CONSTANT
45 02007 005377 NEGJY: JSR -1,2     ;CONSTANT
46 02010 000401 NEGJD: JMP .+1      ;REPLACE WITH JMP TO LOOP
47
48

```

A 0028 .MAIN

```

01
02
03
04      ;TEST THAT PC WILL COUNT THROUGH THE BUFFER.
05
06 02011 034127 PINC1: LDA 3,BUFF     ;
07 02012 024130      LDA 1,FIN       ;
08 02013 020415      LDA 0,CINC      ;FILL THE BUFFER
09 02014 041400 PINC1: STA 0,0,3     ;WITH INC INSTRUCTIONS
10 02015 175400      INC 3,3         ;
11 02016 136414      SUB# 1,3,SZR    ;
12 02017 000775      JMP PINC1      ;
13 02020 020411      LDA 0,CPINC     ;
14 02021 041400      STA 0,0,3       ;PUT JMP IN LAST LOC
15 02022 030410 PINC2: LDA 2,PINCR   ;C(2)=RETURN TO PROG
16 02023 034127      LDA 3,BUFF     ;
17 02024 001400      JMP 0,3        ;GO TO BUFFER
18
19 02025 136414 PINC3: SUB# 1,3,SZR   ;BUFFER INST SHOULD COUNT
20 02026 063077      HALT           ;C(1)=CORRECT
21 02027 000404      JMP .+4        ;C(3)=BUFF+INC RESULT
22 02030 175400 CINC: INC 3,3        ;
23 02031 001000 CPINC: JMP 0,2      ;
24 02032 002025 PINCR: PINC3       ;
25 02033 101000      MOV 0,0        ;CHANGE TO LOOP TEST

```

A 0020 .MAIN

01
02
03
04
05
06
07 02034 102520 ISZ01 SUBZL 0,0 IC(0)=1, LOOP ON ERROR SWITCH
08 02035 024135 LDA 1,C377 INUMBER OF ITERATIONS
09 02036 152000 ADC 2,2 IC(2) NOT USED THIS TEST
10 02037 176400 SUB 3,3 ITHE VALUE STORED (0)
11 02040 054122 ISZ11 STA 3,TEM ISTORE FOR INCREMENT
12 02041 060377 NIOP CPU ISYNC AT 7=A74
13 02042 010122 ISZ TEM IINC A ZEROS WORD
14 02043 000404 JMP ,+4 IJMP PAST THE ERROR
15 02044 020122 LDA 0,TEM ISZ SKIPPED C(0)=RESULT
16 02045 063077 HALT IOF ISZ, CORRECT VALUE= +1
17 02046 102400 SUB 0,0 ILOOP ON ERR SWITCH
18 02047 106404 SUB 0,1,SNR ITERATE THE PROGRAM
19 02050 000770 JMP ISZ1
20 02051 150014 COMW 2,2,SNR IC(2) MODIFIED!! CHECK ISZ
21 02052 063077 HALT IREPLACE TO LOOP TEST

22
23
24
25
26
27
28
29 02053 102520 ISZ21 SUBZL 0,0 IC(0)=1, LOOP ON ERROR SWITCH
30 02054 024135 LDA 1,C377 IC(1)=NUMBER OF ITERATIONS
31 02055 152400 SUB 2,2 IC(2) NOT USED THIS TEST
32 02056 176120 ADCZL 3,3 IC(3)=+2 (177776)
33 02057 054122 ISZ31 STA 3,TEM ISTORE AND THEN INCREMENT IT
34 02060 060377 NIOP CPU ISYNC AT 7=A74
35 02061 010122 ISZ TEM ISZ SHOULD NOT SKIP
36 02062 000404 JMP ,+4 IJMP PAST THE ERROR
37 02063 020122 LDA 0,TEM IC(0)=ISZ RESULT
38 02064 063077 HALT IIT SHOULD BE 177777
39 02065 102401 SUB 0,0,SNR ISZ LOGIC FAIL
40 02066 106404 SUB 0,1,SNR ITERATION COUNTER
41 02067 000770 JMP ISZ3
42 02070 151014 MOVW 2,2,SNR IC(2) MODIFIED !!
43 02071 063077 HALT ICHANGE MOV TO LOOP TEST

44

A 0030 .MAIN

01
02
03
04
05
06
07 02072 102520 ISZ41 SUBZL 0,0 IC(0)=1, LOOP ON ERROR SWITCH
08 02073 126620 SUBZL 1,1 IC(1)=100000, ITERATION COUNT
09 02074 152000 ADC 2,2 IC(2)=177777
10 02075 050122 ISZ51 STA 2,TEM ISTORE THE (-1)
11 02076 060377 NIOP CPU ISYNC AT 7=A74
12 02077 010122 ISZ TEM IINC IT SHOULD SKIP
13 02100 000404 JMP ISZ6 IFAIL TO SKIP, ISZ FAIL
14 02101 107004 ADD 0,1,SNR ITERATION COUNTER
15 02102 000773 JMP ISZ5 IC=CONTINUE TEST
16 02103 000405 JMP ISZ7 IEND OF THIS TEST
17 02104 020122 ISZ61 LDA 0,TEM IC(0)=RESULT OF FAIL ISZ
18 02105 063077 HALT ICORRECT VALUE=0
19 02106 102400 SUB 0,0 ICLEAR C(0) TO LOOP
20 02107 000766 JMP ISZ5 ION ERROR FOREVER
21 02110 101234 ISZ71 MOVZRN 0,0,SNR IREPLACE TO LOOP TEST
22 02111 063077 HALT IC(0) SHOULD BE +1
23 02112 000401 JMP ISZ10

24
25
26
27
28
29
30 02113 152520 ISZ101 SUBZL 2,2 IC(2)=1
31 02114 102020 SUBZL 0,0 IC(0)=100000
32 02115 143005 ADD 2,0,SNR IUPDATE THE COUNT BY 1
33
34 02116 000423 JMP ISZ13 IISZ FAIL TO SKIP ON 177777
35 02117 040122 STA 0,TEM ISTORE THE VALUE FOR ISZ
36 02120 060377 NIOP CPU ISYNC A 7=A74
37 02121 010122 ISZ TEM I"THE" ISZ
38 02122 101011 MOVW 0,0,SNR ISZ SHOULD SKIP AT END OF TEST
39 02123 000411 JMP ISZ12 IONLY
40 02124 024122 LDA 1,TEM IODES VALUE OF TEM=COUNT+1
41 02125 100015 ADCW 0,1,SNR IIN C(0)
42 02126 000767 JMP ISZ10+2 IYES KEEP CHECKING
43 02127 000401 JMP ISZ11
44 02130 115400 ISZ111 INC 0,3 IC(1)=ISZ COUNT, C(3)=CORRECT
45 02131 063077 HALT IISZ INCREMENT LOGIC FAILED
46 02132 152400 SUB 2,2 ICLEAR C(2) TO LOOP ON ERROR
47 02133 000762 JMP ISZ10+2
48 02134 115405 ISZ121 INC 0,3,SNR IISZ SKIPPED WAS THAT OK?
49 02135 000410 JMP ISZ14 IYES, EXIT THIS TEST
50 02136 024122 LDA 1,TEM IC(1)=ISZ COUNT, C(3)=INC COUNT
51 02137 063077 HALT IC(1) SHOULD = C(3), SKIP OR NOT
52 02140 000772 JMP ISZ12-2 ICLEAR COUNT AND LOOP ERROR
53 02141 102000 ISZ131 ADC 0,0 IISZ FAIL TO SKIP ON
54 02142 024122 LDA 1,TEM IC(TEM)=+1, C(1)=ISZ COUNT
55 02143 063077 HALT ILOOP ON ERROR
56 02144 000766 JMP ISZ12-2
57 02145 000401 ISZ141 JMP ,+1 ICHANGE TO LOOP TEST

A 0031 ,MAIN

```

01
02
03      ;A TEST OF ISZ
04      ;FIVE CONSECUTIVE ISZ INSTRUCTIONS TO LOCATION "TEM" ARE
05      ;EXECUTED, THE VALUE IN C(TEM) SHOULD EQUAL C(3)
06      ;THE NUMBERS ARE ARRANGED SUCH THAT NO ISZ INSTRUCTION
07      ;SHOULD SKIP, SHOULD THE TEST FAIL PRESSING CONTINUE
08      ;WILL PLACE THE PROGRAM IN A FAILING LOOP UNTILL
09      ;RESTARTED.
10
11 02146 030134 ISZ201 LDA 2,C5      ;C(2)=ERROR SWITCH
12 02147 102400 SUB 0,0      ;START WITH C(0)=0
13 02150 034134 LDA 3,C5      ;C(3)=EXPECTED RESULT OF
14 02151 117022 ADDZ 0,3,SZC    ;ISZ, ALSO TEST FOR END
15 02152 000417 JMP ISZ22    ;GO TO NEXT TEST
16 02153 040122 ISZ211 STA 0,TEM    ;STORE THE VALUE TO BE INC
17 02154 000377 NIOP CPU      ;SYNC AT 7-A74
18 02155 010122 ISZ TEM        ;DO IT 5 TIMES
19 02156 010122 ISZ TEM
20 02157 010122 ISZ TEM
21 02158 010122 ISZ TEM
22 02159 010122 ISZ TEM
23 02160 024122 LDA 1,TEM        ;C(1)=RESULT OF ISZ+5
24 02161 130414 SUB# 1,3,SZR    ;C(3)=CORRECT RESULT
25 02162 003077 HALT          ;ISZ FAILED
26 02163 130414 SUB# 1,3,SZR    ;IF A ERROR OCCURED
27 02164 152400 SUB 2,2      ;CLEAR C(0) TO LOOP ERR
28 02165 143000 ADD 2,0      ;UPDATE FOR NEXT GO
29 02166 000750 JMP ISZ2F+2    ;ROUND
30 02167 000750 JMP ISZ2F+2    ;CHANGE TO LOOP TEST
31
32      ;EACH WORD IN THE BUFFER IS SET TO -1 (177777).
33      ;ISZ THEN INCREMENTS IT TO ZERO AND SKIPS, THE PROGRAM
34      ;WILL LOOP ON ERROR IF CONTINUE IS PRESSED
35
36 02172 102520 ISZ301 SUBZL 0,0    ;C(0)=1, THE LOOP ON ERR SWIT
37 02173 024130 LDA 1,FIN        ;C(1)=FINAL BUFFER ADDRESS
38 02174 030127 LDA 2,BUFF      ;C(2)=FIRST BUFFER ADDRESS
39 02175 170000 ADC 3,3        ;C(3)=177777 (-1)
40 02176 055000 ISZ311 STA 3,0,2    ;STORE THE (-1) IN BUFFER
41 02177 000377 NIOP CPU        ;SYNC AT 7-A74
42 02200 011000 ISZ 0,2        ;INCREMENTATION SHOULD SKIP
43 02201 000412 JMP ISZ32+2    ;ISZ FAIL TO SKIP
44 02202 035000 LDA 3,0,2      ;DOES BUFFER=0?
45 02203 175014 MOV# 3,3,SZR    ;IF NOT ISZ FAIL TO COUNT
46 02204 000405 JMP ISZ32    ;CORRECT, BUT SKIPPED
47 02205 113000 ADD 0,2        ;INC TO NEXT BUFFER LOC
48 02206 132414 SUB# 1,2,SZR    ;AND TEST FOR END
49 02207 000766 JMP ISZ31-1    ;END GO NEXT TEST
50 02210 000407 JMP ISZ33      ;ISZ SKIPPED BUT STORED NG
51 02211 003077 ISZ321 HALT      ;C(3)=ISZ RESULT, C(2)=OK
52 02212 000403 JMP ,+3        ;C(3)=ADDRESS OF ERR
53 02213 035000 LDA 3,0,2      ;ISZ FAIL TO SKIP, C(3)=
54 02214 003077 HALT          ;ISZ RESULT, C(2)=EFF ADDRESS
55 02215 102400 SUB 0,0        ;LOOP ON THE ERROR
56 02216 000757 JMP ISZ31-1    ;CHANGE TO LOOP TEST
57 02217 000401 ISZ331 JMP ,+1
58

```

A 0032 ,MAIN

```

01
02
03      ;ANOTHER TEST OF ISZ
04      ;THIS TEST WILL ADD ONE TO EACH LOCATION OF A BUFFER
05      ;THAT IS PRESET TO ONES, THE LAST LOCATION IN THE
06      ;BUFFER IS SET TO CONTAIN ZEROS, THUS PREVENTING
07      ;A ISZ SKIP ON THAT LOCATION, A ISZ INSTRUCTION WILL
08      ;PREFERANCE THE BUFFER VIA INDEX REGISTER 2, IF THE
09      ;ISZ INSTRUCTION SKIPS AC2 IS INCREMENTED AND THE
10      ;SEQUENCE REPEATS, WHEN THE ISZ FAILS TO SKIP THE
11      ;BUFFER POINTER IS CHECKED, IF IT IS EQUAL TO THE END
12      ;OF THE BU
13      ;BUFFER THE TEST IS SUCCESSFUL, IF HOWEVER IT IS NOT
14      ;THE BUFFER END THE PROGRAM WILL HALT, IF A ERROR
15      ;OCCURES THE ENTIRE TEST WILL BE REPEATED UNTILL THE
16      ;PROGRAM IS RESTARTED.
17
18 02220 126400 ISZ401 SUB 1,1      ;LOOP ON ERR SWITCH
19 02221 102000 ADC 0,0        ;FINAL BUFFER ADDRESS
20 02222 034130 LDA 3,FIN      ;FIRST BUFFER ADDRESS
21 02223 030127 LDA 2,BUFF      ;FIRST BUFFER ADDRESS
22 02224 041000 STA 0,0,2      ;FILL THE BUFFER WITH
23 02225 151400 INC 2,2        ;(-1)
24 02226 156414 SUB# 2,3,SZR    ;EXCEPT THE LAST
25 02227 000775 JMP ,+3        ;WHICH IS FILLED WITH
26 02228 045400 STA 1,0,3      ;ZEROS
27
28 02231 030127 ISZ411 LDA 2,BUFF    ;C(2) POINTS TO BUFFER
29 02232 000377 NIOP CPU        ;SYNC AT 7-A74
30 02233 011000 ISZ 0,2        ;ISZ SHOULD SKIP EVERY
31 02234 000403 JMP ISZ42      ;TIME EXCEPT LAST
32 02235 151400 INC 2,2        ;POINT NEXT LOCATION
33 02236 000774 JMP ISZ41+1
34
35 02237 021000 ISZ421 LDA 0,0,2    ;C(0)=RESULT OF ISZ
36 02240 156414 SUB# 2,3,SZR    ;SHOULD BE A 0
37 02241 120001 ADC 1,1,SKP      ;C(1)=LOOP ON ERR SWIT
38 02242 156414 SUB# 2,3,SZR    ;C(2)=EFF ADDRESS OF ISZ
39 02243 003077 HALT          ;OPERAND,
40 02244 120004 MOV 1,1,SZR      ;IF C(1) NOT 0 PROG
41 02245 000754 JMP ISZ40+1      ;WILL LOOP
42

```

A 0033 .MAIN

```

01
02
03
04
05
06
07
08
09 02240 102520 DSZ0: SURZL 0,0      IC(0)=1, LOOP ON ERR SWITCH
10 02247 024135 LDA 1,C377
11 02250 152000 ADC 2,2      IC(2) NOT USED THIS TEST
12 02251 176000 ADC 3,3      IC(3)=1
13
14 02252 054122 DSZ1: STA 3,TEM      ISTORE THE VALUE
15 02253 060377 NIOP CPU      ISYNC AT 7-A74
16 02254 014122 DSZ TEM      IDECREMENT C(TEM)
17 02255 000404 JMP ,+4      IT SHOULD NOT SKIP
18 02256 020122 LDA 0,TEM      IT SKIPPED, C(0)=DSZ RESULT
19 02257 063077 HALT      ICORRECT VALUE =-2
20 02260 102401 SUB 0,0,SKP      ICLEAR C(0) TO LOOP ERR
21 02261 100404 SUP 0,1,SKP      ICHANGE AND TEST ITERATION
22 02262 000770 JMP DSZ1      ICOUNTER, ANOTHER PASS
23 02263 150014 COM# 2,2,SKP      IC(2)MODIFIED 1
24 02264 063077 HALT
25
26
27
28
29
30
31
32 02265 102520 DSZ2: SURZL 0,0      IC(0)=1, LOOP ON ERR SWITCH
33 02266 126620 SURZR 1,1      IC(1)=10000, ITERATIONS
34 02267 152520 SURZL 2,2      IC(2)=1
35 02270 050122 DSZ3: STA 2,TEM      ISTORE THE N
36 02271 060377 NIOP CPU      ISYNC AT 7-A74
37 02272 014122 DSZ TEM      IDECREMENT, IT SHOULD SKIP
38 02273 000404 JMP DSZ4      IOSZ FAIL TO SKIP
39 02274 107004 ADD 0,1,SKP      ITEST ITERATION COUNT
40 02275 000770 JMP DSZ3      ICONTINUE ON
41 02276 000405 JMP DSZ5      INEXT TEST
42 02277 020122 DSZ4: LDA 0,TEM      IC(0)=RESULT OF FAIL DSZ
43 02300 063077 HALT      ICORRECT COUNT=N
44 02301 102400 SUB 0,0      ICLEAR C(0) TO LOOP
45 02302 000766 JMP DSZ3
46 02303 101234 DSZ5: MOVZR# 0,0,SKP      IC(0) SHOULD BE +1
47 02304 063077 HALT      ICHANGE MOVZR TO LOOP TEST
48

```

A 0034 .MAIN

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15 02305 152000 DSZ10: ADC 2,2      IC(2)=1
16 02306 176000 ADC 3,3      IC(3)=1
17 02307 102620 SURZR 0,0
18 02310 101400 INC 0,0      IC(0)=100001
19 02311 143025 ADDZ 2,0,SKP      ITEST FOR PASSING THROUGH P
20 02312 000424 JMP DSZ13      IOSZ FAILED TO SKIP
21 02313 040122 STA 0,TEM      ISTORE THE VALUE FOR DSZ
22 02314 060377 NIOP CPU      ISYNC AT 7-A74
23 02315 014122 DSZ TEM      IT, DECREMENT, THE DSZ SKIPS
24 02316 176001 ADC 3,3,SKP      ION EXIT, SET C(3)=-1
25 02317 000411 JMP DSZ12      IOSZ SKIPPED.
26 02320 024122 LDA 1,TEM      ICOUNT OBTAINED FROM DSZ
27 02321 120415 ADC# 1,0,SKP      IIS COUNT CORRECT?
28 02322 000767 JMP DSZ10+4      IYES CONTINUE ON
29
30 02323 117000 DSZ11: ADD 0,3      IC(3)=CORRECT COUNT
31 02324 063077 HALT      IC(1)=DSZ COUNT
32 02325 152400 SUB 2,2      ICLEARING AC2 WILL CAUSE
33 02326 176000 ADC 3,3      IPROGRAM TO LOOP ON ERR
34 02327 000762 JMP DSZ10+4
35
36 02330 101235 DSZ12: MOVZR# 0,0,SKP      IOSZ SKIPPED, WAS THAT OK?
37 02331 000412 JMP DSZ14      IYES GO TO NEXT TEST
38 02332 024122 LDA 1,TEM      IC(1)=DSZ COUNT CAUSING
39 02333 117000 ADD 0,3      ITHE SKIP. C(3)=CORRECT
40 02334 063077 HALT      IOSZ SKIP LOGIC FAIL
41 02335 000770 JMP DSZ11+2      ILOOP ON THE ERROR
42
43 02336 024122 DSZ13: LDA 1,TEM      IOSZ FAILED TO SKIP
44 02337 176000 SUB 3,3
45 02340 102400 SUB 0,0      IFROM P.C(1)=DSZ COUNT
46 02341 063077 HALT      IC(3)=CORRECT COUNT
47 02342 000763 JMP DSZ11+2      ILOOP ON ERR
48
49 02343 000401 DSZ14: JMP ,+1      ICHANGE TO LOOP TEST
50

```

A 0035 .MAIN

```
01
02      ;A TEST OF DSZ
03      ;FIVE CONSECUTIVE DSZ INSTRUCTIONS TO LOCATION TFM
04      ;ARE EXECUTED. THE VALUE IN TEM SHOULD EQUAL C(3)
05      ;WHICH IS 5 LESS THAN THE INITIAL VALUE STORED IN
06      ;C(TEM)
07      ;THE NUMBERS ARE ARRANGED SUCH THAT NO DSZ INSTRUCTION
08      ;SHOULD SKIP.
```

```
10 02344 030134 DSZ20: LDA 2,C5      ;C(2)=ERROR LOOP SWITCH
11 02345 102020 SURZR 0,0      ;START AT 100000
12 02346 034123 LDA 3,C45      ;C(3)=EXPECTED DSZ RESULT
13 02347 117000 ADD 0,3
14 02350 040122 DSZ21: STA 0,TEM      ;STORE THE VALUE FOR DEC
15 02351 060377 NIOP CPU      ;SYNC AT 7-A74
16 02352 014122 DSZ TEM      ;DECREMENT C(TEM)
17 02353 014122 DSZ TEM
18 02354 014122 DSZ TEM
19 02355 014122 DSZ TEM
20 02356 014122 DSZ TEM
21 02357 024122 LDA 1,TEM      ;C(1)=RESULT OF DECREMENT
22 02360 136414 SUB# 1,3,SZR      ;C(3)=CORRECT COUNT
23 02361 063077 HALT          ;DSZ FAILED
24 02362 136414 SUB# 1,3,SZR
25 02363 152400 SUR 2,2      ;CLEAT C(2) TO LOOP
26 02364 143023 ADDZ 2,0,SAC      ;UPDATE FOR NEXT
27 02365 000741 JMP DSZ20+2      ;GO ROUND
28 02366 000401 DSZ22: JMP ,+1      ;CHANGE TO LOOP TEST
```

```
29
30      ;A TEST OF DSZ
31      ;EACH WORD IN THE BUFFER IS SET TO +1. DSZ THEN
32      ;DECREMENTS IT TO (+1) AND SKIPPES, IF A ERROR
33      ;OCCURES THE PROGRAM WILL CYCLE CONTINUSELY
34      ;UNTILL IT IS RESTARTED.
```

```
35
36 02367 102520 DSZ30: SURZL 0,0      ;C(0)=1, LOOP ON ERR SWITCH
37 02370 024130 LDA 1,FIN      ;C(1)=FINAL BUFFER ADDRESS
38 02371 030127 LDA 2,BUFF      ;C(2)=FIRST BUFFER ADDRESS
39 02372 176520 SUBZL 3,3      ;C(3)=+1
40 02373 055000 DSZ31: STA 3,0,2      ;STORE WORD OF 0 IN BUFFER
41 02374 060377 NIOP CPU      ;SYNC AT 7-A74
42 02375 015000 DSZ 0,2      ;DECREMENTATION SHOULD SKIP
43 02376 000412 JMP DSZ32+2      ;DSZ FAIL TO SKIP
44 02377 035000 LDA 3,0,2      ;DSZ SKIPPED OK, IS MEM
45 02400 175014 MOV# 3,3,SZR      ;WORD 0?
46 02401 000405 JMP DSZ32      ;NO! ERROR
47 02402 113000 ADD 0,2      ;ADVANCE TO NEXT LOCATION
48 02403 132414 SUB# 1,2,SZR      ;MORE LOCATIONS IN BUFFER
49 02404 000760 JMP DSZ31-1      ;GO TO NEXT TEST
50 02405 000407 JMP DSZ33      ;DSZ SKIPPED BUT STORE NG
51 02406 063077 DSZ32: HALT          ;C(2)=EFF ADDRESS,C(3)=RESULT
52 02407 000403 JMP ,+3      ;OF THE DSZ
53 02410 035000 LDA 3,0,2      ;DSZ FAIL TO SKIP
54 02411 063077 HALT          ;C(2)=EFFECTIVE ADDRESS
55 02412 102400 SUR 0,0      ;C(2)=EFFECTIVE ADDRESS
56 02413 000757 JMP DSZ31-1      ;LOOP THIS TEST
57 02414 000401 DSZ33: JMP ,+1      ;CHANGE TO LOOP TEST
```

A 0036 .MAIN

```
01
02      ;ANOTHER TEST OF DSZ
03      ;THIS TEST WILL SUBTRACT ONE FROM EACH LOCATION OF A
04      ;BUFFER SET TO +1, THE TEST HAS BEEN WRITTEN TO GIVE
05      ;A HIGH REP RATE FOR THE DSZ INSTRUCTION, PRESSING
06      ;CONTINUE AFTER A ERPOP WILL CAUSE THE TEST TO BE
07      ;ITERATED UNTILL RESTARTED.
```

```
08
09 02415 126000 DSZ40: ADC 1,1      ;C(1)=LOOP ON ERROR SWITCH
10 02416 102520 SURZL 0,0      ;C(0)=+1
11 02417 034130 LDA 3,FIN
12 02420 030127 LDA 2,BUFF
13 02421 041000 STA 0,0,2      ;SET ALL LOCATIONS IN
14 02422 151400 INC 2,2      ;THE BUFFER TO ZERO
15 02423 156414 SUB# 2,3,SZR      ;EXCEPT THE LAST LOCATION
16 02424 000775 JMP ,+3
17 02425 051400 STA 2,0,3      ;DSZ SKIPS ALL BUT LAST
18
19 02426 102000 DSZ41: ADC 0,0      ;C(0)=+1, FOR ERROR CHECK
20 02427 034124 LDA 3,C4      ;C(3)=4, BUFFER INCREMENT
21 02430 030127 LDA 2,BUFF      ;C(2) POINTS TO THE BUFFER
22
23 02431 173000 DSZ42: ADD 3,2      ;INC BUFFER POINTER BY 4
24 02432 060377 NIOP CPU      ;SYNC 7-A74
25 02433 015374 DSZ -4,2      ;DECREMENT THE BUFFER
26 02434 000410 JMP DSZ43      ;VIA DSZ, DSZ SHOULD SKIP
27 02435 015375 DSZ -3,2      ;UNLESS END OF BUFFER
28 02436 000407 JMP DSZ43+1
29 02437 015376 DSZ -2,2
30 02440 000406 JMP DSZ43+2
31 02441 015377 DSZ -1,2
32 02442 000405 JMP DSZ43+3
33 02443 000760 JMP DSZ42      ;NOT YET BUFFERS END
```

```
34
35 02444 113000 DSZ43: ADD 0,2      ;C(2) WILL BE SET
36 02445 113000 ADD 0,2      ;TO THE EFFECTIVE ADDRESS
37 02446 113000 ADD 0,2      ;OF THE ENTERING DSZ
38 02447 113000 ADD 0,2      ;INSTRUCTION
39 02450 034130 LDA 3,FIN      ;VOID DSZ ENTER AT END OF
40 02451 021000 LDA 0,0,2      ;BUFFER OR DID IT FAIL?
41 02452 156414 SUB# 2,3,SZR      ;C(3)=END OF BUFFER
42 02453 126400 SUB 1,1      ;C(2)=EFFECTIVE ADDRESS
43 02454 156414 SUB# 2,3,SZR      ;C(0)=RESULT OF DSZ INBT
44 02455 063077 HALT          ;A DSZ FAILED TO SKIP
45 02456 124014 COM# 1,1,SZR      ;IF C(1)=0 LOOP ON ERROR
46 02457 000737 JMP DSZ40+1
```

A 0037 .MAIN

```

01
02      JA TEST OF INDIRECT ADDRESSING
03      I THE C(TEM) IS SET TO POINT TO A BUFFER. A LDA
04      INSTRUCTION THEN REFFERANCES THE BUFFER INDIRECTLY.
05      I THE C(TEM) IS CHECKED TO INSURE THAT IT HAS NOT
06      I CHANGED. PRESSING CONTINUE AFTER A ERROR WILL CAUSE
07      I THE PROGRAM TO LOOP ON THE FAILING CONDITIONS
08      UNTILL RESTARTED.
09 02450 102520 DEF0: SUBZL 0,0      IC(0)=1, LOOP ON ERR SWITCH
10 02451 030127      LDA 2,BUFF      IFIRST ADDRESS IN BUFFER
11 02452 034130      LDA 3,FIN      IFINAL ADDRESS IN BUFFER
12 02453 050122      STA 2,TEM      IC(2) STORED IN C(TEM)
13 02454 060377      NIOP CPU      ISYNC AT 7-A74
14 02455 020122      LDA 1,0TEM      IREFFERENCE REGISTER TEM
15 02456 024122      LDA 1,TEM      I HAS C(TEM) CHANGED
16 02457 132414      SUB# 1,2,SZR  IF+OR= ONE PERHAPS AUTO IDX
17 02458 063077      HALT          I YES, C(1)=VALUE OF TEM
18 02459 132414      SUB# 1,2,SZR  IC(2)=VALUE STORED IN TEM
19 02460 102400      SUB 0,0        ICLEAR C(0) TO LOOP
20 02461 113000      ADD 0,2        IINC BUFFER POINTER
21 02462 156414      SUB# 2,3,SZR  ITEST FOR END OF BUFF
22 02463 000766      JMP DEF0+3
23 02464 000401      JMP .+1      ICHANGE TO LOOP TEST

```

A 0038 .MAIN

```

01
02      JA TEST OF INDIRECT ADDRESSING
03      I A ADDRESS PATTERN (C(ADDRESS)=ADDRESS) IS STORED IN
04      I THE BUFFER VIA INDEX REGISTER 3. A LDA INSTRUCTION
05      I THEN
06      I REFFERANCES EACH LOCATION IN THE BUFFER INDIRECTLY,
07      I BECAUSE THE BUFFER CONTAINS A ADDRESS PATTERN THE DATA
08      I OBTAINED REPRESENTS THE ADDRESS OBTAINED. PRESSING
09      I CONTINUE AFTER A ERROR CAUSES THE PROGRAM TO LOOP ON
10      I THE
11      I ERROR UNTILL RESTARTED.
12
13 02477 102520 DEF10: SUBZL 0,0      IC(0)=1, LOOP ON ERR SWITCH
14 02500 034127      LDA 3,BUFF      IFIRST BUFFER LOCATION
15 02501 030130      LDA 2,FIN      IFINAL BUFFER LOCATION
16 02502 055400      STA 3,0,3      ISTORE THE ADDRESS
17 02503 175400      INC 3,3        IPATTERN IN THE BUFFER
18 02504 156414      SUB# 2,3,SZR  IEXAMPLE: C(4105)=4105
19 02505 000775      JMP .-3
20 02506 034127      LDA 3,BUFF      IADDRESS PATTERN IS STORED
21 02507 054122 DEF11: STA 3,TEM      IPRESET C(TEM) FOR INDIRECT
22 02508 060377      NIOP CPU      ISYNC AT 7-A74
23 02509 020122      LDA 1,0TEM      IGO INDIRECT TO GET DATA
24 02510 136414      SUB# 1,3,SZR  IC(1)=DATA/ADDRESS OBTAINED
25 02511 063077      HALT          IC(3)=CORRECT DATA/ADDRESS
26 02512 136414      SUB# 1,3,SZR  IIF IT FAILS
27 02513 102400      SUB 0,0        ICLEAR 0 TO LOOP ON ERROR
28 02514 117000      ADD 0,3        IADVANCE TO NEXT ADDRESS
29 02515 156414      SUB# 2,3,SZR  ITEST FOR END OF BUFFER
30 02516 000767      JMP DEF11
31 02517 000401      JMP .+1      ICHANGE TO LOOP TEST

```

A 0039 ,MAIN

```

01
02      JA TEST OF INDIRECT ADDRESSING
03      IWE DO NOT YET KNOW THAT INDIRECT ADDRESSING WORKS
04      ITHIS TEST INSURES THAT THE STA INST STORES INDIRECT
05      IAND NOT DIRECTLY, IF A ERR OCCURES PRESSING CONTINUE
06      IWILL CAUSE THE PROGRAM TO LLOP ON ERROR UNTILL
07      IRESTART.
08
09 02522 102520 DEF14: SUBZL R,P      IC(0)=1, LOOP ON ERROR SWITCH
10 02523 030127      LDA 2,BUFF      IC(2) POINTS TO THE BUFFER
11 02524 034135      LDA 3,C377      IC(3)=ITERATION COUNTER
12 02525 050122      STA 2,TEM      ISTORE BUFF POINTER IN C(TEM)
13 02526 150000      COM 2,2      ICHANGE C(2)
14 02527 060377      NIOP CPU      ISYNC AT 7-A74
15 02530 052122      STA 2,TEM      ISTORE VIA C(TEM) INDIRECT
16 02531 150000      COM 2,2      IRESTORE C(2)
17 02532 024122      LDA 1,TEM      IOIO C(TEM) CHANGE?
18 02533 132414      SUB# 1,2,SZR   IC(1)=VALUE IN C(TEM)
19 02534 063077      HALT          IC(2)=ORIG VALUE OF TEM
20 02535 146414      SUB# 2,1,SZR   IIF C(1) IS COMP OF C(2)
21 02536 192400      SUB 0,P      ISTA STORED DIRECT
22 02537 113000      ADD 0,2      ILOOP ON ERROR
23 02540 116494      SUB 0,3,SZR   ITERATION COUNTER
24 02541 000764      JMP DEF14+3
25 02542 000401 DEF15: JMP ,+1      ICHANGE TO LOOP PROGRAM
26
27      JA TEST OF INDIRECT ADDRESSING
28      ITHIS ROUTINE WILL TEST THE ABILITY OF THE STA
29      IINSTRUCTION
30      ITO STORE IN A BUFFER USING INDIRECT ADDRESSING, THE
31      IDATA
32      ISTORED IS THE COMPLEMENT OF THE ADDRESS INTO WHICH IT
33      IIS
34      ISTORED, EXAMPLE: C(5473)=172704. IF A ERROR OCCURES
35      IPRESS
36      ICONTINUE TO LOOP ON THE FAILING CONDITIONS.
37
38 02543 102520 DEF20: SUBZL R,P      IC(0)=1, LOOP ON ERR SWITCH
39 02544 030127      LDA 2,BUFF      IC(2)=FIRST BUFFER LOCATION
40 02545 050122      STA 2,TEM      ISET TEM TO POINT TO BUFFER
41 02546 154000      COM 2,3
42 02547 060377      NIOP CPU      ISYNC AT 7-A74
43 02550 056122      STA 3,TEM      ISTORE COMP OF ADDRESS
44 02551 025000      LDA 1,0,2      IVALUE STORED TO AC1
45 02552 136414      SUB# 1,3,SZR   IDIS IT STORE OK?
46 02553 063077      HALT          IC(2)=ADDRESS IN BUFFER
47 02554 136414      SUB# 1,3,SZR   IC(3)=DATA STORED
48 02555 102400      SUB 0,P      IC(1)=DATA FROM BUFFER
49 02556 113000      ADD 0,2      IADVANCE TO NEXT LOC
50 02557 034130      LDA 3,FIN      ITEST FOR END OF BUFFER
51 02560 156414      SUB# 2,3,SZR
52 02561 000764      JMP DEF20+2
53 02562 000401 DEF21: JMP ,+1      ICHANGE TO LOOP TEST
54

```

A 0040 ,MAIN

```

01
02      JA TEST OF INDIRECT ADDRESSING
03      ITHE BUFFER IS SET TO CONTAIN A ADDRESS PATTERN, A LDA
04      IINSTRUCTION VIA C(2) THEN CAUSES A INDIRECT REFERENCE
05      ITO BE MADE AT EACH LOCATION IN THE BUFFER, BECAUSE
06      IEACH LOCATION CONTAINS ITS ADDRESS THE LDA INSTRUCTION
07      IWILL LOAD THE LOCATION INDIRECTED THROUGH, PRESSING
08      ICONTINUE AFTER A ERR WILL CAUSE THE PROGRAM TO LOOP ON
09      IERROR UNTILL RESTARTED.
10
11 02563 102520 DEF30: SUBZL R,P      IC(0)=1, LOOP ON ERROR SWITCH
12 02564 030127      LDA 2,BUFF      IFIRST ADDRESS IN BUFFER
13 02565 034130      LDA 3,FIN      IFINAL ADDRESS IN BUFFER
14 02566 051000      STA 2,0,2      ISTORE A ADDRESS PATTERN
15 02567 151400      INC 2,2      IIN THE BUFFER
16 02570 156414      SUB# 2,3,SZR   IEXAMPLE: C(7314)=7314
17 02571 000775      JMP ,+3
18
19 02572 030127 DEF31: LDA 2,BUFF      IFIRST BUFFER LOCATION
20 02573 060377      NIOP CPU      ISYNC AT 7-A74
21 02574 027000      LDA 1,0,2      IWOULD LOAD WD POINTED TO
22 02575 132414      SUB# 1,2,SZR   IC(2)=ADDRESS INDIRECT THROUGH
23 02576 063077      HALT          IC(1)=VALUE OBTAINED
24 02577 132414      SUB# 1,2,SZR
25 02580 102400      SUB 0,P      ICLEAR 0 TO LOOP ON ERROR
26 02581 113000      ADD 0,2      INC TO NEXT BUFFER LOC
27 02582 156414      SUB# 2,3,SZR   ITEST FOR END OF BUFFER
28 02583 000770      JMP DEF31+1
29 02584 000401 DEF32: JMP ,+1      ICHANGE TO LOOP TEST
30
31      JA TEST OF INDIRECT ADDRESSING
32      I A MODIFIED ADDRESS PATTERN (C(ADDRESS)=ADDRESS-1) IS
33      ISTORED IN THE BUFFER, A STA INSTRUCTION INDIRECTS
34      ITHROUGH EACH LOCATION IN THE BUFFER STORING IN THAT
35      ILOCATION-1. PRESSING CONTINUE AFTER A ERROR CAUSES THE
36      IPROGRAM TO LOOP ON THE ERROR UNTILL RESTARTED.
37
38 02605 102520 DEF34: SUBZL R,P      IC(0)=1
39 02606 034130      LDA 3,FIN      IFINAL LOCATION IN BUFFER
40 02607 030127      LDA 2,BUFF      IFIRST LOCATION IN BUFFER
41 02610 051001      STA 2,1,2      IIN EACH BUFFER LOCATION
42 02611 151400      INC 2,2      ISTORE THE ADDRESS OF THAT
43 02612 156414      SUB# 2,3,SZR   ILOCATION-1
44 02613 000775      JMP ,+3
45 02614 030127 DEF35: LDA 2,BUFF      IFIRST BUFFER LOCATION
46 02615 060377      NIOP CPU      ISYNC AT 7-A74
47 02616 053001      STA 2,0,1,2      IDEFER THROUGH BUFFER, STORE
48 02617 025000      LDA 1,0,2      IC(2) IN BUFFER-1, THEREFOR
49 02620 132414      SUB# 1,2,SZR   ISTORE ADDRESS IN ADDRESS.
50 02621 063077      HALT          IFOR EXAMPLE: C(4050)=XXX
51 02622 132414      SUB# 1,2,SZR   IC(4051)=4050, C(2)=4050
52 02623 102400      SUB 0,0      IINDIRECT THROUGH 4051 TO
53 02624 113000      ADD 0,2      ISTORE C(2) IN LOCATION 4050
54 02625 156014      ADD# 2,3,SZR   ITEST FOR END OF BUFFER
55 02626 000767      JMP DEF35+1
56 02627 000401 DEF36: JMP ,+1      ICHANGE TO LOOP TEST
57

```

A 0041 ,MAIN

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

IA TEST OF INDIRECT ADDRESSING (2 LEVELS)
THE BUFFER IS SET TO CONTAIN A ADDRESS PATTERN, A LDA
INSTRUCTION USING TWO LEVELS OF INDIRECT ADDRESSING
THEN REFFERANCES EACH LOCATION IN THE BUFFER, PRESSING
CONTINUE AFTER A ERROR WILL CAUSE THE PROGRAM TO LOOP
ON THE ERROR UNTILL RESTARTED.

DEF40: SURZL 0,P ;IC(0)=1,LOOP ON ERR SWITCH
LDA 2,BUFF ;FIRST BUFFER LOCATION
LDA 3,FIN ;FINAL BUFFER LOCATION
STA 2,0,2 ;STORE A ADDRESS PATTERN
INC 2,2 ;IN THE BUFFER
SURM 2,3,SZR
JMP -3
DEF41: LDA 2,BUFF ;FIRST BUFFER LOCATION
SUBZR 1,1 ;IC(1)=100000,THE DEFER BIT
ADD 2,1 ;IC(2)=INDIRECT POINTER
STA 1,TEM ;TO BUFFER
NIOP CPU ;SYNC AT 7-A74
LDA 1,TEM ;GO INDIRECT FOR DATA
SURM 1,2,SZR ;IC(1)=DATA FROM BUFFER
HALT ;IC(2)=CORRECT DATA
SURM 1,2,SZR ;2 LEVELS INDIRECT FAIL
SUR 0,P ;CLEAR TO LOOP ON ERROR
ADD 0,2 ; INC TO NEXT BUFF LOC
SURM 2,3,SZR ;TEST FOR END OF BUFFER
JMP DEF41+1
DEF42: JMP -3 ;CHANGE TO LOOP TEST

IA TEST OF JSR (HARD TEST)
THE BUFFER IS FILLED WITH THE FOLLOWING
JSR 2,3 ;GO TO THE LOCATION SPECIFIED BYC(3)+2
JSR 0,2 ;GO BACK TO THE CONTROL PROGRAM
JSR 0,2 ;" " " " " "
JSR 0,2 ;" " " " " "
JSR 2,3 ;GO TO THE LOCATION SPECIFIED BY C(3)+2
THE C(3) IS INITIALLY SET TO BUFFER+1
THE C(2) " " " " RETURN TO MAIN PROGRAM
THE PROGRAM THEN JUMPS TO THE FIRST BUFFER LOCATION,
THE BUFFER SEQUENCE IS AS FOLLOWS:
#1 JSR TO BUFFER+3 C(3)=BUFFER+1
#2 JSR TO BUFFER+3 C(3)=BUFFER+1
#3 JSR TO BUFFER+6 C(3)=BUFFER+4
#4 JSR TO BUFFER+6 C(3)=BUFFER+7
#5 JSR TO BUFFER+11 C(3)=BUFFER+7
#6 ETC

A 0042 ,MAIN

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

THIS PROCESS CONTINUES UNTILL THE END OF THE BUFFER IS
REACHED.
AT FINAL ADDEESS,FINAL-1,FINAL+1,THE PROGRAM RETURNS
TO THE
MAIN SEQUENCE. THE PC STORED BY THE FINAL JSR IS
CHECKED.

JSR20: LDA 2,BUFF ;FIRST BUFFER LOCATION
LDA 3,FIN ;LAST BUFFER LOCATION
LDA 0,CJSR3 ;CONSTANT (JSR 2,3)=CONTINUE
LDA 1,CJSR2 ;CONSTANT (JSR 0,2)=RETURN
STA 0,0,2 ;STORE CONTINUE
STA 1,1,2 ;" RETURN
STA 1,2,2 ;" RETURN
INC 2,2
INC 2,2
INC 2,2
SURZM 2,3,SZR ;TEST FOR END OF BUFFER
JMP JSR21 ;PERHAPS 1,2 EXTRA WORDS
STA 1,-3,2 ;MAKE LAST CONTINUE A RETURN

JSR22: LDA 3,BUFF ;INITIAL SETUP OF C(3)
INC 3,3 ;FOR FIRST JSR IN BUFFER
LDA 2,CJSR4 ;THE PROGRAM RETURN
SUR 0,P ;BOTH C(0),C(1) SET TO 0
SUR 1,1 ;WILL CHECK FOR NO EFFECT BY
JSR ;JSR
JMP -1,3 ;GO TO THE BUFFER!

JSR23: JSR 2,2 ;THREE CONSTANTS
JSR 2,3
-3
ADD 0,1,SZR ;IC(0),C(1) SHOULD NOT HAVE
HALT ;BEEN CHANGED!!
LDA 0,FIN ;TEST C(3) FOR FINAL,FINAL-1
SUBOR 3,0,57R ;FOR FINAL+1,ADDRESS
COM 0,0,SUR ;IC(3) = ADDRESS OF RETURNING
MOV 0,0,SUR
HALT
MOV 0,0 ;CHANGE TO LOOP TEST

A 0243 .MAIN

```

01
02      JA TEST OF INDIRECT ADDRESSING (2 LEVELS)
03      I THE FIRST HALF OF THE BUFFER IS FILLED WITH (JSR 0,3)
04      INSTRUCTIONS, THE SECOND HALF IS FILLED
05      I AND ADDRESS POINTING TO THE (JSR) INSTRUCTIONS IN THE
06      I BOTTOM. C(TEM) POINTS TO THE TOP HALF OF THE BUFFER.
07      I THE SEQUENCE IS AS FOLLOWS:
08      I #1 GO INDIRECT THROUGH REGISTER TEM
09      I #2 WORD IN TOP POINTS TO BOTTOM
10      I #3 INSTRUCTION IN BOTTOM IS EXECUTED
11      I #4 PC (C(3)) FROM BOTTOM IS CHECKED
12      I PRESSING CONTINUE AFTER A ERROR WILL CAUSE THE PROGRAM
13      I TO LOOP ON THE ERROR UNTILL RESTARTED.
14
15 02713 034130 DEF50: LDA 3,FIN      I FINAL BUFFER ADDRESS
16 02714 030127 LDA 2,BUFF      I FIRST BUFFER ADDRESS
17 02715 024127 LDA 1,BUFF
18 02716 020437 LDA 0,CJDF1      I CONSTANT (JSR 0,3)
19 02717 166440 SUBOR 3,1      I C(1)=(LAST-FIRST)/2==WC
20 02720 041000 STA 0,0,2      I STORE THE JSRS
21 02721 151400 INC 2,2      I IN BOTTOM PART OF BUFFER
22 02722 125404 INC 1,1,SZR
23 02723 000775 JMP ,=3
24 02724 050122 STA 2,TEM      I C(TEM)=FIRST LOC IN TOP
25 02725 020127 LDA 0,BUFF
26 02726 041000 DEF51: STA 0,0,2      I STORE A MODIFIED ADDRESS
27 02727 101400 INC 0,0      I PATTERN IN TOP HALF OF
28 02730 151400 INC 2,2      I THE BUFFER, C(TOP)
29 02731 156414 SUB# 2,3,SZR      I BOTTOM ADDRESS+100000
30 02732 000774 JMP ,=4
31
32 02733 030122 DEF52: LDA 2,TEM      I C(2) POINTS TO TOP HALF
33 02734 102620 SURZR 0,0      I C(0)=100000
34 02735 143000 ADD 2,0      I THIS ROUTINE ADDS THE
35 02736 040122 STA 0,TEM      I INDIRECT BIT TO C(TEM)
36 02737 024127 LDA 1,BUFF      I C(1) USED FOR CHECK
37 02740 102520 SUBZL 0,0      I C(0)=1, LOOP ON ERR SWITCH
38
39 02741 060377 DEF53: NIOP CPU      I SYNC AT 7-474
40 02742 006122 JSR 0,TEM      I GO TO THE BUFFER C(3)
41 02743 136014 ADC# 1,3,SZR      I JSR RETURN, C(1)=CORRECT
42 02744 063077 HALT      I FINAL ADDRESS, INDIRECT WITH
43 02745 136014 ADC# 1,3,SZR      I JSR FAILED
44 02746 102400 SUB 0,0      I CLEAR TO LOOP ON ERROR
45 02747 107000 ADD 0,1      I POINT TO NEXT BUFFER LOC
46 02750 101014 MOV# 0,0,SZR
47 02751 010122 ISZ TEM      I INC IF NOT ERROR
48 02752 132014 ADC# 1,2,SZR      I TEST FOR END OF BUFFER
49 02753 000760 JMP DEF53
50 02754 101011 DEF54: MOV# 0,0,SKP      I CHANGE TO LOOP TEST
51 02755 005400 CJDF1: JSR 0,3      I A CONSTANT
52

```

A 0244 .MAIN

```

01
02 02756 020127 RANFL: LDA 0,BUFF      I FILL THE BUFFER WITH
03 02757 040122 STA 0,TEM      I RANDOM ADDRESS IN
04 02760 034130 LDA 3,FIN      I BITS 4-15, THIS VALUE
05 02761 024000 LDA 1,0      I WILL POINT TO SOME
06 02762 121000 RANFC: MOV 1,0      I OTHER SPOT IN THE BUFFER
07 02763 024433 LDA 1,C1347      I BIT 0 IS RANDOMLY 0/1
08 02764 152020 SUBZR 2,2
09 02765 107222 ADDZR 0,1,SZR      I THIS
10 02766 147000 ADD 2,1
11 02767 131000 MOV 1,2      I ROUTINE PRODUCES A
12 02770 113520 ANDZL 0,2      I RANDOM 16 BIT NUMBER
13 02771 107000 ADD 0,1      I C(1)=RANDOM
14 02772 146400 SUM 2,1
15
16 02773 020125 LDA 2,MSIZE      I MASK OF MEM SIZE
17 02774 123400 AND 1,0      I C(0)=RANDOM MODULO MEMORY
18 02775 030127 LDA 2,BUFF
19 02776 162433 SURZR 3,0,SNC      I IS C(0) WITH IN THE
20 02777 142033 ADCZ# 2,0,SNC      I RANGE OF THE BUFFER?
21 03000 000762 JMP RANFC      I NO TO BIG OR TO SMALL
22 03001 030122 LDA 2,TEM
23 03002 112415 SUB# 0,2,SNR
24 03003 000757 JMP RANFC      I REJECT IF ADDRESS SAME
25 03004 101100 MOVZL 0,0
26 03005 125100 MOVL 1,1
27 03006 101200 MOV# 0,0
28 03007 125200 MOV# 1,1
29 03010 042122 STA 0,0,TEM      I STORE THE WORD
30 03011 010122 ISZ TEM
31 03012 030122 LDA 2,TEM      I TEST FOR END OF BUFFER
32 03013 156014 ADC# 2,3,SZR
33 03014 000746 JMP RANFC      I MAKE MORE RANDOM
34 03015 101011 MOV# 0,0,SKP      I CHANGE TO LOOP
35 03016 135753 C1347: 135753
36

```

A 0045 .MAIN

```

01
02
03      JA TEST OF INDIRECT ADDRESSING
04      !THE BUFFER IS FILLED WITH PSEUDO RANDOM NUMBERS WHICH
05      !ARE ADDRESSES, THE VALUE OF EACH ADDRESS POINTS TO
06      !SOME SPOT IN THE BUFFER, BIT 0 OF EACH BUFFER
07      !WORD IS RANDOMLY SET OR CLEARED, THE PROGRAM FIRST
08      !POINTS TO THE FIRST BUFFER LOCATION, THE FINAL
09      !EFFECTIVE ADDRESS IS CALCULATED UP TO 15 LEVELS OF
10      !INDIRECT, A LDA INSTRUCTION THEN REFFERENCES
11      !INDIRECTLY
12      !THE BUFFER, THE RESULT OF THE INDIRECT LDA SHOULD BE
13      !THE SAME AS THE CALCULATED VALUE, IF IT IS NOT THE
14      !SAME THE CORRECT CHAIN SHOULD BE CALCULATED WITH PAPER
15      !AND PENCIL, RESTARTING THE PROGRAM AT THE LOCATION
16      !FOLLOWING THE HALT WILL CAUSE THE PROGRAM TO ENTER
17      !A FAILING LOOP UNTILL RESTARTED, A SYNC PULSE AT
18      !7-A74 PROCEEDS THE INDIRECT CHAIN.
19
20 03017 024127 DEF00: LDA 1,BUFF      !FIRST LOCATION IN THE BUFFER
21 03020 121120      MOVZL 1,0
22 03021 121240      MOVOR 0,0
23 03022 040136      STA 0,CF00
24 03023 122520      SUBZL 0,0
25 03024 131000      MOV 1,2
26 03025 121125      MOVZL 0,0,SNR      !COUNT INDIRECT LEVELS
27 03026 020417      JMP DEF02      !15 LEVELS,GIVE UP THIS CHAIN
28 03027 031000      LDA 2,0,2      !GET BUFFER WORD
29 03030 151132      MOVZL# 2,2,SYZ      !DOES IT HAVE A INDIRECT BIT
30 03031 020774      JMP ,=4      !YES KEEP LOOKING
31 03032 050122      STA 2,TEM
32 03033 031000      LDA 2,0,2      !C(2)=FINAL VALUE
33
34 03034 122520 DEF01: SUBZL 0,0      !C(0)=1,LOOP ON EKR SWITCH
35 03035 060377      NIOP CPU      !SYNC AT 7-A74
36 03036 036136      LDA 3,0,CF00      !REFFERENCE THE BUFFER
37 03037 172414      SUB# 3,2,SYR      !C(3)=WORD OBTAINED
38 03040 063077      HALT      !C(2)=CORRECT
39 03041 172414      SUB# 3,2,SYW      !C(1)=PLACE BUFFER WAS ENTERED
40 03042 122401      SUB 0,0,SKP      !WRITE DOWN THE CORRECT
41 03043 121005      MOV 0,0,SNR      !INDIRECT CHAIN,SINGLE STEP
42 03044 020771      JMP DEF01+1      !LOOP ON THE ERROR
43
44 03045 125400 DEF02: INC 1,1      !GO TO THE NEXT BUFFER LOC
45 03046 034130      LDA 3,FIN      !TEST FOR FINAL
46 03047 136414      SUB# 1,3,SYZ
47 03050 020750      JMP DEF00+1      !NOT YET BUFFER END
48
49 03051 020401 DEF03: JMP ,+1      !CHANGE TO LOOP TEST
50

```

A 0046 .MAIN

```

01
02      JA TEST OF INDIRECT ADDRESSING
03      !THIS TEST IS SIMILAR TO THE PREVIOUS TEST IN THAT THE
04      !BUFFER IS FILLED WITH PSEUDO RANDOM NUMBERS, EACH
05      !LOCATION IN THE BUFFER MAY CONTAIN A INDIRECT BIT,
06      !EACH LOCATION ALSO CONTAINS A N BIT ADDRESS OF A
07      !BUFFER LOCATION, A ROUTINE CALCULATES EFFECTIVE
08      !ADDRESS UP TO 15 LEVELS, THE EFFECT OF A INDIRECT
09      !ISZ/DSZ INSTRUCTION PAIR IS CHECKED.
10
11 03052 030127 EXCH: LDA 2,BUFF      !CAN'T USE AUTO INC/DEC YET
12 03053 034130      LDA 3,FIN      !EXCHANGE BUFFER ENDS SO
13 03054 021000      LDA 0,0,2      !IT WON'T BE THE SAME AS
14 03055 025777      LDA 1,-1,3      !LAST TEST
15 03056 045000      STA 1,0,2
16 03057 041777      STA 0,-1,3
17 03060 151400      INC 2,2
18 03061 174400      NEG 3,3      !INC TOWARDS TOP,DEC TOWARDS
19 03062 174000      COM 3,3      !BOTTEM
20 03063 172433      SUBZ# 3,2,SYN      !TEST FOR MIDDLE
21 03064 020770      JMP EXCH+2      !OF THE BUFFER
22
23 03065 024127 DEF04: LDA 1,BUFF      !FIRST BUFFER LOCATION
24 03066 121120      MOVZL 1,0
25 03067 121240      MOVOR 0,0
26 03070 040136      STA 0,CF00
27 03071 122520      SUBZL 0,0
28 03072 131000      MOV 1,2
29 03073 121125      MOVZL 0,0,SNR      !IF MORE THAN 15 LEVELS
30 03074 020422      JMP DEF06+2      !OF INDIRECT,GIVE UP
31 03075 031000      LDA 2,0,2      !CALCULATE EFFECTIVE
32 03076 151132      MOVZL# 2,2,SYZ      !ADDRESS, LOOK FOR INDIRECT
33 03077 020774      JMP ,=4      !BIT 0 A ONE
34 03100 050122      STA 2,TEM      !C(TEM)=EFFECTIVE ADDRESS
35 03121 031000      LDA 2,0,2      !C(2)=DATA THAT ADDRESS
36
37

```

A 0047 .MAIN

```

01
02 03102 102520 DEF65: SUBZL 0,0      IC(0)=1, LOOP ON ERROR SWITCH
03 03103 060377      NIOP CPU      ISYNC AT 7-A74
04 03104 012136      ISZ 0CF00
05 03105 036122      LDA 3,0TEM
06 03106 016122      OSZ 0TEM
07 03107 156015      ADC# 2,3,SNR
08 03110 000413      JMP DEF67
09 03111 063077      HALT
10 03112 102400      SUB 0,0
11 03113 052122      STA 2,0TEM
12 03114 101005 DEF66: MOV 0,0,SNR
13 03115 000766      JMP DEF65+1
14 03116 125400      INC 1,1
15 03117 034130      LDA 3,FIN
16 03120 130414      SUB# 1,3,SZR
17 03121 000745      JMP DEF64+1
18 03122 000406      JMP .+6
19
20 03123 030122 DEF67: LDA 3,0TEM
21 03124 172415      SUB# 3,2,SNR
22 03125 000767      JMP DEF66
23 03126 063077      HALT
24 03127 000763      JMP DEF65-2
25 03130 000401      JMP .+1

```

```

INCREMENT CORRECT?
IC(2)=CORRECT,C(TEM)=ADDRESS
IC(3)=WORD FROM BUFFER
ICLEAR TO LOOP ON ERROR
IMOPFULLY RESTORE MEMORY
ITEST THE ERR SWITCH
ILOOP ON ERROR
IADVANCE BUFFER POINTER
ITEST FOR FINAL ADDRESS

IGO TO NEXT TEST

ISZ WORKED TEST OSZ
IC(2)=CORRECT,C(TEM)=ADDRESS
IC(3)=DATA FROM MEMORY
IPRESS CONTINUE TO
IRESTORE AND LOOP ON ERR
ICHANGE TO LOOP TEST

```

A 0048 .MAIN

```

01
02      IA TEST OF AUTO INCREMENT/DECREMENT
03      ITHIS TEST WILL INSURE THAT REGISTERS 20-37
04      IDO NOT CHANGE VALUE WHEN REFERENCED WITHOUT A
05      INDIRECT BIT.
06
07 03131 102400 ID01: SUB 0,0      IFIRST PASS STORES (0)
08 03132 152400      SUB 2,2      IIN REGISTERS 20-37
09 03133 126020      SUBZL 1,1    ISECOND PASS STORES (-1)
10 03134 041020      STA 0,20,2
11 03135 151400      INC 2,2      INC ADDRESS
12 03136 125224      MOVZL 1,1,SZR IITERATION COUNT OF 20
13 03137 000775      JMP .-3
14
15 03140 176620 ID11: SUBZL 3,3    ISET ITERATION COUNT
16 03141 021000      LDA 0,0,2    ILOOK FOR A CHANGE
17 03142 025000      LDA 1,0,2    IIN TO REFFERENCES
18 03143 106414      SUB# 0,1,SZR ITO AUTO INC/DEC LOCATIONS
19 03144 063077      HALT        IC(2)=ADDRESS,C(0)=FIRST
20 03145 151400      INC 2,2      IREFERENCE,C(1)=SECOND
21 03146 175224      MOVZL 3,3,SZR IGO THROUGH ALL INC/DEC
22 03147 000772      JMP ID1+1    IREGISTERS
23
24 03150 102000      ADC 0,0      IC(0)=-1
25 03151 151112      MOVL# 2,2,SZC IFIRST OR SECOND PASS
26 03152 000403      JMP ID2     ISECOND,EXIT TEST
27 03153 152620      SUBZL 2,2    IC(2)=100000
28 03154 000757      JMP ID0+2
29 03155 000421 ID21: JMP .+1     ICHANGE TO LOOP TEST
30

```

A 0049 ,MAIN

```

01
02
03      I AUTO INCREMENT CHECK
04      I COUNT ALL THE AUTO INCREMENT REGISTERS FROM
05      I 60000-77777
06      I CHECK THE VALUE AGAINST A ADD INSTRUCTION. PRESS
07      I CONTINUE AFTER A HALT TO LOOP ON ERROR.
08
09 03156 102520 ID10:  SUBZL 0,0      IC(0)=1, LOOP ON ERROR SWITCH
10 03157 176400      SUB 3,3      IC(3)=0-7, REGISTER POINTER
11 03160 152620      SURZR 2,2
12 03161 151240      MOVZR 2,2
13 03162 151220      MOVZR 2,2      I STARTING VALUE OF 60000
14 03163 051420      STA 2,20,3    I SETUP AUTO INC REGISTER
15 03164 060377      NIOP CPU     I SYNC AT 7-A74
16 03165 027420      LDA 1,020,3   I THIS SHOULD INC THE REG
17 03166 025420      LDA 1,20,3    IC(1)=VALUE OF AUTO REG
18 03167 146014      ADC# 2,1,SZR  IC(2)=CORRECT VALUE-1
19 03170 063077      HALT          IC(3)+20=FAILING REG
20 03171 146014      ADC# 2,1,SZR
21 03172 102400      SUB 0,0      IC CLEAR TO LOOP ON ERROR
22 03173 113000      ADD 0,2
23 03174 151113      MOVL# 2,2,3NC I TEST FOR COUNT OF 100000
24 03175 000766      JMP ID10+5    I NOT YET
25 03176 175400      INC 3,3      I ADVANCE TO NEXT AUTO REG
26 03177 024111      LDA 1,C10
27 03200 136414      SUR# 1,3,SZR  I TEST FOR LAST REGISTER
28 03201 000757      JMP ID10+2
29 03202 000401 ID11:  JMP .+1     I CHANGE TO LOOP TEST
30
31      I AUTO DECREMENT CHECK
32      I COUNT ALL THE AUTO DECREMENT REGISTERS FROM 20000-7
33      I CHECK THE VALUE AGAINST A ADD INSTRUCTION. PRESS
34      I CONTINUE AFTER A ERROR TO SCOPE THE FAILURE
35
36 03203 102520 ID14:  SUBZL 0,0      IC(0)=1, LOOP ON ERROR SWIT
37 03204 176400      SUB 3,3      IC(3)=0-7, REGISTER POINTER
38 03205 152620      SURZR 2,2
39 03206 151220      MOVZR 2,2
40 03207 151220      MOVZR 2,2      I STARTING VALUE OF 20000
41 03210 051430      STA 2,30,3    I SETUP AUTO DECREMENT REG
42 03211 060377      NIOP CPU     I SYNC AT 7-74
43 03212 027430      LDA 1,030,3   I SHOULD DECREMENT THE REG
44 03213 025430      LDA 1,30,3    IC(1)=VALUE OF AUTO DEC REG
45 03214 132014      ADC# 1,2,SZR  IC(2)=CORRECT VALUE+1
46 03215 063077      HALT          IC(3)+30=FAILING AUTO REG
47 03216 132014      ADC# 1,2,SZR
48 03217 102400      SUB 0,0      IC CLEAR TO LOOP TEST
49 03220 112404      SUB 0,2,SZR  I TEST FOR ZERO COUNT
50 03221 000767      JMP ID14+5
51 03222 175400      INC 3,3      I ADVANCE TO NEXT AUTO REG
52 03223 024111      LDA 1,C10
53 03224 136414      SUR# 1,3,SZR  I TEST FOR LAST AUTO
54 03225 000760      JMP ID14+2    I DECREMENT REGISTER
55 03226 000401 ID15:  JMP .+1     I CHANGE TO LOOP TEST
56

```

A 0250 ,MAIN

```

01
02
03      I A TEST OF AUTO INCREMENT
04      I EACH LOCATION IN THE BUFFER IS SET TO A ADDRESS
05      I PATTERN-1
06      I VIA THE AUTO INDEX REGISTERS. THE REGISTERS COUNTING
07      I ABILITY AS WELL AS THE DATA IS CHECKED, THE PROGRAM
08      I WILL LOOP ON ERROR IF CONTINUED AFTER A HALT.
09
10 03227 102000      ADC 0,0
11 03230 176400 ID20:  SUB 3,3
12 03231 030127      LDA 2,BUFF   I FIRST BUFFER LOCATION
13 03232 051420      STA 2,20,3    I SETUP AUTO INC REGISTER
14 03233 060377      NIOP CPU     I SYNC AT 7-A74
15 03234 053420      STA 2,020,3   I STORE C(2) VIA AUTO INC
16 03235 025420      LDA 1,20,3    IC(AUTO) TO C(1)
17 03236 146015 ID21:  ADC# 2,1,SZR I CHECK INCREMENT FEATURE
18 03237 000415      JMP ID23     I ITS OK
19 03240 063077      HALT          IC(1)=VALUE OF AUTO INC REG
20 03241 102400      SUB 0,0      IC(2)=CORRECT VALUE-1
21 03242 041000 ID22:  STA 0,0,2   IC(3)+20=AUTO REGISTER INVOLVED
22 03243 112400      SUB 0,2      IC(BUFFER) DESTROYED AFTER TEST
23 03244 024130      LDA 1,FIN    IC(1)=FINAL ADDRESS
24 03245 132414      SUR# 1,2,SZR  I TEST FOR END OF BUFFER
25 03246 000764      JMP ID20+2
26 03247 175400      INC 3,3      I GO TO NEXT AUTO REGISTER
27 03250 024111      LDA 1,C10
28 03251 136414      SUR# 1,3,SZR  I TEST FOR LAST ONE
29 03252 000757      JMP ID20+1
30 03253 000406      JMP ID24     I GO TO NEXT TEST
31
32 03254 025001 ID23:  LDA 1,1,2   I AUTO INC VALUE OK BUT
33 03255 132415      SUR# 1,2,SZR  I DATA STORED IS WRONG
34 03256 000764      JMP ID22     IC(1)=DATA FROM MEMORY
35 03257 063077      HALT          IC(2)=CORRECT VALUE
36 03260 000761      JMP ID22-1   IC(3)+20=AUTO REGISTER
37 03261 000401 ID24:  JMP .+1     I CHANGE TO LOOP TEST
38

```

A 0051 .MAIN

```

01
02      ;A TEST OF AUTO DECREMENT
03      ;EACH LOCATION IN THE BUFFER IS SET TO A ADDRESS
04      ;PATTERN+1 VIA THE AUTO DECREMENT REGISTERS, THE
05      ;REGISTERS COUNTING ABILITY AS WELL AS THE DATA IS
06      ;CHECKED,THE PROGRAM WILL LOOP ON ERROR IF CONTINUED
07      ;AFTER A HALT
08
09 03252 102000 ID30:  ADC 0,0      ;C(0)=1,LOOP ON ERR SWITCH
10 03253 176400      SUB 3,3      ;C(3)=0,FIRST INDEX USED
11 03254 030130      LDA 2,FIN    ;FINAL BUFFER ADDRESS
12 03255 051430      STA 2,30,3   ;TO AUTO DECREMENT REGISTER
13 03256 000377      NIOP CPU    ;SYNC AT 7-474
14 03257 053430      STA 2,030,3  ;STORE C(2) IN BUFFER VIA
15 03270 025430      LDA 1,30,3   ;C(1)=VALUE OF AUTO DEC REG
16 03271 132015 ID31:  ADC# 1,2,SNR ;C(2)=CORRECT VALUE+1
17 03272 000415      JMP ID33     ;C(3)+30=AUTO REGISTER
18 03273 063077      HALT        ;AUTO DECREMENT FAILED
19 03274 102400      SUB 0,0      ;CLEAR TO LOOP ON ERROR
20 03275 041000 ID32:  STA 0,0,2   ;CLEAR OUT THE PATTERN
21 03276 113000      ADD 0,2      ;C(2)=C(2)-1
22 03277 024127      LDA 1,BUFF   ;TEST FOR BEGIN OF BUFFER
23 03300 132414      SUB# 1,2,SZR  ;ADVANCE TO NEXT AUTO REG
24 03301 000764      JMP ID30+3   ;
25 03302 175400      INC 3,3      ;
26 03303 024111      LDA 1,C10    ;TEST FOR END
27 03304 136414      SUB# 1,3,SZR  ;AGAIN WITH THIS REGISTER
28 03305 000757      JMP ID30+2   ;GO TO NEXT TEST
29 03306 000406      JMP ID34     ;
30
31 03307 025377 ID33:  LDA 1,-1,2  ;AUTO DEC VALUE IS OK
32 03310 132415      SUB# 1,2,SNR  ;BUT THE DATA STORED IS NOT
33 03311 000764      JMP ID32     ;C(1)=DATA FROM BUFFER
34 03312 063077      HALT        ;C(2)=CORRECT VALUE
35 03313 000761      JMP ID32-1   ;C(3)+30=AUTO REGISTER
36
37 03314 000401 ID34:  JMP ,+1     ;CHANGE TO LOOP TEST
38

```

A 0052 .MAIN

```

01
02      ;A TEST OF AUTO INC USING ISZ
03      ;THE PROGRAM STORES A (-1) IN THE BUFFER.
04      ;A ISZ INSTRUCTION VIA AUTO INCREMENT REGISTER 27
05      ;THEN INCREMENTS THE BUFFER, IF A ERROR SHOULD OCCURE
06      ;PRESSING CONTINUE WILL PLACE THE PROGRAM IN A FAILING
07      ;LOOP UNTILL RESTARTED.
08
09 03315 102500 ID40:  SUBZL 0,0   ;C(0)=1,LOOP ON ERROR SWITCH
10 03316 126000      ADC 1,1      ;C(1)=-1
11 03317 030127      LDA 2,BUFF   ;FIRST LOCATION IN THE BUFFER
12 03320 034130      LDA 3,FIN    ;FINAL LOCATION IN THE BUFFER
13 03321 050027 ID41:  STA 2,27   ;SETUP AUTO INCREMENT REG
14 03322 045001      STA 1,1,2   ;STORE (-1) IN MEMORY
15 03323 000377      NIOP CPU    ;SYNC AT 7-74
16 03324 012027      ISZ 027     ;INCREMENT AND SKIP
17 03325 102401      SUB 0,0,SKP  ;ISZ FAIL TO SKIP
18 03326 113001      ADD 0,2,SKP  ;INC TO NEXT LOC
19 03327 063077      HALT        ;EXAMINE LOC 27
20 03330 156414      SUB# 2,3,SZR ;TEST FOR END OF BUFFER
21 03331 000770      JMP ID41     ;
22 03332 000441 ID42:  JMP ,+1    ;CHANGE TO LOOP TEST
23
24      ;A TEST OF AUTO INC/DEC
25      ;VIA REGISTER 34 THE BUFFER IS FILLED WITH JSR INST.
26      ;A JSR VIA REGISTER 25 IS EXECUTED,THE RETURN FROM
27      ;THE BUFFER IS CHECKED, PRESS CONTINUE TO LOOP ERROR.
28
29 03333 024425 ID44:  LDA 1,CIDJ  ;A CONSTANT (JSR 0,3)
30 03334 034127      LDA 3,BUFF   ;FIRST BUFFER LOCATION
31 03335 030130      LDA 2,FIN    ;FINAL BUFFER LOCATION
32 03336 050034      STA 2,34    ;C(34) POINTS TO THE BUFFER
33 03337 156400      SUB 2,3     ;C(3)= -WORDS IN BUFFER
34 03340 046034      STA 1,034   ;FILL BUFFER WITH (JSR)
35 03341 175444      INC 3,3,SZR  ;
36 03342 000770      JMP ,+2     ;
37
38 03343 102500 ID45:  SUBZL 0,0   ;C(0)=1,LOOP ON ERROR SWITCH
39 03344 024127      LDA 1,BUFF   ;
40 03345 044025      STA 1,25     ;SETUP AUTO INDEX REG
41 03346 006025      JSR 025     ;GO TO THE BUFFER
42 03347 174400      NEG 3,3     ;EFFECTIVELY SUBTRACT 1
43 03350 174000      COM 3,3     ;FROM C(3)
44 03351 136014      ADC# 1,3,SZR ;C(3)=POINT BUFFER ENTERED
45 03352 102401      SUB 0,0,SKP  ;C(2)=CORRECT-1
46 03353 107001      ADD 0,1,SKP  ;C(25)=AUTO INC VALUE
47 03354 063077      HALT        ;25=AUTO INC REGISTER
48 03355 132014      ADC# 1,2,SZR ;TEST FOR END OF BUFFER
49 03356 000767      JMP ID45+2   ;
50 03357 101011 ID46:  MOV# 3,0,SKP ;CHANGE TO LOOP TEST
51 03358 005400 CIDJ:  JSR 0,3   ;A CONSTANT
52

```

A 0053 .MAIN

```

01
02 03361 024127 ID50: LDA 1,BUFF /FIRST LOCATION OF THE BUFF
03 03362 030130 LDA 2,FIN /FINAL LOCATION OF THE BUFF
04 03363 102520 SURZL 0,P /C(0)=1, LOOP ON ERROR SWIT
05 03364 050033 STA 2,33 /SETUP AUTO DECREMENT
06 03365 000033 JSR 033 /GO TO THE BUFFER
07 03366 156414 SUB# 2,3,SZR /C(3)=PC FROM BUFFER
08 03367 102401 SUR 0,0,SKP /C(2)=INITIAL VALUE OF AUTO
09 03370 112401 SUB 0,2,SKP /DEC REGISTER 33
10 03371 053077 HALT /PROG WILL LOOP ON ERROR
11 03372 132014 ADC# 1,2,SZR /TEST FOR BEGIN OF BUFF
12 03373 000771 JMP ID50+3
13
14 03374 034402 DGCA: LDA 3,0+2 /TEST DSZ COUNT IN
15 03375 152001 ADC 2,2,SKP /EACH MEMORY MODULE,
16 03376 004000 4000 /C(2)=ERROR SWITCH.
17 03377 020125 LDA 0,MSIZE
18 03400 116432 SURZ# 0,3,SZC
19 03401 000421 JMP LAST
20 03402 102620 SUBZR 0,P
21 03403 143005 DGCA: ADD 2,0,SNR
22 03404 000413 JMP DGCR
23 03405 041400 STA 0,0,3
24 03406 060377 NIUP CPU /ISYNC 7-A74
25 03407 015400 DSZ 0,3
26 03410 000401 JMP 0+1
27 03411 025400 LDA 1,0,3
28 03412 102015 ADC# 1,0,SNR /C(1)=RESULT OF DSZ
29 03413 000770 JMP DGCA /C(0)=VALUE BEFORE DSZ
30 03414 152400 SUB 2,2
31 03415 060377 HALT
32 03416 000755 JMP DGCA
33 03417 020757 DGCA: LDA 0,DGCA+2 /GO TO NEXT LOCATION
34 03420 117000 ADD 0,3
35 03421 000754 JMP DGCA+1
36 03422 010203 LAST: ISZ PASS /INCREMENT PASS CNT
37 03423 101000 MOV 0,0
38 03424 024046 LDA 1,EGGS
39 03425 125005 MOV 1,1,SNR
40 03426 002410 JMP 0+0
41 03427 014051 DSZ EGGS+3
42 03430 000407 JMP 0+7
43 03431 062677 INRST
44 03432 034052 LDA 3,EGGS+4
45 03433 020051 LDA 0,EGGS+3
46 03434 041776 STA 0,-2,3
47 03435 001400 JMP 0,3
48 03436 000504 REG1
49 03437 010050 ISZ EGGS+2
50 03440 000140 JMP #MAIN
51 03441 000171 JSR #ICAT
52 03442 002140 JMP #MAIN

```

A 0054 .MAIN

```

01
02 03443 047503 .TXT /COPYRIGHT(C)DGC,1960,70,71,73,74,75,76
03444 054520
03445 044522
03446 044107
03447 024124
03450 024503
03451 043504
03452 026103
03453 034461
03454 034466
03455 033454
03456 026060
03457 030467
03458 033454
03461 026063
03462 032067
03463 033454
03464 026065
03465 033067
03 03466 046101 ALL RIGHTS RESERVED
03467 020114
03470 044522
03471 044107
03472 051524
03473 051040
03474 051505
03475 051105
03476 042526
03477 000104
04
05 03500 003521 PRGEN0: DINT+20
06 03501 154305 DIRT: .TXTE IFXFCISERR01
03502 151305
03503 140703
03504 142523
03505 030322
03506 000270
07 03507 000000 000000
08 03510 000200 000000
09 03511 002002 000000
10 03512 000000 000000
11 03513 000000 000000
12 03514 000000 000000
13 03515 000000 000000
14 03516 000000 000000
15
16 .END

```

0055 MAIN

ACTIO	000105	5/45	8/06	8/09	8/31	8/32	9/32	12/51	13/26
BEG	000500	14/02	16/20	16/49	17/17				
BEG1	000504	5/06	8/04						
HUF20	000132	8/09	53/48						
BUFF	000127	6/10	25/31	27/26					
		6/07	19/15	19/35	20/05	20/37	20/40	21/11	21/25
		22/21	22/29	23/18	24/19	24/36	25/13	26/14	26/32
		26/06	28/16	31/30	32/21	32/20	35/30	36/12	36/21
		37/10	38/14	39/20	39/10	39/39	40/12	40/19	40/40
		40/45	41/10	41/17	42/09	42/23	43/16	43/17	43/25
		43/36	44/02	44/10	45/20	46/11	46/23	50/12	51/22
		52/11	52/30	52/39	53/02				
C10	000111	5/49	12/13	49/26	49/52	50/27	51/26		
C100P	000142	6/18	9/03						
C1347	000306	44/07	44/35						
C1355	000126	5/46	12/36	14/16	14/40	15/30	16/00	17/31	18/07
		18/18							
C170	000104	5/44							
C174X	000137	6/15	9/31						
C177	000110	5/46	12/30	15/03	16/10				
C200	000143	6/19	9/27						
C2002	000174	6/46	9/19						
C202	000173	6/45	9/16						
C24	000743	12/14	12/24						
C3	000133	6/11	10/03	24/28	25/22	27/10			
C3000	000141	6/17	9/02						
C37	000150	6/26	12/00						
C377	000135	6/13	24/40	25/34	26/40	27/34	29/00	29/30	33/10
		39/11							
C4	000124	6/04	36/20						
C40	000126	6/06	12/10						
C400	000172	6/44	9/22						
C5	000134	6/12	26/24	31/11	31/13	35/10			
C5252	0001476	20/07	20/24						
CF00	000130	6/14	45/23	45/36	46/26	47/04			
CIDJ	000360	52/29	52/51						
CINC	000030	20/08	20/22						
CJDF1	000255	43/18	43/51						
CJMP1	0001610	23/19	23/30						
CJSR1	0001571	22/22	22/41						
CJSR2	0002700	42/12	42/31						
CJSR3	0002701	42/11	42/32						
CJSR4	0002702	42/25	42/33						
CLDAP	0001645	24/25	24/51						
CLOCK	0001201	5/23	15/53						
CM5	000123	6/03	35/12						
CPINC	0002031	20/13	20/23						
CXRAM	000112	5/50	12/30	13/40	14/42	15/23	16/02	17/10	17/54
		18/11							
DEF0	0002460	37/09	37/22						
DEF10	0002477	38/13							
DEF11	0002507	38/21	38/30						
DEF14	0002522	39/09	39/24						
DEF15	0002542	39/25							
DEF20	0002543	39/38	39/52						
DEF21	0002562	39/53							
DEF30	0002563	40/11							
DEF31	0002572	40/19	40/20						
DEF32	0002604	40/29							

0056 MAIN

DEF34	0002605	40/38							
DEF35	0002614	40/45	40/55						
DEF36	0002627	40/56							
DEF40	0002630	41/09							
DEF41	0002637	41/17	41/29						
DEF42	0002654	41/30							
DEF50	0002713	43/15							
DEF51	0002726	43/26							
DEF52	0002733	43/32							
DEF53	0002741	43/39	43/49						
DEF54	0002754	43/50							
DEF60	0003017	45/20	45/47						
DEF61	0003034	45/34	45/42						
DEF62	0003045	45/27	45/44						
DEF63	0003051	45/49							
DEF64	0003065	46/23	47/17						
DEF65	0003082	47/02	47/13						
DEF66	0003114	46/30	47/12	47/22	47/24				
DEF67	0003123	47/08	47/20						
DGCA	0003374	53/14	53/33	53/35					
DGCB	0003417	53/22	53/33						
DGCM	0003423	53/21	53/29	53/32					
DIRT	0003501	5/03	54/05	54/06					
DISMI	0001700	6/42	12/50	12/55	13/19	13/25	13/30	13/45	14/10
		14/27	14/47	14/50	15/30	15/51	15/57	16/19	16/25
		16/44	16/48	16/50	17/15	17/33	17/39	18/05	18/09
		18/26	18/30						
DSZ0	0002246	33/09							
DSZ1	0002252	33/14	33/22						
DSZ10	0002305	34/15	34/20	34/34					
DSZ11	0002323	34/30	34/41	34/47					
DSZ12	0002330	34/25	34/36						
DSZ13	0002336	34/20	34/43						
DSZ14	0002343	34/37	34/49						
DSZ2	0002265	33/32							
DSZ20	0002344	35/10	35/27						
DSZ21	0002350	35/14							
DSZ22	0002366	35/28							
DSZ3	0002270	33/35	33/40	33/45					
DSZ30	0002367	35/36							
DSZ31	0002373	35/40	35/49	35/56					
DSZ32	0002406	35/43	35/46	35/51					
DSZ33	0002414	35/50	35/57						
DSZ4	0002277	33/38	33/42						
DSZ40	0002415	36/09	36/46						
DSZ41	0002426	36/19							
DSZ42	0002431	36/23	36/33						
DSZ43	0002444	36/26	36/28	36/30	36/32	36/35			
DSZ5	0002303	33/41	33/46						
DT030	0002000	6/53	54/08						
EGGS	0000046	5/10	5/11	9/17	53/30	53/41	53/44	53/45	53/49
ENDIT	0002733	6/42	12/20						
EXCH	0003052	46/11	46/21						
FIN	0001300	6/00	9/20	19/16	19/36	20/06	20/30	21/12	22/23
		23/17	24/20	25/14	25/41	26/15	27/27	28/07	31/37
		32/20	35/37	36/11	36/39	37/11	38/15	39/50	40/13
		40/39	41/11	42/10	42/36	43/15	44/04	45/45	46/12
		47/15	50/23	51/11	52/12	52/31	53/03		
FIN20	000131	6/09	9/30	24/45	26/33				

0057 .MAIN

MSR	001026	13/32	14/07	14/13	14/15
MSR1	001044	5/21	14/02		
MSR3	001062	13/39	14/16		
MSR8	000073	5/35	13/33	13/34	
MSRCH	000074	5/36	13/35	13/42	
ICAT	000171	6/43	9/25	53/51	
IDP	003131	48/07	48/28		
ID1	003144	48/15	48/22		
ID1P	003156	49/09	49/24	49/28	
ID11	003202	49/29			
ID14	003203	49/36	49/50	49/54	
ID15	003226	49/55			
ID2	003155	48/26	48/29		
ID2P	003230	50/11	50/25	50/29	
ID21	003236	50/17			
ID22	003242	50/21	50/34	50/36	
ID23	003254	50/18	50/32		
ID24	003261	50/30	50/37		
ID30	003262	51/09	51/24	51/28	
ID31	003271	51/16			
ID32	003275	51/20	51/33	51/35	
ID33	003307	51/17	51/31		
ID34	003314	51/29	51/37		
ID40	003315	52/09			
ID41	003321	52/13	52/21		
ID42	003332	52/22			
ID44	003333	52/29			
ID45	003343	52/38	52/49		
ID46	003357	52/50			
ID50	003361	53/02	53/12		
INTR	000071	5/04	12/01		
ISAV0	000113	5/51	12/01	12/22	
ISAV1	000114	5/52	12/02	12/23	
ISAV2	000115	5/53	12/03	12/24	
ISAV3	000116	5/54	12/04	12/25	
ISAVC	000117	5/55	12/06	12/20	
ISZ0	002034	29/07			
ISZ1	002040	29/11	29/19		
ISZ10	002113	30/23	30/30	30/42	30/47
ISZ11	002130	30/43	30/44		
ISZ12	002134	30/39	30/48	30/52	30/56
ISZ13	002141	30/34	30/53		
ISZ14	002145	30/49	30/57		
ISZ2	002053	29/29			
ISZ20	002146	31/11	31/29		
ISZ21	002153	31/16			
ISZ22	002171	31/15	31/30		
ISZ3	002057	29/33	29/41		
ISZ30	002172	31/36			
ISZ31	002175	31/40	31/49	31/56	
ISZ32	002211	31/43	31/46	31/51	
ISZ33	002217	31/50	31/57		
ISZ4	002072	30/07			
ISZ40	002220	32/16	32/41		
ISZ41	002231	32/28	32/33		
ISZ42	002237	32/31	32/35		
ISZ5	002075	30/10	30/15	30/20	
ISZ6	002104	30/13	30/17		
ISZ7	002110	30/16	30/21		

0058 .MAIN

ITAR	000054	5/19	12/18		
JMP10	001572	23/16	23/28		
JMP11	001600	23/22	23/30		
JSP10	001546	22/20			
JSR11	001556	22/29	22/39		
JSR20	002055	42/09			
JSR21	002661	42/13	42/20		
JSR22	002072	42/23			
JSR23	002703	42/34			
KRAN	000072	5/34	14/43	14/49	
LAST	003422	53/19	53/36		
LSR	001065	5/19	14/23		
LSR1	001116	14/41	14/48		
LSR2	000070	5/32	14/36	14/37	
LSRCH	000071	5/33	14/38	14/44	
LT	000101	5/41	12/32	15/25	16/04
MATN	000140	6/16	8/07	9/35	18/13
MEMS7	000556	6/54	9/02	9/13	53/50
MSIZE	000125	6/05	9/05	9/08	53/52
		44/16	53/17	9/09	9/10
NEGFI	001707	25/19	25/48	9/11	9/14
NEGJ	001755	27/15		9/20	9/28
NEGJ1	001757	27/18	27/30		
NEGJ2	001766	27/26	27/42		
NEGJ3	001773	27/16	27/19	27/21	27/32
NEGJ0	002010	27/24	27/46		
NEGJX	002006	27/22	27/44		
NEGJY	002007	27/15	27/45		
NEGND	001711	25/20	25/50		
NEGK	001650	25/13	25/18		
NEGK1	001660	25/22	25/45		
NEGK2	001671	25/20	25/23	25/25	25/29
NEGKX	001710	25/26	25/49	25/32	25/44
ORT	000100	5/40			
PASS	000203	6/56	8/05	53/36	
PINC	002011	28/06			
PINC1	002014	28/09	28/12		
PINC2	002022	28/15			
PINC3	002025	28/19	28/24		
PINC4	002032	28/15	28/24		
POSFI	001646	24/33	24/52		
POSJ	001712	26/14	26/20		
POSJ1	001723	26/24	26/36		
POSJ2	001732	26/32	26/45		
POSJ3	001740	26/22	26/25	26/27	26/38
POSJF	001752	26/28	26/49	26/50	
POSJN	001754	26/30	26/51		
POSJX	001751	26/16	26/48		
POSJY	001753	26/37	26/50		
POSJZ	001750	26/21	26/47		
POSND	001647	24/35	24/53		
POSX	001611	24/19	24/24		
POSX1	001621	24/28	24/49		
POSX2	001631	24/26	24/30	24/32	24/37
PRAN	000077	5/39	12/31	12/37	24/48
PRGEN	003500	6/07	6/08	6/09	6/10
PTPS0	000035	8/35	10/17		54/05
PTPS1	000032	8/13	10/14		
PTPS2	000041	8/37	10/22		

0059 MAIN

PTRSU	000643	8/15	10/24						
PUN	000744	5/22	12/30						
PUN1	000776	12/35	13/03						
PUNCH	000102	5/42	10/10	12/33	13/03	13/05	13/14	14/08	
PUNX	000775	12/42	12/56						
RANFC	002762	44/06	44/21	44/24	44/33				
RANFL	002756	44/02							
READE	000075	5/37	10/20	13/15	13/36	14/03	14/05	14/09	
RER	000107	5/47							
RPOIF	000103	5/43	13/16	14/10	16/41	17/25			
RRAN	000076	5/38	13/41	14/17					
RTCSU	000620	8/21	10/03						
STA0	001422	6/16	6/43	19/15					
STA01	001426	19/20							
STA02	001437	19/35							
STA03	001443	19/40							
STA04	001454	20/05							
STA05	001462	20/11							
STA06	001466	20/16							
STA10	001477	20/36							
STA11	001502	20/40	20/46						
STA12	001511	20/48	20/54						
STA13	001520	20/56							
STA20	001522	21/11							
STA21	001524	21/14	21/22						
STA22	001535	21/25	21/32						
STP	000144	6/20	10/19	12/39					
STT	000145	6/21	10/12	15/33					
TEM	000122	6/02	29/11	29/13	29/15	29/33	29/35	29/37	30/10
		30/12	30/17	30/35	30/37	30/40	30/50	30/54	31/16
		31/18	31/19	31/20	31/21	31/22	31/23	33/14	33/16
		33/18	33/35	33/37	33/42	34/21	34/23	34/26	34/38
		34/43	35/14	35/16	35/17	35/18	35/19	35/20	35/21
		37/12	37/14	37/15	38/21	38/23	39/12	39/15	39/17
		39/40	39/43	41/20	41/22	43/24	43/32	43/35	43/40
		43/47	44/03	44/22	44/29	44/30	44/31	45/31	46/34
		47/05	47/06	47/11	47/20				
TIN	000146	6/22	10/33	14/32	14/33	14/35			
TINEY	000147	6/23	10/29	10/34	14/23				
TRAN	000120	5/56	15/24	15/31					
TTIS0	000647	8/39	10/29						
TTISU	000651	8/17	10/31						
TTOS0	000626	8/41	10/10						
TTOSU	000623	8/19	10/07						
TYO	001145	5/20	15/23						
TYO1	001168	15/29	15/41						
TYPE	000121	5/57	10/11	15/27	15/41	15/43			
XRAND	001121	5/50	15/02	15/12					
MSR	001273	17/02	17/22	17/20	17/30				
MSR1	001311	5/29	17/17						
MSR3	001327	17/09	17/31						
MSRB	000156	6/32	17/03	17/04					
MSRC	000157	6/33	17/05	17/12					
PRAN	000164	6/30	17/55	18/00					
LSR	001332	5/27	17/35						
LSR1	001363	17/53	18/07						
LSRB	000162	6/36	17/48	17/49					
LSRC	000163	6/37	17/50	18/02					
PRAN	000153	6/29	16/03	16/09					

0050 MAIN

PTR	000053	6/58	11/19	16/16	16/31	16/32	16/34	16/35	16/36
PTR	000052	6/59	11/26	11/28	16/17	16/18	16/24	16/46	16/47
		16/57	17/02	17/29					
PUN	001206	5/30	16/02						
PUN1	001236	16/07	16/20						
PUNC	000154	6/30	11/22	16/05	16/28	16/30	16/39	17/23	
PUNX	001235	16/14	16/26						
READ	000155	5/31	11/30	16/40	17/06	17/18	17/20	17/24	
RRAN	000152	6/28	17/11	17/32					
STP	000151	6/27	11/23	16/11					
STT	000167	6/41	11/15	18/21					
TIN	000161	6/35	11/06	17/44	17/45	17/47			
TINE	000160	6/34	11/02	11/07	17/35				
TISS	000656	8/49	11/02						
TISSU	000660	8/27	11/04						
TOSU	000670	8/51	11/13						
TOSU	000665	8/29	11/10						
TPSD	000677	8/45	11/21						
TPSU	000674	8/23	11/10						
TRAN	000166	6/40	18/12	18/19					
TRSD	000703	8/47	11/26						
TRSU	000705	8/25	11/20						
TTI	000050	7/01	11/04	17/38	17/40	17/41	17/42		
TTO	000051	7/02	11/11	18/25	18/32	18/33	18/35	18/36	18/37
TYO	001305	5/28	18/11						
TYO1	001407	18/17	18/29						
TYPE	000165	5/39	11/14	18/15	18/29	18/31			
PRAN	001132	15/10	15/11						