

DataGeneral

**DIAGNOSTIC
LISTING**

LISTING

096-000144-07

PROGRAM

EXERCISER

TAPE

095-000012-07

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.



0

0

0

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

```

```

1 NAME: EXERCISER,SR          PART NUMBER: 094-000011
2
3 DESCRIPTION: EXERCISER
4
5 REVISION HISTORY:
6
7 REV.          DATE
8
9 00           05/01/69
10 01           03/18/71
11 02           09/18/73
12 03           11/30/73
13 04           01/25/74
14 05           06/07/74
15 06           04/18/75
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

```

```

1 COPYRIGHT (C) DATA GENERAL CORPORATION, 1969, 1970, 1971,
2 1973, 1974, 1975
3 ALL RIGHTS RESERVED.
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

```

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

```

12. MACHINE REQUIREMENTS

```

12.1 STANDRED NOVA 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 =000002
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 PTP
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 000002
14.3 PRESS START
14.4 THE PROGRAM WILL RUN UNTILL MANUALLY STOPPED
14.5 OR A ERROR IS DETECTED.
14.5.1 IN/OUT EQUIPMENT TESTING
14.5.1.1 BY SETTING SWITCHES 0-4 THE CORRESPONDIN
14.5.1.2 DEVICE WILL BE ACTIVATED, THE SWITCHS AR
14.5.1.3 READ AT THE COMPLETION OF EACH PASS OF T
14.5.1.4 INSTRUCTION TEST, IF A CONSOLE SWITCH HA
14.5.1.5 BEEN SET/CLEARED DURING THE INSTRUCTION
14.5.1.6 TEST THE DEVICE WILL BE ACTIVATED/DEACTI
14.5.1.7 AT THE END OF A PASS.
14.5.1.8 TO TEST THE HIGH SPEED READER AND PUNCH
14.5.2 RAISE THE READER LEVER
14.5.2.1 SET SWITCHES 0 AND 1
14.5.2.2 WHEN 5 TO 6 FEET OF TAPE HAVE
14.5.2.3 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 0011  MAIN
01 00612 040113 INTR: STA 0,ISAV0
02 00613 044114 STA 1,ISAV1
03 00614 050115 STA 2,ISAV2
04 00615 054116 STA 3,ISAV3
05 00616 175200 MOVR 3,3
06 00617 054117 STA 3,ISAVC
07 00620 061477 INTA 0
08 00621 030151 LDA 2,C37
09 00622 113520 ANOZL 0,2,
10 00623 024126 LDA 1,C40
11 00624 107020 ANDR 0,1
12 00625 133220 ADDR 1,2
13 00626 024111 LDA 1,C10
14 00627 034415 LDA 3,C24
15 00630 172433 SUBZ 3,2,SNC
16 00631 132423 SUBZ 1,2,SNC
17 00632 063077 HALT
18 00633 003053 JMP #ITAR,2
19
20 00634 020117 ENDIT: LDA 0,ISAVC
21 00635 101100 MOVL 0,0
22 00636 020113 LDA 0,ISAV0
23 00637 024114 LDA 1,ISAV1
24 00640 030115 LDA 2,ISAV2
25 00641 034116 LDA 3,ISAV3
26 00642 000177 INTEN
27 00643 002000 JMP #0
28
29 00644 00024 C24: PRAN #CXRAND
30 00645 006112 PUN: LDA 1,L,T
31 00646 000077 LDA 2,PUNCH
32 00647 024101 LDA 1,L,T
33 00650 030102 AOCZ 1,2,SZC
34 00651 132052 JMP PUN1
35 00652 000425 LDA 0,C130525
36 00653 020106 STA 0,PRAN
37 00654 040077 LDA 1,C177
38 00655 024110 LDA 1,C177
39 00656 034145 AND 1,2,SNR
40 00657 133405 MOVL 0,0,SZR
41 00658 175005 MOV 3,3,SNR
42 00661 000415 JMP PUNX
43 00662 000403 JMP #+3
44 00663 000000
45 00664 000000
46
47 00665 000213 NIOC PTP
48 00666 063412 SKPBN PTR
49 00667 063712 SKPOZ PTR
50 00670 002171 JMP #DISMISS
51 00671 020105 LDA 0,ACTION
52 00672 101100 MOVL 0,0
53 00673 101102 MOVL 0,0,SZC
54 00674 000112 NIOS PTR
55 00675 002171 JMP #DISMISS
56 00676 102400 PUNX: SUB 0,0
57
A 0012  MAIN
01
02
03 00677 010102 PUN1: ISZ PUNCH
04 00678 101001 MOV 0,0,SMP
05 00679 010102 ISZ PUNCH
06 00682 063413 SKPBN PTP
07 00683 063013 SKPBN PTP
08 00684 063077 HALT
09 00685 061113 DOAS 0,PTP
10 00686 063513 SKPBN PTP
11 00687 063713 SKPOZ PTP
12 00690 063077 HALT
13
14 00711 020102 LDA 0,PUNCH
15 00712 024075 LDA 1,READER
16 00713 030103 LDA 2,RPOIF
17 00714 100400 SUB 0,1
18 00715 133404 AND 1,2,SZR
19 00716 002171 JMP #DISMISS
20
21
22
23 00717 063412 SKPBN PTR
24 00720 063712 SKPOZ PTR
25 00721 002171 JMP #DISMISS
26 00722 020105 LDA 0,ACTION
27 00723 101100 MOVL 0,0
28 00724 101112 MOVL 0,0,SZC
29 00725 000112 NIOS PTR
30 00726 002171 JMP #DISMISS
31
32 00727 064512 HSR: DIAS 1,PTR
33 00730 020073 LDA 0,HSRB
34 00731 044073 STA 1,HSRB
35 00732 040074 LDA 2,READER
36 00733 030075 MOV 0,0,SZR
37 00734 101004 MOVZRN 2,2,SNR
38 00735 151235 JMP HSR3
39 00736 000425 JSR #CXRAND
40 00737 006112 RRAN
41 00740 000076 LDA 1,HSRCH
42 00741 024074 SUB# 0,1,SZR
43 00742 100414 HALT
44 00743 063077 JMP #DISMISS
45 00744 002171
46
47

```

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS

FINTEERRUPT DISPATCH
 ISAVE INTERRUPTS


```

A 0019 ,MAIN
01
02
03          IFILL AND CHECK BUFFER, A BUFFER OF ALTERNATING BITS
04
05 01363 034130 STA04: LDA 2,BUFF      }FIRST LOCATION OF THE BUFFER
06 01364 034131          LDA 3,FIN      }LAST LOCATION OF THE BUFFER
07 01365 020420          LDA 0,C52525
08 01366 104000          COM 0,1
09 01367 045000          STA 1,0,2      }BUFFER = 125252
10 01370 041001          STA 0,1,2      }BUFFER = 052525
11 01371 021000 STA05: LDA 0,0,2      }THIS PAIR OF WORDS
12 01372 151400          INC 2,2      }IS MOVED THROUGH THE
13 01373 156015          ADC# 2,3,SNR   }BUFFER
14 01374 000774          JMP ,-4
15
16 01375 021000 STA06: LDA 0,0,2      }GET BUFFER'S LAST AND
17 01376 025377          LDA 1,-1,2     }LAST-1 WORDS
18 01377 123020          ADDZ 1,0      }C(0) SHOULD=177777
19 01400 100014          COM# 0,0,SZR   }BUFFER SHOULD CONTAIN
20 01401 063077          HALT          }ALTERNATING WORDS
21 01402 101012          MOV# 0,0,SZC   }EXAMINE BUFFER FOR ERRORS
22 01403 063077          HALT          }HOW DID CARRY CHANGE?
23 01404 101011          MOV# 0,0,SKP   }SKIP OVER CONSTANT
24 01405 052525 C52525: 052525
25
26
27          }MAKE SURE THAT ALL LOCATIONS IN THE BUFFER EXIST.
28          }STORE A ADDRESS PATTERN (EACH LOCATION CONTAINS ITS
29          }ADDRESS),
30          }IF STA WERE ATTEMPTING TO STORE A 2400 IN LOCATION
31          }2400
32          }BUT INSTEAD STORED IN 2000, THE CONTENTS OF 2000 WOULD
33          }CONTAIN 2400,ONE SHOULD SUSPECT MA BIT 7 OR OTHER
34          }MEMORY ADDRESS DECODING.
35
36 01406 102400 STA10: SUB 0,0      }C(0) NOT USED THIS TEST
37 01407 034130          LDA 3,BUFF   }FIRST LOCATION OF BUFFER
38 01410 024131          LDA 1,FIN    }LAST LOCATION OF BUFFER
39
40 01411 055400 STA11: STA 3,0,3     }STORE THE ADDRESS AT ADDRESS
41 01412 031400          LDA 2,0,3     }LOAD C(2) WITH DATA JUST
42 01413 156414          SUB# 2,3,SZR   }STORED, C(3)=DATA SENT,
43 01414 063077          HALT          }C(3)=ADDRESS,C(2)=RECEIVED
44 01415 175400          INC 3,3      }INC TO NEXT ADDR
45 01416 136414          SUB# 1,3,SZR   }TEST FOR END OF BUFF
46 01417 000772          JMP STA11
47
48 01420 034130 STA12: LDA 3,BUFF   }CORE SHOULD BE FILLED WITH
49 01421 031400          LDA 2,0,3     }ADDRESS PATTERN, GET WORD
50 01422 156414          SUB# 2,3,SZR   }C(2) SHOULD EQUAL C(3)
51 01423 063077          HALT          }LOOK FOR PATTERN OF ERRORS
52 01424 175400          INC 3,3      }GO TO NEXT ADDRESS
53 01425 136414          SUB# 1,3,SZR   }TEST FOR END OF BUFFER
54 01426 000773          JMP STA12+1 }MORE TO GO
55
56 01427 101014 STA13: MOV# 0,0,SZR   }CHANGE TO (JMP STA10) FOR LOOP
57 01430 063077          HALT          }C(0) NOT USED THIS TEST!
58

```

```

A 0020 ,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          }OCCURS
07          }EXAMINE THE BUFFER AND TRY TO FIND A PATTERN OF ERRORS
08          }FLOA AND STA HAVE PREVIOUSLY WORKED WITH A SIMILAR
09          }PATTERN.
10
11 01431 034130 STA20: LDA 3,BUFF     }C(3)=FIRST BUFFER LOCATION
12 01432 024131          LDA 1,FIN    }C(1)=LAST BUFFER LOCATION
13
14 01433 160005 STA21: COM 3,0,SNR   }THE CONTENTS OF 3 SHOULD
15 01434 063077          HALT          }ALWAYS PRODUCE NOT ZERO
16 01435 041400          STA 0,0,3     }STORE LOGICAL COMPLEMENT
17 01436 031400          LDA 2,0,3     }OF ADDRESS IN ADDRESS
18 01437 142414          SUB# 2,0,SZR   }C(2) SHOULD BE SAME AS
19 01440 063077          HALT          }C(0) WHICH WAS STORED,
20 01441 175400          INC 3,3      }INC TO NEXT ADDRESS
21 01442 136414          SUB# 1,3,SZR   }TEST FOR END OF BUFFER
22 01443 000770          JMP STA21    }NOT YET END OF BUFFER
23
24
25 01444 034130 STA22: LDA 3,BUFF     }RESET C(3) TO FIRST LOCATION
26 01445 031400          LDA 2,0,3     }IN THE BUFFER, WORD FROM
27 01446 140000          COM 2,0      }BUFFER SHOULD BE COMP OF
28 01447 116414          SUB# 0,3,SZR   }ADDRESS IN 3, C(2)=BUFFER WD
29 01450 063077          HALT          }C(0)=ITS COMP, C(3)=ADDRESS
30 01451 175400          INC 3,3      }INCREMENT TO NEXT ADDRESS
31 01452 136414          SUB# 1,3,SZR   }TEST FOR END OF BUFFER
32 01453 000772          JMP STA22+1 }MORE WORDS TO CHECK
33
34 01454 000401          JMP ,+1     }REPLACE WITH (JMP STA20) TO
35          }LOOP
36

```

```

A 0021 ,MAIN
01
02      JA 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 01455 102520 JSR10: SUBZL 0,0      !LOOP ON ERR SWITCH
21 01456 034130      LDA 3,BUFF      !C(3)=FIRST BUFFER LOCATION
22 01457 030421      LDA 2,CJSR10    !CONSTANT (JSR 1,3)
23 01460 024131      LDA 1,FIN      !C(1)=FINAL BUFFER LOCATION
24 01461 051400      STA 2,0,3      !FILL THE BUFFER
25 01462 175400      INC 3,3        !WITH JSR INSTRUCTIONS
26 01463 166414      SUB# 3,1,SZR    !TEST FOR END OF THE BUFFER
27 01464 000775      JMP ,+3        !MORE WORDS TO STORE
28
29 01465 030130 JSR11: LDA 2,BUFF      !C(2)=FIRST BUFFER LOCATION
30 01466 005000      JSR 0,2        !GO TO THE BUFFER| C(3)=
31 01467 063077      HALT          !RETURN ADDR, FAIL TO PC XFER
32 01470 150014      ADC# 2,3,SZR    !CHECK NEW C(3) FROM JSR IN
33                          !BUFF
34 01471 063077      HALT          !C(3)SHOULD=C(2)+1, WENT WRONG
35 01472 150014      ADC# 2,3,SZR
36 01473 102400      SUB 0,0
37 01474 113000      ADD 0,2
38 01475 146414      SUB# 2,1,SZR    !NEXT BUFF LOC,TEST FOR END
39 01476 000770      JMP JSR11+1    !MORE LOCATIONS IN BUFF
40 01477 101011      MOV# 0,0,SKP   !SKIP OVER CONSTANTS
41 01500 005401 CJSR10: JSR 1,3
42

```

```

A 0022 ,MAIN
01
02      JA TEST OF JMP
03      !THIS PROGRAM WILL EXECUTE A JMP TO EVERY LOCATION IN
04      !THE BUFFER.
05      !THE BUFFER CONTAINS A (JSR 1,3) INSTRUCTION IN EACH
06      !LOCATION.
07      !THE VALUE IN C(3) IS SET SUCH THAT THE JSR IN THE
08      !BUFFER WILL
09      !RETURN TO THE MAIN PROGRAM, THE PROGRAM THEN CHECKS
10      !THE PC
11      !PROGRAM COUNTER) STORED BY THE BUFFERS' JSR
12      !INSTRUCTION, THIS
13      !PC SHOULD BE ONE GREATER THAN THE LOCATION JUMPED TO,
14
15
16 01501 126520 JMP10: SUBZL 1,1      !LOOP ON ERR SWIT
17 01502 020131      LDA 0,FIN      !FINAL ADDRESS OF BUFFER
18 01503 030130      LDA 2,BUFF      !FIRST ADDRESS OF BUFFER
19 01504 034413      LDA 3,CJMP11    !C(3)=A RETURN TO THE PROG
20 01505 001000      JMP 0,2        !GO TO A LOCATION IN BUFFER
21 01506 003077      HALT          !NO JMP? INDEX+1 IN BUFFER?
22 01507 150014 JMP11: ADC# 2,3,SZR    !CHECK PC STORED BY BUFFER JSR
23 01510 063077      HALT          !C(2)=JMP ADDRESS,C(3)=JSR PC
24 01511 150014      ADC# 2,3,SZR
25 01512 126400      SUB 1,1
26 01513 133000      ADD 1,2
27 01514 142414      SUB# 2,0,SZR    !TEST FOR END OF THE BUFFER
28 01515 000707      JMP JMP10+3    !NOT YET END OF BUFFER
29 01516 101011      MOV# 0,0,SKP   !SKIP OVER THE CONSTANT
30 01517 001506 CJMP11: JMP11-1    !CONSTANT

```

```

A 0023 ,MAIN
01
02      ;A 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 REFERANCES 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 01520 034130 POSX:  LDA 3,BUFF      ;FIRST BUFFER LOCATION
20 01521 024131      LDA 1,FIN        ;LAST BUFFER LOCATION
21 01522 055400      STA 3,0,3      ;STORE A ADDRESS PATTERN
22 01523 175400      INC 3,3        ;IN THE BUFFER
23 01524 136414      SUB# 1,3,SZR    ;C(BUFFER)=ADDRESS OF BUFF
24 01525 000775      JMP POSX+2     ;EXAMPLE: C(3015)=3015
25 01526 030426      LDA 2,CLODAPX  ;CONSTANT (LDA 2,0,3) INST
26 01527 050411      STA 2,POSX2    ;INITIALIZE THE INSTRUCTION
27
28 01530 020134 POSX1: LDA 0,C3      ;ADD 3 TO THE INDEX
29                          ;DISPLACEMENT=-
30                          ;MENT OF THE LDA INSTRUCTION
31 01531 024407      LDA 1,POSX2
32 01532 107000      ADD 0,1
33 01533 044405      STA 1,POSX2
34 01534 020421      LDA 0,POSFIN
35 01535 100032      ADCZ# 0,1,SZC
36 01536 000420      JMP POSND
37 01537 034130      LDA 3,BUFF
38 01540 000000 POSX2:  0
39
39 01541 024777      LDA 1,,=1
40 01542 020130      LDA 0,C377
41 01543 107400      AND 0,1
42 01544 107000      ADD 3,1
43 01545 140414      SUB# 2,1,SZR
44 01546 063077      HALT
45 01547 020132      LDA 0,FIN200
46 01550 175400      INC 3,3
47 01551 116414      SUB# 0,3,SZR
48 01552 000766      JMP POSX2
49 01553 000755      JMP POSX1
50
51 01554 031400 CLDAPX: LDA 2,0,3    ;CONSTANT FIRST DISPLACEMENT
52 01555 031577 POSFIN: LDA 2,177,3 ;CONSTANT FINAL "
53 01556 000401 POSND:  JMP ,+1     ;REPLACE WITH JMP TO LOOP
54

```

```

A 0024 ,MAIN
01
02      ;A TEST OF NEGATIVE DISPLACEMENT USING "LDA"
03      ;THE BUFFER IS FILLED WITH A ADDRESS PATTERN
04      ;C(ADDRESS)=ADDRESS), A "LDA" INSTRUCTION
05      ;WITH A NEGATIVE DISPLACEMENT THEN REFERANCES
06      ;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)=CORECT EFFECTIVE ADDRESS,
10      ;C(2)=INDEXING VALUE,C(NEGX2)=FAILING INSTRUCTION
11      ;DISPLACEMENT IN BITS 8-15,
12
13 01557 034130 NEGX:  LDA 3,BUFF      ;FIRST BUFFER LOCATION
14 01560 024131      LDA 1,FIN        ;FINAL BUFFER LOCATION
15 01561 055400      STA 3,0,3      ;FILL THE BUFFER WITH A
16 01562 175400      INC 3,3        ;ADDRESS PATTERN,FOR
17 01563 136414      SUB# 1,3,SZR    ;EXAMPLE: C(3417)=3417
18 01564 000775      JMP NEGX+2     ;C(4150)=4150
19 01565 030431      LDA 2,NEGFIN   ;INITIALIZE THE LDA INST
20 01566 050412      STA 2,NEGX2    ;TO (LDA 3,377,2)
21
22 01567 020134 NEGX1: LDA 0,C3      ;SUBTRACT (3) FROM THE LDA'S
23 01570 024410      LDA 1,NEGX2    ;DISPLACEMENT
24 01571 100400      SUB 0,1
25 01572 044406      STA 1,NEGX2
26 01573 020424      LDA 0,NEGXX
27 01574 100033      ADCZ# 0,1,SNC
28 01575 000423      JMP NEGND
29 01576 000401      JMP NEGX2=1
30
31 01577 030133      LDA 2,BUF200
32 01600 000000 NEGX2:  0
33 01601 024777      LDA 1,,=1
34 01602 020136      LDA 0,C377
35 01603 107400      AND 0,1
36 01604 100000      COM 0,0
37 01605 107000      ADD 0,1
38 01606 147000      ADD 2,1
39 01607 136414      SUB# 1,3,SZR
40 01610 063077      HALT
41 01611 020131      LDA 0,FIN
42 01612 151400      INC 2,2
43 01613 112414      SUB# 0,2,SZR
44 01614 000764      JMP NEGX2
45 01615 000752      JMP NEGX1
46
47
48 01616 035377 NEGFIN: LDA 3,-1,2
49 01617 035200 NEGXX:  LDA 3,-200,2
50 01620 000401 NEGND:  JMP ,+1
51

```


A 0025 .MAIN

```

01
02      ;A TEST OF POSITIVE DISPLACEMENT USING THE "JMP".
03      ;THE BUFFER IS FILLED WITH (JSR 0,3) INSTRUCTIONS
04      ;THUS 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      ;C(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(POSJ3)=FAILING INST-
12      ;RUCTION,DISPLACEMENT IN BITS 8-15.
13
14 01621 034130 POSJ1: LDA 3,BUFF      ;FIRST LOCATION IN BUFFER
15 01622 024131      LDA 1,FIN        ;FINAL LOCATION IN BUFFER
16 01623 030435      LDA 2,POSJX     ;FILL THE BUFFER WITH(JSR 0,3)
17 01624 051400      STA 2,0,3       ;INSTRUCTIONS
18 01625 175400      INC 3,3
19 01626 166414      SUB# 3,1,SZR     ;TEST FOR END OF BUFFER
20 01627 000774      JMP POSJ+2       ;MORE LOCATIONS TO FILL
21 01630 020427      LDA 0,POSJZ     ;INITIALIZE THE JUMP
22 01631 040416      STA 0,POSJZ     ;WITH POSITIVE DISPLACEMENT
23
24 01632 020135 POSJ1: LDA 0,C5        ;INCREASE POSITIVE DISPLACE
25 01633 024414      LDA 1,POSJ3     ;BY FIVE
26 01634 107000      ADD 0,1
27 01635 044412      STA 1,POSJ3     ;NEW "JMP" INSTRUCTION
28 01636 020423      LDA 0,POSJF     ;ANY MORE + DISPLACEMENTS
29 01637 106032      ADC# 0,1,SZC    ;TO GO
30 01640 000423      JMP POSJND      ;NO GO TO NEXT TEST
31
32 01641 030130 POSJ2: LDA 2,BUFF     ;FIRST LOCATION IN BUFFER
33 01642 020132      LDA 0,FIN200    ;FINAL LOCATION IN BUFFER
34 01643 151400      INC 2,2
35 01644 142415      SUB# 2,0,SNR    ;POINT TO NEXT BUFFER
36 01645 000705      JMP POSJ1       ;LOCATION, IS IT THE LAST?
37 01646 034414      LDA 3,POSJY     ;YES TRY NEW DISPLACEMENT
38 01647 000000 POSJ3: 0           ;SETUP C(3) FOR BUFFER RET
39 01650 024777      LDA 1,,-1       ;"THE" JMP INSTRUCTION
40 01651 020136      LDA 0,C377      ;CALCULATE THE EFFECTIVE
41 01652 107400      AND 0,1         ;ADDRESS OF THIS JUMP
42 01653 147000      ADD 2,1         ;C(INDEX)+DISPLACEMENT
43 01654 130014      ADC# 1,3,SZR   ;C(1)=EFFECTIVE ADDRESS,C(3)
44 01655 003077      HALT          ;JSR PC,C(2)=INDEX VALUE
45 01656 000764      JMP POSJ2+1     ;GO AGAIN AT NEXT LOC
46
47 01657 001000 POSJZ: JMP 0,2       ;CONSTANTS
48 01660 005400 POSJX: JSR 0,3       ;"
49 01661 001177 POSJF: JMP 177,2
50 01662 001650 POSJY: POSJ3+1
51 01663 000401 POSJND: JMP ,+1     ;REPLACE WITH JMP TO LOOP
52

```

A 0026 .MAIN

```

01
02      ;A TEST OF NEGATIVE DISPLACEMENT USING "JSR"
03      ;THE BUFFER IS FILLED WITH (JSR 0,3) INSTRUCTIONS.
04      ;THE BUFFER IS ENTERED WITH A JSR VIA C(3) WITH A
05      ;NEGATIVE DISPLACEMENT IN BITS 8-15, EFFECTIVE
06      ;ADDRESS CALCULATION SHOULD SUBTRACT THE DIS-
07      ;PLACEMENT FROM THE VALUE IN INDEX REGISTER 3,
08      ;THE JSR IN THE BUFFER WILL RETURN TO THE PROGRAM
09      ;SAVING ITS LOCATION +1 IN C(3), COMPARISON OF
10      ;CALCULATED EFFECTIVE ADDRESS AND C(3) DETERMIN
11      ;IF THE BUFFER WAS ENTERED CORRECTLY.
12
13
14
15 01664 030432 NEGJ1: LDA 2,NEGJY     ;C(2)=CONSTANT (JSR -1,2)
16 01665 050415      STA 2,NEGJ3     ;STORE "THE" JSR
17
18 01666 020134 NEGJ1: LDA 0,C3       ;DECREMENT THE VALUE OF
19 01667 024413      LDA 1,NEGJ3     ;THE DISPLACEMENT BY 3.
20 01670 106400      SUB 0,1         ;EACH DISPLACEMENT IS TEST-
21 01671 044411      STA 1,NEGJ3     ;ED WITH THE BUFFER
22 01672 020423      LDA 0,NEGJX     ;DISPLACEMENT END TEST
23 01673 106032      ADC# 0,1,SZC    ;GO TO NEXT TEST
24 01674 000423      JMP NEGJD
25
26 01675 030133 NEGJ2: LDA 2,BUF200   ;FIRST ADDRESS+200 IN BUFFER
27 01676 020131      LDA 0,FIN        ;THIS ROUTINE SETS CARRY!!!
28 01677 151440      INCD 2,2       ;ITS TESTED LATER! INC BUFF
29 01700 142435      SUBZ# 2,0,SNR   ;ADDRESS AND TEST FOR END
30 01701 000765      JMP NEGJ1       ;END,GET NEW "JSR"
31
32 01702 000000 NEGJ3: 0           ;GO TO THE BUFFER HERE!
33 01703 024777      LDA 1,,-1       ;OBTAIN THE EFFECTIVE
34 01704 020136      LDA 0,C377      ;ADDRESS IN C(1)
35 01705 107403      AND 0,1,SNR    ;THE CARRY SHOULD NOT HAVE
36 01706 003077      HALT          ;CHANGED! SEE ABOVE
37 01707 100000      COM 0,0
38 01710 107000      ADD 0,1
39 01711 147000      ADD 2,1
40 01712 136014      ADC# 1,3,SZR   ;CHECK EFFECTIVE ADDRESS
41 01713 003077      HALT          ;CHECK EFFECTIVE ADDRESS
42 01714 000702      JMP NEGJ2+1     ;WITH PC OF BUFFER JSR IN
43                                     ;C(3),C(2)=INDEX VALUE
44 01715 005200 NEGJX: JSR +200,2   ;CONSTANT
45 01716 005377 NEGJY: JSR +1,2     ;CONSTANT
46 01717 000401 NEGJD: JMP ,+1     ;REPLACE WITH JMP TO LOOP
47
48

```

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

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

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

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

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

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

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

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

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

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

```

A 0029 ,MAIN
01
02
03
04
05
06
07 02001 102520 ISZ4: SUBZL 0,0      IC(0)=1, LOOP ON ERROR SWITCH
08 02002 126620      SUBZR 1,1      IC(1)=100000, ITERATION COUNT
09 02003 152000      ADC 2,2        IC(2)=177777
10 02004 050122 ISZ5: STA 2,TEM      ISTORE THE (-1)
11 02005 060377      NIOP CPU      ISYNC AT 7-A74
12 02006 010122      ISZ TEM       IINC IT SHOULD SKIP
13 02007 000404      JMP ISZ6      IFAIL TO SKIP, ISZ FAIL
14 02010 107004      ADD 0,1, SZR  IITERATION COUNTER
15 02011 000773      JMP ISZ5      ICONTINUE TEST
16 02012 000405      JMP ISZ7      IEND OF THIS TEST
17 02013 020122 ISZ6: LDA 0,TEM      IC(0)=RESULT OF FAIL ISZ
18 02014 063077      HALT         ICORRECT VALUE=0
19 02015 102400      SUB 0,0      ICLEAR C(0) TO LOOP
20 02016 000766      JMP ISZ5      ION ERROR FOREVER
21 02017 101234 ISZ7: MOVZRN 0,0, SZR IREPLACE TO LOOP TEST
22 02020 063077      HALT         IC(0) SHOULD BE +1
23 02021 000407      JMP ISZ10
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58

```

```

A 0030 ,MAIN
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58

```

```

IA TEST OF ISZ
IF FIVE CONSECUTIVE ISZ INSTRUCTIONS TO LOCATION "TEM" ARE
EXECUTED, THE VALUE IN C(TEM) SHOULD EQUAL C(3)
THE NUMBERS ARE ARRANGED SUCH THAT NO ISZ INSTRUCTION
SHOULD SKIP. SHOULD THE TEST FAIL PRESSING CONTINUE
WILL PLACE THE PROGRAM IN A FAILING LOOP UNTILL
RESTARTED.

```

```

ISZ20: LDA 2,C5      IC(2)=ERROR SWITCH
        SUB 0,0      ISTART WITH C(0)=0
        LDA 3,C5      IC(3)=EXPECTED RESULT OF
        ADDZ 0,3, SZC ISZ, ALSO TEST FOR END
        JMP ISZ22     IGO TO NEXT TEST
ISZ21: STA 0,TEM      ISTORE THE VALUE TO BE INC
        NIOP CPU      ISYNC AT 7-A74
        ISZ TEM       IDO IT 5 TIMES
        ISZ TEM
        ISZ TEM
        ISZ TEM
        ISZ TEM
        LDA 1,TEM     IC(1)=RESULT OF ISZ=5
        SUB# 1,3, SZR IC(3)=CORRECT RESULT
        HALT         ISZ FAILED
        SUB# 1,3, SZR IIF A ERROR OCCURED
        SUB 2,2      ICLEAR C(0) TO LOOP ERR
        ADD 2,0      IUPDATE FOR NEXT GO
        JMP ISZ20+2   IROUND
        JMP ,+1      ICHANGE TO LOOP TEST

```

```

EACH WORD IN THE BUFFER IS SET TO -1 (177777),
ISZ THEN INCREMENTS IT TO ZERO AND SKIPS, THE PROGRAM
WILL LOOP ON ERROR IF CONTINUE IS PRESSED

```

```

ISZ30: SUBZL 0,0      IC(0)=1, THE LOOP ON ERR SWIT
        LDA 1,FIN     IC(1)=FINAL BUFFER ADDRESS
        LDA 2,BUFF    IC(2)=FIRST BUFFER ADDRESS
        ADC 3,3       IC(3)=177777 (-1)
ISZ31: STA 3,0,2     ISTORE THE (-1) IN BUFFER
        NIOP CPU      ISYNC AT 7-A74
        ISZ 0,2       IINCREMENTION SHOULD SKIP
        JMP ISZ32+2   ISZ FAIL TO SKIP
        LDA 3,0,2     IDOES BUFFER=0?
        MOV# 3,3, SZR IIF NOT ISZ FAIL TO COUNT
        JMP ISZ32     ICORRECT , BUT SKIPPED
        ADD 0,2       IINC TO NEXT BUFFER LOC
        SUB# 1,2, SZR IAND TEST FOR END
        JMP ISZ31-1
        JMP ISZ33
ISZ32: HALT         IEND GO NEXT TEST
        JMP ,+3       ISZ SKIPPED BUT STORED NG
        LDA 3,0,2     IC(3)=ISZ RESULT, C(2)=OK
        HALT         IC(2)=ADDRESS OF ERR
        SUB 0,0       ISZ FAIL TO SKIP, C(3)=
        JMP ISZ31-1   ISZ RESULT, C(2)=EFF ADDRESS
        JMP ,+1       ILOOP ON THE ERROR
        JMP ,+1       ICHANGE TO LOOP TEST

```

```

A 0031 .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      ;REFERANCE 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 02135 126400 ISZ40: SUB 1,1      ;LOOP ON ERR SWITCH
19 02136 102000      ADC 0,0
20 02137 034131      LDA 3,FIN      ;FINAL BUFFER ADDRESS
21 02140 030130      LDA 2,BUFF     ;FIRST BUFFER ADDRESS
22 02141 041000      STA 0,0,2     ;FILL THE BUFFER WITH
23 02142 151400      INC 2,2      ;(-1)
24 02143 156414      SUB# 2,3,SZR   ;EXCEPT THE LAST
25 02144 000775      JMP ,-3      ;WHICH IS FILLED WITH
26 02145 045400      STA 1,0,3     ;ZEROS
27
28 02146 030130 ISZ41: LDA 2,BUFF   ;C(2) POINTS TO BUFFER
29 02147 060377      NIOP CPU      ;SYNC AT 7-A74
30 02150 011000      ISZ 0,2      ;ISZ SHOULD SKIP EVERY
31 02151 000403      JMP ISZ42     ;TIME EXCEPT LAST
32 02152 151400      INC 2,2      ;POINT NEXT LOCATION
33 02153 000774      JMP ISZ41+1
34
35 02154 021000 ISZ42: LDA 0,0,2     ;C(0)=RESULT OF ISZ
36 02155 156414      SUB# 2,3,SZR   ;SHOULD BE A 0
37 02156 126001      ADC 1,1,SKP   ;C(1)=LOOP ON ERR SWIT
38 02157 156414      SUB# 2,3,SZR   ;C(2)=EFF ADDRESS OF ISZ
39 02160 063077      HALT        ;OPERAND.
40 02161 125004      MOV 1,1,SZR   ;IF C(1) NOT 0 PROG
41 02162 000754      JMP ISZ40+1   ;WILL LOOP
42

```

```

A 0032 .MAIN
01
02      ;A TEST OF DSZ ABILITY TO NOT SKIP
03      ;THE C(TEM) IS SET TO (-1). DECREMENTING SHOULD
04      ;PRODUCE +2 AND NOT SKIP. PRESSING CONTINUE AFTER A
05      ;ERROR WILL CAUSE THE PROGRAM TO LOOP ON ERROR
06      ;UNTILL RESTARTED.
07
08
09 02163 102520 DSZ0: SUBZL 0,0     ;C(0)=1, LOOP ON ERR SWITCH
10 02164 024136      LDA 1,C377
11 02165 152000      ADC 2,2     ;C(2) NOT USED THIS TEST
12 02166 176000      ADC 3,3     ;C(3)=+1
13
14 02167 054122 DSZ1: STA 3,TEM     ;STORE THE VALUE
15 02170 060377      NIOP CPU      ;SYNC AT 7-A74
16 02171 014122      DSZ TEM      ;DECREMENT C(TEM)
17 02172 000404      JMP ,+4      ;IT SHOULD NOT SKIP
18 02173 020122      LDA 0,TEM     ;IT SKIPPED, C(0)=DSZ RESULT
19 02174 063077      HALT        ;CORRECT VALUE =+2
20 02175 102401      SUB 0,0,SKP   ;CLEAR C(0) TO LOOP ERR
21 02176 106404      SUB 0,1,SZR   ;CHANGE AND TEST ITERATION
22 02177 000770      JMP DSZ1     ;COUNTER, ANOTHER PASS
23 02200 150014      COM# 2,2,SZR  ;C(2)MODIFIED 1
24 02201 063077      HALT
25
26      ;A TEST OF DSZ ABILITY TO SKIP
27      ;C(TEM) IS SET TO 0. DECREMENTING C(TEM) TO 0
28      ;SHOULD PRODUCE A DSZ SKIP, PRESSING CONTINUE
29      ;AFTER A ERROR WILL CAUSE THE PROGRAM TO LOOP
30      ;UNTILL RESTARTED.
31
32 02202 102520 DSZ2: SUBZL 0,0     ;C(0)=1, LOOP ON ERR SWITCH
33 02203 126620      SUBZR 1,1     ;C(1)=100000, ITERATIONS
34 02204 152520      SUBZL 2,2     ;C(2)=1
35 02205 050122 DSZ3: STA 2,TEM     ;STORE THE 0
36 02206 060377      NIOP CPU      ;SYNC AT 7-A74
37 02207 014122      DSZ TEM      ;DECREMENT, IT SHOULD SKIP
38 02210 000404      JMP DSZ4     ;DSZ FAIL TO SKIP
39 02211 107004      ADD 0,1,SZR   ;TEST ITERATION COUNT
40 02212 000773      JMP DSZ3     ;CONTINUE ON
41 02213 000405      JMP DSZ5     ;NEXT TEST
42 02214 020122 DSZ4: LDA 0,TEM     ;C(0)=RESULT OF FAIL DSZ
43 02215 063077      HALT        ;CORRECT COUNT=0
44 02216 102400      SUB 0,0     ;CLEAR C(0) TO LOOP
45 02217 000766      JMP DSZ3
46 02220 101234 DSZ5: MOVZR# 0,0,SZR ;C(0) SHOULD BE +1
47 02221 063077      HALT        ;CHANGE MOVZR TO LOOP TEST
48

```

```

A 0033 ,MAIN
01
02
03 ;A TEST OF DSZ
04 ;THE RESULT OF A DSZ INSTRUCTION IS CHECKED
05 ;WITH THE RESULT OF A ADD INSTRUCTION. THE ADD
06 ;IS PRESUMED TO WORK BECAUSE OF PREVIOUS TESTING.
07 ;THE DECREMENT LOGIC OF THE DSZ, ITS ABILITY
08 ;TO SKIP, AND NOT SKIP IS CHECKED. SHOULD THE
09 ;DSZ FAIL, THE PROGRAM WILL HALT. PRESSING
10 ;CONTINUE WILL PLACE THE PROGRAM IN A FAILING LOOP
11 ;FOR SCOPING PURPOSES (YOU MUST REMOVE THE HALT). THE
12 ;PROGRAM
13 ;MUST BE RESTARTED TO GET OUT OF THE FAILING LOOP
14
15 02222 152000 DSZ10: ADC 2,2 ;C(2)=+1
16 02223 176000 ADC 3,3 ;C(3)=+1
17 02224 102620 SUBZR 0,0
18 02225 101400 INC 0,P ;C(0)=100001
19 02226 143025 ADDZ 2,0,SNR ;TEST FOR PASSING THROUGH 0
20 02227 000424 JMP DSZ13 ;DSZ FAILED TO SKIP
21 02230 040122 STA 0,TEM ;STORE THE VALUE FOR DSZ
22 02231 060377 NIOP CPU ;SYNC AT 7-A74
23 02232 014122 DSZ TEM ;T. DECREMENT. THE DSZ SKIPS
24 02233 176001 ADC 3,3,SKP ;ON EXIT, SET C(3)=+1
25 02234 000411 JMP DSZ12 ;DSZ SKIPPED.
26 02235 024122 LDA 1,TEM ;COUNT OBTAINED FROM DSZ
27 02236 122015 ADC# 1,0,SNR ;IS COUNT CORRECT?
28 02237 000767 JMP DSZ10+4 ;YES CONTINUE ON
29
30 02240 117000 DSZ11: ADD 0,3 ;C(3)=CORRECT COUNT
31 02241 063077 HALT ;C(1)=DSZ COUNT
32 02242 152400 SUB 2,2 ;CLEARING AC2 WILL CAUSE
33 02243 176000 ADC 3,3
34 02244 000762 JMP DSZ10+4 ;PROGRAM TO LOOP ON ERR
35
36 02245 101235 DSZ12: MOVZR# 0,0,SNR ;DSZ SKIPPED, WAS THAT OK?
37 02246 000412 JMP DSZ14 ;YES GO TO NEXT TEST
38 02247 024122 LDA 1,TEM ;C(1)=DSZ COUNT CAUSING
39 02250 117000 ADD 0,3 ;THE SKIP, C(3)=CORRECT
40 02251 063077 HALT ;DSZ SKIP LOGIC FAIL
41 02252 000770 JMP DSZ11+2 ;LOOP ON THE ERROR
42
43 02253 024122 DSZ13: LDA 1,TEM ;DSZ FAILED TO SKIP
44 02254 176000 SUB 3,3
45 02255 102400 SUB 0,0 ;FROM 0,C(1)=DSZ COUNT
46 02256 063077 HALT ;C(3)=CORRECT COUNT
47 02257 000763 JMP DSZ11+2 ;LOOP ON ERR
48
49 02260 000401 DSZ14: JMP ,+1 ;CHANGE TO LOOP TEST
50

```

```

A 0034 ,MAIN
01
02 ;A TEST OF DSZ
03 ;FIVE CONSECUTIVE DSZ INSTRUCTIONS TO LOCATION TEM
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 APANGED SUCH THAT NO DSK INSTRUCTION
08 ;SHOULD SKIP.
09
10 02261 030135 DSZ20: LDA 2,C5 ;C(2)=ERROR LOOP SWITCH
11 02262 102620 SUBZR 0,0 ;START AT 100000
12 02263 034123 LDA 3,CM5 ;C(3)=EXPECTED DSZ RESULT
13 02264 117000 ADD 0,3
14 02265 040122 DSZ21: STA 0,TEM ;STORE THE VALUE FOR DEC
15 02266 060377 NIOP CPU ;SYNC AT 7-A74
16 02267 014122 DSZ TEM ;DECREMENT C(TEM)
17 02270 014122 DSZ TEM
18 02271 014122 DSZ TEM
19 02272 014122 DSZ TEM
20 02273 014122 DSZ TEM
21 02274 024122 LDA 1,TEM ;C(1)=RESULT OF DECREMENT
22 02275 136414 SUB# 1,3,SZR ;C(3)=CORRECT COUNT
23 02276 063077 HALT ;DSZ FAILED
24 02277 136414 SUB# 1,3,SZR
25 02300 152400 SUB 2,2 ;CLEAT C(2) TO LOOP
26 02301 143023 ADDZ 2,0,SNR ;UPDATE FOR NEXT
27 02302 000761 JMP DSZ20+2 ;GO ROUND
28 02303 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 CONTINUUSLY
34 ;UNTILL IT IS RESTARTED.
35
36 02304 102520 DSZ30: SUBZL 0,0 ;C(0)=1, LOOP ON ERR SWITCH
37 02305 024131 LDA 1,FIN ;C(1)=FINAL BUFFER ADDRESS
38 02306 030130 LDA 2,BUFF ;C(2)=FIRST BUFFER ADDRESS
39 02307 176520 SUBZL 3,3 ;C(3)=+1
40 02310 055000 DSZ31: STA 3,0,2 ;STORE WORD OF 0 IN BUFFER
41 02311 060377 NIOP CPU ;SYNC AT 7-A74
42 02312 015000 DSZ 0,2 ;DECREMENTION SHOULD SKIP
43 02313 000412 JMP DSZ32+2 ;DSZ FAIL TO SKIP
44 02314 035000 LDA 3,0,2 ;DSZ SKIPPED OK, IS MEN
45 02315 175014 MOV# 3,3,SZR ;WORD 0?
46 02316 000405 JMP DSZ32 ;NO! ERROR
47 02317 113000 ADD 0,2 ;ADVANCE TO NEXT LOCATION
48 02320 132414 SUB# 1,2,SZR ;MORE LOCATIONS IN BUFFER
49 02321 000766 JMP DSZ31-1 ;GO TO NEXT TEST
50 02322 000407 JMP DSZ33
51 02323 063077 DSZ32: HALT ;DSZ SKIPPED BUT STORE NG
52 02324 000403 JMP ,+3 ;C(2)=EFF ADDRESS,C(3)=RESULT
53 02325 035000 LDA 3,0,2 ;OF THE DSZ
54 02326 063077 HALT ;DSZ FAIL TO SKIP
55 02327 102400 SUB 0,0 ;C(2)=EFFECTIVE ADDRESS
56 02330 000757 JMP DSZ31-1 ;LOOP THIS TEST
57 02331 000401 DSZ33: JMP ,+1 ;CHANGE TO LOOP TEST

```

```

A 0136 .MAIN
01
02
03
04
05
06
07
08
09 02375 102520
10 02376 030130
11 02377 034131
12 02400 050122
13 02401 060377
14 02402 026122
15 02403 024122
16 02404 132414
17 02405 063077
18 02406 132414
19 02407 102400
20 02410 113000
21 02411 156414
22 02412 000766
23 02413 000401

;A TEST OF INDIRECT ADDRESSING
;THE C(TEM) IS SET TO POINT TO A BUFFER, A LDA
;INSTRUCTION THEN REFERENCES THE BUFFER INDIRECTLY.
;THE C(TEM) IS CHECKED TO INSURE THAT IT HAS NOT
;CHANGED. PRESSING CONTINUE AFTER A ERROR WILL CAUSE
;THE PROGRAM TO LOOP ON THE FAILING CONDITIONS
;UNTILL RESTARTED.
DEFB: SUB# 1,2,SZR
LDA 2,BUFF
LDA 3,FIN
STA 2,TEM
NIO# CPU
;REFERENCE REGISTER TEM
LDA 1,TEM
LDA 1,TEM
SUB# 1,2,SZR
HALT
;IF C(1)=VALUE OF TEM
;IF C(2)=VALUE STORED IN TEM
;CLEAR C(0) TO LOOP
;INC BUFFER POINTER
;TEST FOR END OF BUFF
;CHANGE TO LOOP TEST

```

```

A 0135 .MAIN
01
02
03
04
05
06
07
08
09 02332 126000
10 02333 102520
11 02334 034131
12 02335 030130
13 02336 041000
14 02337 151400
15 02340 156414
16 02341 000775
17 02342 051400
18
19 02343 102000
20 02344 034124
21 02345 030130
22
23 02346 175000
24 02347 060377
25 02350 015374
26 02351 000410
27 02352 015375
28 02353 000407
29 02354 015376
30 02355 000406
31 02356 015377
32 02357 000405
33 02360 000766
34
35 02361 113000
36 02362 113000
37 02363 113000
38 02364 113000
39 02365 034131
40 02366 021000
41 02367 156414
42 02370 126400
43 02371 156414
44 02372 000777
45 02373 124014
46 02374 000737
47
48
49

;ANOTHER TEST OF DSZ
;THIS TEST WILL SUBTRACT ONE FROM EACH LOCATION OF A
;BUFFER SET TO '+'. THE TEST HAS BEEN WRITTEN TO GIVE
;A HIGH REP RATE FOR THE DSZ INSTRUCTION. PRESSING
;CONTINUE AFTER A ERROR WILL CAUSE THE TEST TO BE
;ITERATED UNTILL RESTARTED.
DSZ40: ADC 1,1
SUB# 1,2,SZR
;IC(1)=LOOP ON ERROR SWITCH
;IC(0)=+1
;SET ALL LOCATIONS IN
;THE BUFFER TO ZERO
;EXCEPT THE LAST LOCATION
;DSZ SKIPS ALL BUT LAST
;IC(0)=1, FOR ERROR CHECK
;IC(3)=+1,BUFFER INCREMENT
;IC(2) POINTS TO THE BUFFER
;INC BUFFER POINTER BY 4
;SYNC 7=A74
;DECREMENT THE BUFFER
;VIA DSZ. DSZ SHOULD SKIP
;UNLESS END OF BUFFER
;NOT YET BUFFERS END
;IC(2) WILL BE SET
;TO THE EFFECTIVE ADDRESS
;OF THE ENTERING DSZ
;INSTRUCTION
;VOID DSZ ENTER AT END OF
;BUFFER OR DID IT FAIL?
;IC(3)=END OF BUFFER
;IC(2)=EFFECTIVE ADDRESS
;IC(0)=RESULT OF DSZ INST
;A DSZ FAILED TO SKIP
;IF C(1)=0 LOOP ON ERROR
;IC(0)=+1

```

```

4 0038 .MAIN
21
22 IA TEST OF INDIRECT ADDRESSING
23 IAE DO NOT YET KNOW THAT INDIRECT ADDRESSING WORKS
24 THIS TEST INSURES THAT THE STACK STORES INDIRECT
25 AND NOT DIRECTLY. IF A ERR OCCURS PRESSING CONTINUE
26 WILL CAUSE THE PROGRAM TO LOOP ON ERROR UNTILL
27 RESTART.
28
29 DEF14: SUBZL 0,0
30 LDA 2,BUFF
31 LDA 3,C577
32 STA 2,TEM
33 COM 2,2
34 ISYNC AT 7-A74
35 ISTORE VIA C(TEM) INDIRECT
36 ISTORE C(2)
37 ISTORE VIA C(TEM) INDIRECT
38 ISTORE C(2)
39 COM 2,2
40 LDA 1,TEM
41 SUB# 1,2,SZR
42 HALT
43 SUB# 2,1,SZR
44 SUB 0,0
45 ADD 0,2
46 SUB 0,3,SZR
47 JMP DEF14+3
48
49 DEF15: JMP +1
50
51 IA TEST OF INDIRECT ADDRESSING
52 THIS ROUTINE WILL TEST THE ABILITY OF THE STA
53 INSTRUCTION
54 TO STORE IN A BUFFER USING INDIRECT ADDRESSING, THE
55 DATA
56 IS
57 STORED, EXAMPLE: C(9873)=172704. IF A ERROR OCCURS
58 IPRESS
59 ICONTINUE TO LOOP ON THE FAILING CONDITIONS.
60
61 DEF20: SUBZL 0,0
62 LDA 2,BUFF
63 STA 2,TEM
64 COM 2,3
65 NIOP CPU
66 STA 3,STEM
67 LDA 1,0,2
68 SUB# 1,3,SZR
69 HALT
70 SUB# 1,3,SZR
71 SUB 0,0
72 ADD 0,2
73 LDA 3,FIN
74 SUB# 2,3,SZR
75 JMP DEF20+2
76
77 DEF21: JMP +1
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

4 0037 .MAIN
21
22 IA TEST OF INDIRECT ADDRESSING
23 IA ADDRESS PATTERN (C(ADDRESS)=ADDRESS) IS STORED IN
24 THE BUFFER VIA INDEX REGISTER 3. A LDA INSTRUCTION
25 THEN
26 PREFERENCES EACH LOCATION IN THE BUFFER INDIRECTLY,
27 BECAUSE THE BUFFER CONTAINS A ADDRESS PATTERN THE DATA
28 OBTAINED REPRESENTS THE ADDRESS OBTAINED. PRESSING
29 ICONTINUE AFTER A PRORR CAUSES THE PROGRAM TO LOOP ON
30 THE
31 ERROR UNTILL RESTARTED.
32
33 DEF10: SUBZL 0,0
34 LDA 3,BUFF
35 LDA 2,FIN
36 STA 3,0,3
37 INC 3,3
38 SUB# 2,3,SZR
39 JMP +3
40
41 DEF11: LDA 3,BUFF
42 STA 3,STEM
43 NIOP CPU
44 LDA 1,STEM
45 SUB# 1,3,SZR
46 HALT
47 SUB# 1,3,SZR
48 SUB 0,0
49 ADD 0,3
50 SUB# 2,3,SZR
51 JMP DEF11
52
53 IA TEST OF INDIRECT ADDRESSING
54 IPRESET C(TEM) FOR INDIRECT
55 ISYNC AT 7-A74
56 IGO INDICTEC TO GET DATA
57 IC(1)=DATA/ADDRESS OBTAINED
58 IC(3)=CORRECT DATA/ADDRESS
59 IIF IT FAILS
60 ICLR 0 TO LOOP ON ERROR
61 IADVANCE TO NEXT ADDRESS
62 ITEST FOR END OF BUFFER
63 ICHANGE TO LOOP TEST
64
65 DEF10: SUBZL 0,0
66 LDA 3,BUFF
67 LDA 2,FIN
68 STA 3,0,3
69 INC 3,3
70 SUB# 2,3,SZR
71 JMP +3
72
73 DEF11: LDA 3,BUFF
74 STA 3,STEM
75 NIOP CPU
76 LDA 1,STEM
77 SUB# 1,3,SZR
78 HALT
79 SUB# 1,3,SZR
80 SUB 0,0
81 ADD 0,3
82 SUB# 2,3,SZR
83 JMP DEF11
84
85 IA TEST OF INDIRECT ADDRESSING
86 IPRESET C(TEM) FOR INDIRECT
87 ISYNC AT 7-A74
88 ISTORE VIA C(TEM) INDIRECT
89 ISTORE C(2)
90 ISTORE VIA C(TEM) INDIRECT
91 ISTORE C(2)
92 COM 2,2
93 LDA 1,TEM
94 SUB# 1,2,SZR
95 HALT
96 SUB# 2,1,SZR
97 SUB 0,0
98 ADD 0,2
99 SUB 0,3,SZR
100 JMP DEF14+3

```

```

A 0039 ,MAIN
01
02
03      ;A TEST OF INDIRECT ADDRESSING
04      ;THE BUFFER IS SET TO CONTAIN A ADDRESS PATTERN, A LDA
05      ;INSTRUCTION VIA C(2) THEN CAUSES A INDIRECT REFERENCE
06      ;TO BE MADE AT EACH LOCATION IN THE BUFFER, BECAUSE
07      ;EACH LOCATION CONTAINS ITS ADDRESS THE LDA INSTRUCTION
08      ;WILL LOAD THE LOCATION INDIRECTED THROUGH, PRESSING
09      ;CONTINUE AFTER A ERR WILL CAUSE THE PROGRAM TO LOOP ON
10      ;ERROR UNTILL RESTARTED.
11 02500 102520 DEF30: SUBZL 0,0      ;C(0)=1, LOOP ON ERROR SWITCH
12 02501 030130      LDA 2,BUFF      ;FIRST ADDRESS IN BUFFER
13 02502 034131      LDA 3,FIN      ;FINAL ADDRESS IN BUFFER
14 02503 051000      STA 2,0,2      ;STORE A ADDRESS PATTERN
15 02504 151400      INC 2,2        ;IN THE BUFFER
16 02505 156414      SUB# 2,3,SZR    ;EXAMPLE: C(7314)=7314
17 02506 000775      JMP ,+3
18
19 02507 030130 DEF31: LDA 2,BUFF      ;FIRST BUFFER LOCATION
20 02510 060377      NIOP CPU      ;SYNC AT 7-A74
21 02511 027000      LDA 1,0,2      ;SHOULD LOAD WD POINTED TO
22 02512 132414      SUB# 1,2,SZR    ;C(2)=ADDRESS INDIRECT THROUGH
23 02513 063077      HALT          ;C(1)=VALUE OBTAINED
24 02514 132414      SUB# 1,2,SZR
25 02515 102400      SUB 0,0        ;CLEAR 0 TO LOOP ON ERROR
26 02516 113000      ADD 0,2        ;INC TO NEXT BUFFER LOC
27 02517 156414      SUB# 2,3,SZR    ;TEST FOR END OF BUFFER
28 02520 000770      JMP DEF31+1
29 02521 000401 DEF32: JMP ,+1      ;CHANGE TO LOOP TEST
30
31      ;A TEST OF INDIRECT ADDRESSING
32      ;A MODIFIED ADDRESS PATTERN (C(ADDRESS)=ADDRESS-1) IS
33      ;STORED IN THE BUFFER, A STA INSTRUCTION INDIRECTS
34      ;THROUGH EACH LOCATION IN THE BUFFER STORING IN THAT
35      ;LOCATION-1, PRESSING CONTINUE AFTER A ERROR CAUSES THE
36      ;PROGRAM TO LOOP ON THE ERROR UNTILL RESTARTED.
37
38 02522 102520 DEF34: SUBZL 0,0      ;C(0)=1
39 02523 034131      LDA 3,FIN      ;FINAL LOCATION IN BUFFER
40 02524 030130      LDA 2,BUFF      ;FIRST LOCATION IN BUFFER
41 02525 051001      STA 2,1,2      ;IN EACH BUFFER LOCATION
42 02526 151400      INC 2,2        ;STORE THE ADDRESS OF THAT
43 02527 156414      SUB# 2,3,SZR    ;LOCATION-1
44 02530 000775      JMP ,+3
45 02531 030130 DEF35: LDA 2,BUFF      ;FIRST BUFFER LOCATION
46 02532 060377      NIOP CPU      ;SYNC AT 7-A74
47 02533 053001      STA 2,0,1,2    ;DEFER THROUGH BUFFER, STORE
48 02534 025000      LDA 1,0,2      ;C(2) IN BUFFER-1, THEREFOR
49 02535 132414      SUB# 1,2,SZR    ;STORE ADDRESS IN ADDRESS.
50 02536 063077      HALT          ;FOR EXAMPLE: C(4050)=XXX
51 02537 132414      SUB# 1,2,SZR    ;C(4051)=4050, C(2)=4050
52 02540 102400      SUB 0,0        ;INDIRECT THROUGH 4051 TO
53 02541 113000      ADD 0,2        ;STORE C(2) IN LOCATION 4050
54 02542 156014      ADC# 2,3,SZR    ;TEST FOR END OF BUFFER
55 02543 000767      JMP DEF35+1
56 02544 000401 DEF36: JMP ,+1      ;CHANGE TO LOOP TEST
57

```

```

A 0040 ,MAIN
01
02
03      ;A TEST OF INDIRECT ADDRESSING (2 LEVELS)
04      ;THE BUFFER IS SET TO CONTAIN A ADDRESS PATTERN, A LDA
05      ;INSTRUCTION USING TWO LEVELS OF INDIRECT ADDRESSING
06      ;THEN REFFERENCES EACH LOCATION IN THE BUFFER, PRESSING
07      ;CONTINUE AFTER A ERROR WILL CAUSE THE PROGRAM TO LOOP
08      ;ON THE ERROR UNTILL RESTARTED.
09 02545 102520 DEF40: SUBZL 0,0      ;C(0)=1, LOOP ON ERR SWITCH
10 02546 030130      LDA 2,BUFF      ;FIRST BUFFER LOCATION
11 02547 034131      LDA 3,FIN      ;FINAL BUFFER LOCATION
12 02550 051000      STA 2,0,2      ;STORE A ADDRESS PATTERN
13 02551 151400      INC 2,2        ;IN THE BUFFER
14 02552 156414      SUB# 2,3,SZR
15 02553 000775      JMP ,+3
16
17 02554 030130 DEF41: LDA 2,BUFF      ;FIRST BUFFER LOCATION
18 02555 126620      SUBZR 1,1      ;C(1)=100000, THE DEFER BIT
19 02556 147000      ADD 2,1        ;C(2)=INDIRECT POINTER
20 02557 044122      STA 1,TEM      ;TO BUFFER
21 02560 060377      NIOP CPU      ;SYNC AT 7-A74
22 02561 026122      LDA 1,0,TEM    ;GO INDIRECT FOR DATA
23 02562 132414      SUB# 1,2,SZR    ;C(1)=DATA FROM BUFFER
24 02563 063077      HALT          ;C(2)=CORRECT DATA
25 02564 132414      SUB# 1,2,SZR    ;2 LEVELS INDIRECT FAIL
26 02565 102400      SUB 0,0        ;CLEAR TO LOOP ON ERROR
27 02566 113000      ADD 0,2        ;INC TO NEXT BUFF LOC
28 02567 156414      SUB# 2,3,SZR    ;TEST FOR END OF BUFFER
29 02570 000765      JMP DEF41+1
30 02571 000401 DEF42: JMP ,+1      ;CHANGE TO LOOP TEST
31
32
33
34
35      ;A TEST OF JSR (HARD TEST)
36      ;THE BUFFER IS FILLED WITH THE FOLLOWING
37      ;      JSR 2,3      ;GO TO THE LOCATION SPECIFIED BYC(3)+2
38      ;      JSR 0,2      ;GO BACK TO THE CONTROL PROGRAM
39      ;      JSR 0,2      ;" " " " " " "
40      ;      JSR 0,2      ;" " " " " " "
41      ;      JSR 2,3      ;GO TO THE LOCATION SPECIFIED BY C(3)+2
42      ;THE C(3) IS INITIALLY SET TO BUFFER+1
43      ;THE C(2) " " " " RETURN TO MAIN PROGRAM
44      ;THE PROGRAM THEN JUMPS TO THE FIRST BUFFER LOCATION,
45      ;THE BUFFER SEQUENCE IS AS FOLLOWS:
46      ;#1 JSR TO BUFFER+3 C(3)=BUFFER+1
47      ;#2 JSR TO BUFFER+3 C(3)=BUFFER+4
48      ;#3 JSR TO BUFFER+6 C(3)=BUFFER+4
49      ;#4 JSR TO BUFFER+6 C(3)=BUFFER+7
50      ;#5 JSR TO BUFFER+11 C(3)=BUFFER+7
51      ;#6 ETC
52

```



```

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
;A TEST OF INDIRECT ADDRESSING (2 LEVELS)
;THE FIRST HALF OF THE BUFFER IS FILLED WITH (JSR 0,3)
;INSTRUCTIONS, THE SECOND HALF IS FILLED
;AND ADDRESS POINTING TO THE (JSR) INSTRUCTIONS IN THE
;BOTTOM. C(TEM) POINTS TO THE TOP HALF OF THE BUFFER.
;THE SEQUENCE IS AS FOLLOWS:
;#1 GO INDIRECT THROUGH REGISTER TEM
;#2 WORD IN TOP POINTS TO BOTTOM
;#3 INSTRUCTION IN BOTTOM IS EXECUTED
;#4 PC C(3) FROM BOTTOM IS CHECKED
;PRESSING CONTINUE AFTER A ERROR WILL CAUSE THE PROGRAM
;TO LOOP ON THE ERROR UNTILL NESTARTED.
DEF50: LDA 3,FIN ;FINAL BUFFER ADDRESS
LDA 2,BUFF ;FIRST BUFFER ADDRESS
LDA 1,BUFF
LDA 0,CJDF1
CONSTANT (JSR 0,3)
SUBOR 3,1
STA 0,0,2
INC 2,2
INC 1,1,1SZR
JMP 1,3
STA 2,TEM
LDA 0,BUFF
STA 0,0,2
INC 0,0
;STORE A MODIFIED ADDRESS
;PATTERN IN TOP HALF OF
;THE BUFFER. C(TOP)=
;BOTTOM ADDRESS+100000
DEF51: LDA 2,TEM
SUBR 0,0
ADD 2,0
STA 0,TEM
LDA 1,BUFF
SUBL 0,0
;SYNC AT 7=474
;GO TO THE BUFFER C(3)=
;JSR RETURN.C(1)=CORRECT
;FINAL ADDRESS,INDIRECT WITH
;JSR FAILED
;CLEAR TO LOOP ON ERROR
;POINT TO NEXT BUFFER LOC
;INC IF NOT ERROR
;TEST FOR END OF BUFFER
;CHANGE TO LOOP TEST
;A CONSTANT
;JSR 0,3

```

```

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
;THIS PROCESS CONTINUES UNTILL THE END OF THE BUFFER IS
;REACHED.
;AT FINAL ADDRESS,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
STA 0,0,2
;STORE CONTINUE
;" RETURN
;"
STA 1,1,2
STA 1,2,2
INC 2,2
INC 2,2
INC 2,2
SUBR 2,3,1SZR
;TEST FOR END OF BUFFER
;PERHAPS 1,2 EXTRA WORDS
JMP JSR21
;MAKE LAST CONTINUE A RETURN
STA 1,-3,2
;INITIAL SETUP OF C(3)
;FOR FIRST JSR IN BUFFER
;THE PROGRAM RETURN
;BOTH C(0),C(1) SET TO 0
;WILL CHECK FOR NO EFFECT BY
;JSR
;GO TO THE BUFFER!
JMP -1,3
;THREE CONSTANTS
JSR 0,2
JSR 2,3
+1
;C(0),C(1) SHOULD NOT HAVE
;BEEN CHANGED!!
HALT
LDA 0,FIN
SUBOR 3,0,1SZR
;FOR FINAL+1,ADDRESS
;C(3) = ADDRESS OF RETURNING
MOV 0,0,SKP
HALT
MOV 0,0
;CHANGE TO LOOP TEST

```

```

A 0043 .MAIN
01
02 02673 024130 RANFL: LDA 0,BUFF      ;FILL THE BUFFER WITH
03 02674 040122      STA 0,TEM      ;RANDOM ADDRESS IN
04 02675 034131      LDA 3,FIN      ;BITS 4-15. THIS VALUE
05 02676 024000      LDA 1,0        ;WILL POINT TO SOME
06 02677 121000 RANFC: MOV 1,0        ;OTHER SPOT IN THE BUFFER
07 02700 024433      LDA 1,C1347     ;BIT 0 IS RANDOMLY 0/1
08 02701 152620      SUBZR 2,2
09 02702 107222      ADDZR 0,1,SZC ;THIS
10 02703 147000      ADD 2,1
11 02704 131000      MOV 1,2
12 02705 113520      ANDZL 0,2
13 02706 107000      ADD 0,1
14 02707 146400      SUR 2,1
15
16 02710 020125      LDA 0,MSIZE     ;MASK OF MEM SIZE
17 02711 123400      AND 1,0
18 02712 030130      LDA 2,BUFF
19 02713 102433      SUBZ# 3,0,SNC
20 02714 142033      ADCZ# 2,0,SNC
21 02715 000762      JMP RANFC
22 02716 030122      LDA 2,TEM
23 02717 112415      SUB# 0,2,SNR
24 02720 000757      JMP RANFC
25 02721 101120      MOVZL 0,0
26 02722 125100      MOVL 1,1
27 02723 101200      MOVR 0,0
28 02724 125200      MOVR 1,1
29 02725 042122      STA 0,0TEM
30 02726 010122      ISZ TEM
31 02727 030122      LDA 2,TEM
32 02730 156014      ADC# 2,3,SZR
33 02731 000740      JNP RANFC
34 02732 101011      MOV# 0,0,SKP
35 02733 135753 C1347: 135753
36

```

```

A 0044 .MAIN
01
02
03 ;A 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 REFERANCES
11 ;INDIRECTLY
12 ;THE BUFFER. THE RESULT OC 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 02734 024130 DEF60: LDA 1,BUFF      ;FIRST LOCATION IN THE BUFFER
21 02735 121120      MOVZL 1,0
22 02736 101240      MOVOR 0,0
23 02737 040137      STA 0,CFOO
24 02740 102520      SUBZL 0,0      ;C(0)=1,LEVEL COUNTER
25 02741 131000      MOV 1,2
26 02742 101125      MOVZL 0,0,SNR ;COUNT INDIRECT LEVELS
27 02743 000417      JMP DEF62     ;15 LEVELS,GIVE UP THIS CHAIN
28 02744 031000      LDA 2,0,2
29 02745 151132      MOVZL# 2,2,SZC ;GET BUFFER WORD
30 02746 000774      JMP -4        ;DOES IT HAVE A INDIRECT BIT
31 02747 050122      STA 2,TEM    ;YES KEEP LOOKING
32 02750 031000      LDA 2,0,2
33 ;C(2)=FINAL VALUE
34 02751 102520 DEF61: SUBZL 0,0
35 02752 060377      NIOP CPU
36 02753 036137      LDA 3,CFOO
37 02754 172414      SUB# 3,2,SZR
38 02755 003077      HALT
39 02756 172414      SUB# 3,2,SZR
40 02757 102401      SUB 0,0,SKP
41 02760 101005      MOV 0,0,SNR
42 02761 000771      JMP DEF61+1
43 ;C(1)=PLACE BUFFER WAS ENTERED
44 02762 125400 DEF62: INC 1,1
45 02763 034131      LDA 3,FIN
46 02764 136414      SUB# 1,3,SZR
47 02765 000750      JMP DEF60+1
48 ;NOT YET BUFFER END
49 02766 000401 DEF63: JMP ,+1
50 ;CHANGE TO LOOP TEST

```

```

A 0045 .MAIN
01
02      ;A 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 02767 030130 EXCH:  LDA 2,BUFF      ;CAN'T USE AUTO INC/DEC YET
12 02770 034131      LDA 3,FIN       ;EXCHANGE BUFFER ENDS SO
13 02771 021000      LDA 0,0,2       ;IT WONT BE THE SAME AS
14 02772 025777      LDA 1,-1,3      ;LAST TEST
15 02773 045000      STA 1,0,2
16 02774 041777      STA 0,-1,3
17 02775 151400      INC 2,2
18 02776 174000      NEG 3,3         ;INC TOWARDS TOP,DEC TOWARDS
19 02777 174000      COM 3,3         ;BOTTEM
20 03000 172433      SUB# 3,2,SNR    ;TEST FOR MIDDLE
21 03001 000770      JMP EXCH+2     ;OF THE BUFFER
22
23 03002 024130 DEF64: LDA 1,BUFF     ;FIRST BUFFER LOCATION
24 03003 121120      MOVZL 1,0
25 03004 101240      MOVOR 0,0
26 03005 040137      STA 0,CF00
27 03006 102520      SUBZL 0,0     ;C(0)=LEVEL COUNTER
28 03007 131000      MOV 1,2
29 03010 101125      MOVZL 0,0,SNR   ;IF MORE THAN 15 LEVELS
30 03011 000422      JMP DEF66+2   ;OF INDIRECT,GIVE UP
31 03012 031000      LDA 2,0,2     ;CALCULATE EFFECTIVE
32 03013 151132      MOVZL# 2,2,SZC  ;ADDRESS, LOOK FOR INDIRECT
33 03014 000774      JMP ,=4     ;BIT 0 A ONE
34 03015 050122      STA 2,TEM     ;C(TEM)=EFFECTIVE ADDRESS
35 03016 031000      LDA 2,0,2     ;C(2)=DATA THAT ADDRESS
36
37

```

```

A 0146 .MAIN
21
22 03017 102520 DEF65: SUBZL 0,0     ;C(0)=1,LOOP ON ERROR SWITCH
23 03020 060377      NIOP CPU     ;SYNC AT 7-A74
24 03021 012137      ISZ 0CF00
25 03022 030122      LDA 3,TEM
26 03023 016122      DSZ *TEM
27 03024 150015      AD# 2,3,SNR
28 03025 000413      JMP DEF67
29 03026 063077      HALT
30 03027 102400      SUB 0,0
31 03030 052122      STA 2,TEM
32 03031 101005 DEF66: MOV 0,0,SNR
33 03032 000766      JMP DEF65+1
34 03033 125400      INC 1,1
35 03034 034131      LDA 3,FIN
36 03035 136414      SUB# 1,3,SZR
37 03036 000745      JMP DEF64+1
38 03037 000406      JMP ,+6
39
40 03040 036122 DEF67: LDA 3,TEM
41 03041 172415      SUB# 3,2,SNR
42 03042 000767      JMP DEF66
43 03043 063077      HALT
44 03044 000763      JMP DEF66+2
45 03045 000401      JMP ,+1

```

```

;INCREMENT CORRECT?
;C(2)=CORRECT,C(TEM)=ADDRESS
;C(3)=WORD FROM BUFFER
;CLEAR TO LOOP ON ERROR
;HOPEFULLY RESTORE MEMORY
;TEST THE ERR SWITCH
;LOOP ON ERROR
;ADVANCE BUFFER POINTER
;TEST FOR FINAL ADDRESS

;GO TO NEXT TEST

;ISZ WORKED TEST DSZ
;C(2)=CORRECT,C(TEM)=ADDRESS
;C(3)=DATA FROM MEMORY
;PRESS CONTINUE TO
;RESTORE AND LOOP ON ERR
;CHANGE TO LOOP TEST

```

```

A 0047 .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

IA TEST OF AUTO INCREMENT/DECREMENT
I THIS TEST WILL INSURE THAT REGISTERS 20-37
I DO NOT CHANGE VALUE WHEN REFERENCED WITHOUT A
I INDIRECT BIT.

I FIRST PASS STORES (0)
I IN REGISTERS 20-37
I SECOND PASS STORES (-1)

I INC ADDRESS
I ITERATION COUNT OF 20

I SET ITERATION COUNT
I LOOK FOR A CHANGE
I IN TO REFERENCE
I TO AUTO INC/DEC LOCATIONS
I C(2)=ADDRESS,C(0)=FIRST
I PREFERENCE,C(1)=SECOND
I GO THROUGH ALL INC/DEC
I REGISTERS

I C(0)=-1
I FIRST OR SECOND PASS
I SECOND,EXIT TEST
I C(2)=100000

I CHANGE TO LOOP TEST

I C(0)=1,LOOP ON ERROR SWITCH
I C(3)=0-7,REGISTER POINTER

I STARTING VALUE OF 0000
I SETUP AUTO INC REGISTER
I SYNC AT 7-A74
I THIS SHOULD INC THE REG
I C(1)=VALUE OF AUTO REG
I C(2)=CORRECT VALUE-1
I C(3)=0-7,FAILING REG

I CLEAR TO LOOP ON ERROR

I TEST FOR COUNT OF 100000
I NOT YET
I ADVANCE TO NEXT AUTO REG

I TEST FOR LAST REGISTER
I CHANGE TO LOOP TEST

I AUTO DECREMENT CHECK
I COUNT ALL THE AUTO DECREMENT REGISTERS FROM 200000-0
I CHECK TEN VALUE AGAINST A ADD INSTRUCTION, PRESS
I CONTINUE AFTER A ERROR TO SCOPE THE FAILURE

I STORE SOME DATA
I DID IT CHANGE VIA
I THE INDIRECT REFERENCE?
I NO ITS OK.

I C(3)=DATA STORED,C(1)=
I DATA AFTER INDIRECT.
I SYNC CPU
I BMA INPUT TO AND GATE
I FAILED.

I FIRST PASS STORES (0)
I IN REGISTERS 20-37
I SECOND PASS STORES (-1)

I INC ADDRESS
I ITERATION COUNT OF 20

I SET ITERATION COUNT
I LOOK FOR A CHANGE
I IN TO REFERENCE
I TO AUTO INC/DEC LOCATIONS
I C(2)=ADDRESS,C(0)=FIRST
I PREFERENCE,C(1)=SECOND
I GO THROUGH ALL INC/DEC
I REGISTERS

I C(0)=-1
I FIRST OR SECOND PASS
I SECOND,EXIT TEST
I C(2)=100000

I CHANGE TO LOOP TEST

I C(0)=1,LOOP ON ERROR SWITCH
I C(3)=0-7,REGISTER POINTER

I STARTING VALUE OF 0000
I SETUP AUTO DECREMENT REG
I SYNC AT 7-A74
I SHOULD DECREMENT THE REG
I C(1)=VALUE OF AUTO DEC REG
I C(2)=CORRECT VALUE+1
I C(3)=0-7,FAILING AUTO REG

I CLEAR TO LOOP TEST
I TEST FOR ZERO COUNT

I ADVANCE TO NEXT AUTO REG
I TEST FOR LAST AUTO
I DECREMENT REGISTER
I CHANGE TO LOOP TEST

```

A 0049 .MAIN

```
01
02
03 ;A TEST OF AUTO INCREMENT
04 ;EACH LOCATION IN THE BUFFER IS SET TO A ADDRESS
05 ;PATTERN-1
06 ;VIA THE AUTO INDEX REGISTERS, THE REGISTERS COUNTING
07 ;ABILITY AS WELL AS THE DATA IS CHECKED, THE PROGRAM
08 ;WILL LOOP ON ERROR IF CONTINUED AFTER A HALT,
09
10 03174 102000 ADC 0,0
11 03175 176400 ID20: SUB 3,3
12 03176 030130 LDA 2,BUFF ;FIRST BUFFER LOCATION
13 03177 051420 STA 2,20,3 ;SETUP AUTO INC REGISTER
14 03200 060377 NIOP CPU ;SYNC AT 7=A74
15 03201 053420 STA 2,020,3 ;STORE C(2) VIA AUTO INC
16 03202 025420 LDA 1,20,3 ;C(AUTO) TO C(1)
17 03203 146015 ID21: ADC# 2,1,SNR ;CHECK INCREMENT FEATURE
18 03204 000415 JMP ID23 ;ITS OK
19 03205 063077 HALT ;C(1)=VALUE OF AUTO INC REG
20 03206 102400 SUB 0,0 ;C(2)=CORRECT VALUE-1
21 03207 041000 ID22: STA 0,0,2 ;C(3)+20=AUTO REGISTER INVOLVED
22 03210 112400 SUB 0,2 ;C(BUFFER) DESTROYED AFTER TEST
23 03211 024131 LDA 1,FIN ;C(1)=FINAL ADDRESS
24 03212 132414 SUB# 1,2,SZR ;TEST FOR END OF BUFFER
25 03213 000764 JMP ID20+2
26 03214 175400 INC 3,3 ;GO TO NEXT AUTO REGISTER
27 03215 024111 LDA 1,C10 ;TEST FOR LAST ONE
28 03216 136414 SUB# 1,3,SZR
29 03217 000757 JMP ID20+1
30 03220 000406 JMP ID24 ;GO TO NEXT TEST
31
32 03221 025001 ID23: LDA 1,1,2 ;AUTO INC VALUE OK BUT
33 03222 132415 SUB# 1,2,SNR ;DATA STORED IS WRONG
34 03223 000764 JMP ID22 ;C(1)=DATA FROM MEMORY
35 03224 063077 HALT ;C(2)=CORRECT VALUE
36 03225 000761 JMP ID22-1 ;C(3)+20=AUTO REGISTER
37 03226 000401 ID24: JMP ,+1 ;CHANGE TO LOOP TEST
38
```

A 0050 .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 03227 102000 ID30: ADC 0,0 ;C(0)=-1, LOOP ON ERR SWITCH
10 03230 176400 SUB 3,3 ;C(3)=0, FIRST INDEX USED
11 03231 030131 LDA 2,FIN ;FINAL BUFFER ADDRESS
12 03232 051430 STA 2,30,3 ;TO AUTO DECREMENT REGISTER
13 03233 060377 NIOP CPU ;SYNC AT 7=A74
14 03234 053430 STA 2,030,3 ;STORE C(2) IN BUFFER VIA
15 03235 025430 LDA 1,30,3 ;C(1)=VALUE OF AUTO DEC REG
16 03236 132015 ID31: ADC# 1,2,SNR ;C(2)=CORRECT VALUE+1
17 03237 000415 JMP ID33 ;C(3)+30=AUTO REGISTER
18 03240 063077 HALT ;AUTO DECREMENT FAILED
19 03241 102400 SUB 0,0 ;CLEAR TO LOOP ON ERROR
20 03242 041000 ID32: STA 0,0,2 ;CLEAR OUT THE PATTERN
21 03243 113000 ADD 0,2 ;C(2)=C(2)-1
22 03244 024130 LDA 1,BUFF ;TEST FOR BEGIN OF BUFFER
23 03245 132414 SUB# 1,2,SZR
24 03246 000764 JMP ID30+3
25 03247 175400 INC 3,3 ;ADVANCE TO NEXT AUTO REG
26 03250 024111 LDA 1,C10
27 03251 136414 SUB# 1,3,SZR ;TEST FOR END
28 03252 000757 JMP ID30+2 ;AGAIN WITH THIS REGISTER
29 03253 000406 JMP ID34 ;GO TO NEXT TEST
30
31 03254 025377 ID33: LDA 1,-1,2 ;AUTO DEC VALUE IS OK
32 03255 132415 SUB# 1,2,SNR ;BUT THE DATA STORED IS NOT
33 03256 000764 JMP ID32 ;C(1)=DATA FROM BUFFER
34 03257 063077 HALT ;C(2)=CORRECT VALUE
35 03260 000761 JMP ID32-1 ;C(3)+30=AUTO REGISTER
36
37 03261 000401 ID34: JMP ,+1 ;CHANGE TO LOOP TEST
38
```

```

A 0051 .MAIN
01
02      JA TEST OF AUTO INC USING ISZ
03      JTHE PROGRAM STORES A (-1) IN THE BUFFER.
04      JA ISZ INSTRUCTION VIA AUTO INCREMENT REGISTER 27
05      JTHEN INCREMENTS THE BUFFER. IF A ERROR SHOULD OCCURE
06      JPRESSING CONTINUE WILL PLACE THE PROGRAM IN A FAILING
07      JLOOP UNTILL RESTARTED.
08
09 03262 102520 ID40: SUBZL 0,0      JC(0)=1,LOOP ON ERROR SWITCH
10 03263 126000      ADC 1,1      JC(1)=-1
11 03264 030130      LDA 2,BUFF      JFIRST LOCATION IN THE BUFFER
12 03265 034131      LDA 3,FIN      JFINAL LOCATION IN THE BUFFER
13 03266 050027 ID41: STA 2,27      JSETUP AUTO INCREMENT REG
14 03267 045001      STA 1,1,2      JSTORE (-1) IN MEMORY
15 03270 060377      NIOP CPU      JSYNC AT 7-74
16 03271 012027      ISZ #27      JINCREMENT AND SKIP
17 03272 102401      SUB 0,R,SKP    JISZ FAIL TO SKIP
18 03273 113001      ADD 0,2,SKP    JINC TO NEXT LOC
19 03274 063077      HALT          JEXAMINE LOC 27
20 03275 156414      SUB# 2,3,SZR   JTEST FOR END OF BUFFER
21 03276 000770      JMP ID41
22 03277 000401 ID42: JMP ,+1      JCHANGE TO LOOP TEST
23
24      JA TEST OF AUTO INC/DEC
25      JVIA REGISTER 34 THE BUFFER IS FILLED WITH JSR INST.
26      JJA JSR VIA REGISTER 25 IS EXECUTED,THE RETURN FROM
27      JTHE BUFFER IS CHECKED. PRESS CONTINUE TO LOOP ERROR.
28
29 03300 024425 ID44: LDA 1,CIDJ      JA CONSTANT (JSR 0,3)
30 03301 034130      LDA 3,BUFF      JFIRST BUFFER LOCATION
31 03302 030131      LDA 2,FIN      JFINAL BUFFER LOCATION
32 03303 050034      STA 2,34      JC(34) POINTS TO THE BUFFER
33 03304 156400      SUB 2,3      JC(3)= -WORDS IN BUFFER
34 03305 040034      STA 1,034     JFILL BUFFER WITH (JSR)
35 03306 178404      INC 3,3,SZR
36 03307 000776      JMP , -2
37
38 03310 102520 ID45: SUBZL 0,0      JC(0)=1,LOOP ON ERROR SWITCH
39 03311 024130      LDA 1,BUFF
40 03312 044025      STA 1,25      JSETUP AUTO INDEX REG
41 03313 006025      JSR #25      JGO TO THE BUFFER
42 03314 174400      NEG 3,3      JEFFECTIVELY SUBTRACT 1
43 03315 174000      COM 3,3      JFROM C(3)
44 03316 136014      ADC# 1,3,SZR  JC(3)=POINT BUFFER ENTERED
45 03317 102401      SUB 0,0,SKP   JC(2)=CORRECT-1
46 03320 107001      ADD 0,1,SKP   JC(25)=AUTO INC VALUE
47 03321 003077      HALT          J25=AUTO INC REGISTER
48 03322 132014      ADC# 1,2,SZR  JTEST FOR END OF BUFFER
49 03323 000767      JMP ID45+2
50 03324 101011 ID46: MOV# 0,0,SKP    JCHANGE TO LOOP TEST
51 03325 005400 CIDJ: JSR 0,3      JA CONSTANT
52

```

```

A 0052 .MAIN
01
02 03326 024130 ID50: LDA 1,BUFF      JFIRST LOCATION OF THE BUFF
03 03327 030131      LDA 2,FIN      JFINAL LOCATION OF THE BUFF
04 03330 102520      SUBZL 0,0      JC(0)=1,LOOP ON ERROR SWIT
05 03331 050033      STA 2,33      JSETUP AUTO DECREMENT
06 03332 006033      JSR #33      JGO TO THE BUFFER
07 03333 156414      SUB# 2,3,SZR   JC(3)=PC FROM BUFFER
08 03334 102401      SUB 0,0,SKP   JC(2)=INITIAL VALUE OF AUTO
09 03335 112401      SUB 0,2,SKP   JDEC REGISTER 33
10 03336 003077      HALT          JPROG WILL LOOP ON ERROR
11 03337 132014      ADC# 1,2,SZR  JTEST FOR BEGIN OF BUFF
12 03340 000771      JMP ID50+3
13
14 03341 034402 DGCA: LDA 3,,+2      JTEST DSZ COUNT IN
15 03342 152001      ADC 2,2,SKP   JEAACH MEMORY MODULE.
16 03343 004000      4000          JC(2)=ERROR SWITCH.
17 03344 020125      LDA 0,MSIZE
18 03345 116432      SUBZ# 0,3,SZC
19 03346 000421      JMP LAST
20 03347 102620      SUBZR 0,0
21 03350 143005 DGCX: ADD 2,0,SNR
22 03351 000413      JMP DGCB
23 03352 041400      STA 0,0,3
24 03353 060377      NIOP CPU      JSYNC 7-A74
25 03354 015400      DSZ 0,3
26 03355 000401      JMP ,+1
27 03356 025400      LDA 1,0,3
28 03357 122015      ADC# 1,0,SNR  JC(1)=RESULT OF DSZ
29 03360 000770      JMP DGCB      JC(0)=VALUE BEFORE DSZ
30 03361 152400      SUB 2,2
31 03362 003077      HALT
32 03363 000765      JMP DGCB
33 03364 020757 DGCB: LDA 0,DGCA+2    JGO TO NEXT LOCATION
34 03365 117000      ADD 0,3
35 03366 000754      JMP DGCA+1
36 03367 024046 LAST: LDA 1,EGGS
37 03370 125005      MOV 1,1,SNR
38 03371 002410      JMP 0,+0
39 03372 014001      DSZ EGGS+3
40 03373 000407      JMP ,+7
41 03374 062077      IORST
42 03375 034052      LDA 3,EGGS+4
43 03376 020051      LDA 0,EGGS+3
44 03377 041770      STA 0,-2,3
45 03400 001400      JMP 0,3
46 03401 000403      BEG1
47 03402 010050      ISZ EGGS+2
48 03403 002141      JMP #MAIN
49 03404 006172      JSR #ICAT
50 03405 002141      JMP #MAIN

```

```

^ 0053 .MAIN
PI
02 03406 047503
03407 054520
03410 044522
03411 044107
03412 024124
03415 024503
03414 043504
03415 026103
03416 034461
03417 034466
03420 033454
03421 026060
03422 030467
03423 033454
03424 026053
03425 032067
03426 033454
03427 040465 ALL RIGHTS RESERVED/
03430 046114
03431 051040
03432 043511
03433 052110
03434 026123
03435 042522
03436 042523
03437 053122
03440 042100
03441 000000

04 03442 003403 PRGEND: DIRT=20
05 03442 003403 DIRT1 .TXTE IEXERCISER071
06 03443 154305
03444 151305
03445 144703
03446 142523
03447 033322
03450 000207
03451 000010
06 03452 000176
09 03453 000002
10 03454 000000
11 03455 000000
12 03456 000000
13 03457 000000
14 03460 000000
15
16

0054 .MAIN
ACTIO 000105 5/45 7/05 7/08 7/32 7/33 8/32 11/51 12/26
13/02 15/20 15/49 16/17
BEG1 000400 5/06 7/04
BUF20 000133 6/11 24/31 26/26
BUFF 000130 6/08 10/15 10/35 19/05 19/37 19/48 20/11 20/25
21/21 21/29 22/18 23/19 23/36 24/13 25/14 25/32
27/06 27/16 28/38 31/21 31/28 34/38 35/12 35/21
36/10 37/14 37/20 38/39 39/12 39/19 39/40
39/45 40/10 40/17 41/09 41/23 42/16 42/17 42/25

000111
000143
000174
000106
000104
000140
000110
000144
000175
000174
000644
000134
000142
000151
000136
000124
000126
000173
000135
001405
000137
000325
001377
000672
001517
001000
000615
000616
000617
001554
001110
000123
000140
000112
000375
002414
000424
000437
000457
000460
000477
000500
000507
000521

C10
C1040
C1347
C1355
C170
C174X
C177
C2002
C2002
C24
C3
C3000
C37
C377
C4
C40
C400
C5
C5252
C700
C10J
C1NC
C1DF1
C1JMP1
C1JRR1
C1JRR2
C1JRR3
C1JRR4
C1DAP
C1CLOCK
C1S
C1PINC
C1CRAN
DEF0
DEF10
DEF11
DEF14
DEF15
DEF20
DEF21
DEF30
DEF31
DEF32

42/35
51/11
5/49
11/13
8/03
43/07
43/35
11/36
17/18
5/44
6/16
5/48
8/27
6/47
6/16
11/14
11/29
6/12
9/03
6/18
8/02
6/27
11/00
6/14
23/40
24/34
25/40
26/34
28/00
28/30
32/10
38/11
6/04
6/06
11/10
47/43
6/45
8/22
6/13
25/24
30/11
30/13
34/10
19/07
19/24
44/30
45/26
46/04
51/29
51/51
27/02
27/08
42/18
42/51
22/19
22/30
21/22
21/41
41/32
41/31
41/25
41/33
23/25
23/51
5/22
14/54
6/03
34/12
27/13
27/25
5/56
11/30
12/40
13/42
14/24
15/02
16/10
16/04
17/11
36/22
37/13
37/30
38/24
38/25
38/38
38/52
38/53
39/11
39/28
39/29

43/22
43/18
44/22
45/11
45/23
49/12
49/22
50/22
49/27
50/26
13/16
13/46
14/31
15/00
16/31
17/07
47/31
14/04
15/10
8/31
8/33
8/27
6/19
6/03
5/46
11/36
5/44
6/16
5/48
11/30
8/27
6/47
6/16
11/14
11/29
6/12
9/03
6/18
8/02
6/27
11/00
6/14
23/40
24/34
25/40
26/34
28/00
28/30
32/10
38/11
6/04
6/06
11/10
47/43
6/45
8/22
6/13
25/24
30/11
30/13
34/10
19/07
19/24
44/30
45/26
46/04
51/29
51/51
27/02
27/08
42/18
42/51
22/19
22/30
21/22
21/41
41/32
41/31
41/25
41/33
23/25
23/51
5/22
14/54
6/03
34/12
27/13
27/25
5/56
11/30
12/40
13/42
14/24
15/02
16/10
16/04
17/11
36/22
37/13
37/30
38/24
38/25
38/38
38/52
38/53
39/11
39/28
39/29

/TXT /COPYRIGHT(C)DGC,1969,70,71,73,74,75

```

0055 .MAIN

DEF34 002522 39/38
 DEF35 002531 39/45
 DEF36 002544 39/55
 DEF40 002545 40/09
 DEF41 002554 40/17
 DEF42 002571 40/39
 DEF50 002639 42/15
 DEF51 002643 42/26
 DEF52 002650 42/32
 DEF53 002656 42/39
 DEF54 002671 42/50
 DEF60 002734 44/30
 DEF61 002751 44/34
 DEF62 002762 44/42
 DEF63 002766 44/44
 DEF64 002766 44/48
 DEF65 003002 45/23
 DEF66 003017 46/02
 DEF67 003031 45/30
 DEF68 003040 46/08
 DGA 003341 52/14
 DGB 003364 52/22
 DGC 003359 52/21
 DIRT 003443 5/03
 DYSMI 000171 6/43

0056 .MAIN

HSR1 000745 5/20
 HSR3 000763 12/39
 HSR8 000873 5/35
 HSRCH 000074 5/36
 ICAT 000172 6/44
 ID0 003046 47/07
 ID1 003055 47/15
 ID10 003123 48/09
 ID11 003147 48/29
 ID14 003150 48/36
 ID15 003173 48/55
 ID2 003072 47/26
 ID20 003175 49/11
 ID21 003203 49/17
 ID22 003207 49/21
 ID23 003221 49/18
 ID24 003226 49/30
 ID30 003227 50/09
 ID31 003236 50/16
 ID32 003242 50/28
 ID33 003254 50/17
 ID34 003261 50/29
 ID4 003073 47/31
 ID40 003202 51/00
 ID41 003266 51/13
 ID42 003277 51/22
 ID44 003300 51/29
 ID45 003310 51/38
 ID46 003324 51/50
 ID5 003315 47/42
 ID50 003326 52/02
 ID6 003122 47/49
 INTR 000012 5/04
 ISAV0 000113 5/51
 ISAV1 000114 5/52
 ISAV2 000115 5/53
 ISAV3 000116 5/54
 ISAVC 000117 5/55
 IS0 001743 20/07
 IS21 001747 20/11
 IS210 002030 20/23
 IS211 002045 20/44
 IS212 002051 20/40
 IS213 002056 20/35
 IS214 002062 20/50
 IS22 001762 20/29
 IS220 002063 30/11
 IS221 002070 30/16
 IS222 002106 30/15
 IS23 001766 20/33
 IS230 002107 30/36
 IS231 002113 30/40
 IS232 002120 30/43
 IS233 002134 30/50
 IS24 002001 20/07
 IS240 002135 31/18
 IS241 002146 31/20
 IS242 002154 31/31
 IS25 002004 20/10

0057 .MAIN

DEF34 002522 39/38
 DEF35 002531 39/45
 DEF36 002544 39/55
 DEF40 002545 40/09
 DEF41 002554 40/17
 DEF42 002571 40/39
 DEF50 002639 42/15
 DEF51 002643 42/26
 DEF52 002650 42/32
 DEF53 002656 42/39
 DEF54 002671 42/50
 DEF60 002734 44/30
 DEF61 002751 44/34
 DEF62 002762 44/42
 DEF63 002766 44/44
 DEF64 002766 44/48
 DEF65 003002 45/23
 DEF66 003017 46/02
 DEF67 003031 45/30
 DEF68 003040 46/08
 DGA 003341 52/14
 DGB 003364 52/22
 DGC 003359 52/21
 DIRT 003443 5/03
 DYSMI 000171 6/43

0058 .MAIN

DEF34 002522 39/38
 DEF35 002531 39/45
 DEF36 002544 39/55
 DEF40 002545 40/09
 DEF41 002554 40/17
 DEF42 002571 40/39
 DEF50 002639 42/15
 DEF51 002643 42/26
 DEF52 002650 42/32
 DEF53 002656 42/39
 DEF54 002671 42/50
 DEF60 002734 44/30
 DEF61 002751 44/34
 DEF62 002762 44/42
 DEF63 002766 44/44
 DEF64 002766 44/48
 DEF65 003002 45/23
 DEF66 003017 46/02
 DEF67 003031 45/30
 DEF68 003040 46/08
 DGA 003341 52/14
 DGB 003364 52/22
 DGC 003359 52/21
 DIRT 003443 5/03
 DYSMI 000171 6/43

D3Z0 003163 32/00
 D3Z1 003167 32/14
 D3Z10 002222 33/15
 D3Z11 002224 33/39
 D3Z12 002245 33/20
 D3Z13 002253 33/43
 D3Z14 002260 33/37
 D3Z2 002202 32/32
 D3Z20 002261 34/10
 D3Z21 002265 34/14
 D3Z22 002303 34/28
 D3Z3 002205 32/35
 D3Z30 002304 34/30
 D3Z31 002310 34/40
 D3Z32 002323 34/43
 D3Z33 002331 34/50
 D3Z4 002314 32/30
 D3Z40 002332 35/09
 D3Z41 002343 35/19
 D3Z42 002346 35/23
 D3Z43 002361 35/26
 D3Z5 002220 32/41
 E6G5 000046 5/10
 ENDIT 000634 6/43
 FICH 000767 45/11
 FIN 000131 6/09

12/25 12/30 12/45 13/18
 14/52 14/58 15/19 15/25
 16/33 16/39 17/05 17/09
 46/22 46/24
 52/35 52/32
 52/29 53/06
 13/27 13/47 13/50 13/58
 15/44 15/48 15/58
 17/06 17/39
 32/00 32/14
 33/15 33/28
 33/39 33/41
 33/20 33/43
 33/37 33/40
 34/10 34/27
 34/14 34/28
 32/35 32/45
 34/30 34/36
 34/40 34/46
 34/43 34/51
 34/50 34/57
 32/30 32/42
 35/09 35/16
 35/19 35/33
 35/23 35/30
 35/26 35/38
 32/41 32/46
 5/10 5/11
 6/43 6/43
 45/11 45/21
 6/09 6/20
 22/17 24/14
 31/20 33/37
 39/39 35/11
 46/11 41/10
 45/23 50/11
 6/10 8/30
 12/32 13/07

13/02 13/16
 12/33 12/34
 12/35 12/42
 5/26 5/28
 47/07 47/28
 48/09 48/24
 48/36 48/54
 48/36 48/54
 48/55 48/55
 47/26 47/29
 49/11 49/25
 49/17 49/34
 49/21 49/34
 49/18 49/32
 49/30 49/37
 50/09 50/24
 50/16 50/33
 50/17 50/31
 50/29 50/37
 47/31 47/46
 51/22 51/21
 51/29 51/49
 51/50 51/58
 47/42 47/51
 52/02 52/12
 47/49 47/56
 5/04 11/01
 5/51 11/01
 5/52 11/02
 5/53 11/03
 5/54 11/04
 5/55 11/06
 20/07 20/19
 20/11 20/31
 20/23 20/45
 20/44 20/45
 20/40 20/49
 20/35 20/54
 20/50 20/58
 20/29 20/58
 30/11 30/29
 30/16 30/30
 30/15 30/33
 20/41 20/41
 30/36 30/40
 30/40 30/49
 30/43 30/51
 30/50 30/57
 31/18 31/41
 31/20 31/33
 31/31 31/35
 20/10 20/20

0259 .MAIN

.LSRC	000104	6/38	16/50	17/02					
.PRAN	000154	6/30	15/03	15/09					
.PTP	000053	6/51	10/19	15/16	15/31	15/32	15/34	15/35	15/36
.PTR	000052	6/52	10/26	10/28	15/17	15/18	15/24	15/46	15/47
		15/57	16/02	16/20					
.PUN	001115	5/29	15/02						
.PUN1	001145	15/07	15/28						
.PUNC	000155	6/31	10/22	15/05	15/28	15/30	15/39	16/23	
.PUNX	001144	15/14	15/26						
.READ	000156	6/32	10/30	15/40	16/06	16/18	16/20	16/24	
.RRAN	000153	6/29	16/11	16/32					
.STP	000152	6/28	10/23	15/11					
.STT	000170	6/42	10/15	17/21					
.TIN	000162	6/36	10/06	16/44	16/45	16/47			
.TINE	000161	6/35	10/02	10/07	16/35				
.TISD	000557	7/50	10/02						
.TISU	000061	7/26	10/04						
.TOSD	000071	7/52	10/13						
.TOSU	000566	7/30	10/10						
.TPSD	000600	7/46	10/21						
.TPSU	000575	7/22	10/18						
.TRAN	000167	6/41	17/12	17/19					
.TRSD	000004	7/48	10/26						
.TRSU	000606	7/24	10/28						
.TTI	000050	6/53	10/04	16/38	16/40	16/41	16/42		
.TTO	000051	6/54	10/11	17/25	17/32	17/33	17/35	17/36	17/37
.TYO	001275	5/27	17/11						
.TYO1	001316	17/17	17/29						
.TYPE	000166	6/40	10/14	17/15	17/29	17/31			
.XLAN	001041	14/11	14/12						