DATA GENERAL
CORPORATION
Southboro,
Massachusetts 01772
(617) 485-9100

PROGRAM

Super Logic Test

TAPES

Binary: 085-000002

ABSTRACT

Super Logic Test is a maintenance program designed to test the Supernova Central Processing Unit. It is a gate by gate test of the logic used to implement the Supernova instruction set. The test does not include any input-output or optional equipment.

087-000001-00

```
;                    SUPER LOGIC TEST

;1.      ABSTRACT
;            SUPER LOGIC TEST IS A MAINTENANCE PROGRAM
;            DESIGNED TO TEST THE SUPERNOVA CENTRAL
;            PROCESSING UNIT. IT IS A GATE BY GATE TEST
;            OF THE LOGIC USED TO IMPLEMENT THE SUPER-
;            NOVA INSTRUCTION SET. THE TEST DOES NOT IN-
;            CLUDE ANY INPUT-OUTPUT OR OPTIONAL EQUIP-
;            MENT.

;2.      MACHINE REQUIREMENTS
;2.1         SUPERNOVA PROCESSOR
;2.2         2K READ/WRITE MEMORY

;3.      SWITCH SETTINGS
;            STARTING ADDRESS=000077

;4.      OPERATING PROCEEDURE
;4.1         LOAD THE PROGRAM VIA THE BINARY LOADER
;4.2         SET THE SWITCHES TO 000077
;4.3         PRESS START
;4.4         MACHINE SHOULD HALT. PRESS CONTINUE

;5.      ERROR DESCRIPTION
;5.1         THE HALT INSTRUCTION IS USED TO INDICATE
;            ERRORS.
;5.2         WHEN A ERROR IS DETECTED RECORD THE STATE
;            OF THE MACHINE. CONSULT THE LISTING FOR
;            POSSIABLE CAUSES OF FAILURE. CONSTRUCT A
;            LOOP WHICH WILL REPRODUCE THE ERROR.
;            SCOPE THE LOGIC.

;6.      PROGRAM DESCRIPTION
;            THIS PROGRAM IS A COLLECTION OF SMALL
;            ROUTINES EACH DESIGNED TO TEST A PORTION
;            OF THE PROCESSOR LOGIC. EACH ROUTINE IS
;            DESIGNED TO TEST AS SMALL A PART OF THE
;            LOGIC AS POSSIABLE. EACH TEST IN THE SEQ-
;            UENCE IS BASED ON PREVIOUS TEST WORKING.

;7.      MISC
;            THE TIME FOR ONE COMPLETE PASS IS MEASURED
;            IN MILLISECONDS.
```

```
        000000  .LOC 0
00000 063077            HALT
00001 063077            HALT
00002 063077            HALT                    ;START AT 100,NOT HERE
00003 063077            HALT

        000040  .LOC 40
00040 063077            HALT                    ;START AT 100,NOT HERE
00041 000000 K0:        0
00042 000001 KB15:      1
00043 000002 KB14:      2
00044 000004 KB13:      4
00045 000010 KB12:      10
00046 000020 KB11:      20
00047 000040 KB10:      40
00050 000100 KB9:       100
00051 000200 KB8:       200
00052 000400 KB7:       400
00053 001000 KB6:       1000
00054 002000 KB5:       2000
00055 004000 KB4:       4000
00056 010000 KB3:       10000
00057 020000 KB2:       20000
00060 040000 KB1:       40000
00061 100000 KB0:       100000

00062 177777 M1:        -1
00063 125252 K12525:    125252
00064 052525 K52525:    052525
00065 000377 K377:      377
00066 177701 M77:       -77
00067 065432 K65432:    65432
00070 177601 M177:      -177
00071 000077 K77:       77
00072 000000 PASS:      0

        000077  .LOC 77                        ;PROGRAM STARTS AT 77
00077 063077            HALT

                .EOT
```

```
00100 102021 A1:    ADCZ 0,0,SKP    ;FAIL TO SKIP, CHECK ALC
00101 063077        HALT            ;SKIP FLOP ECT. (CPU-2)

00102 102020 A2:    ADCZ 0,0        ;IR15(0) BUT ADC SKIPED.
00103 102021        ADCZ 0,0,SKP    ;CHECK SKIP LOGIC (CPU-2)
00104 063077        HALT

00105 102441 A3:    SUBO 0,0,SKP    ;FAIL TO SKIP WHEN AC=0.
00106 063077        HALT            ;CHECK IR13 INPUT TO SKIP
                                    ;LOGIC. (CPU-2)

00107 020111 A4:    LDA 0,.+2       ;LOAD AC WITH A NUMBER
00110 101025        MOVZ 0,0,SNR    ;NOT A ZERO. FAIL TO SKIP.
00111 063077        HALT            ;CHECK ADDER=0 INPUT
                                    ;TO SKIP LOGIC, ETC.

00112 020041 A5:    LDA 0,K0        ;LOAD AC WITH ZEROS!
00113 101025        MOVZ 0,0,SNR    ;THIS INSTRUCTION CAUSED SKIP.
00114 101021        MOVZ 0,0,SKP    ;CHECK ADDER=0 INPUT TO
00115 063077        HALT            ;SKIP LOGIC. (CPU-2)

00116 020041 A6:    LDA 0,K0        ;AC0 LOADED WITH ZEROS.
00117 101024        MOVZ 0,0,SZR    ;THIS INSTRUCTION FAILS TO SKIP.
00120 063077        HALT            ;CHECK ALC SKIP LOGIC. (CPU-2)
                                    ;PERHAPS (ALC SKIP ENAB) IS GRO

00121 020124 A7:    LDA 0,.+3       ;AC0 LOADED WITH NOT ZERO.
00122 101024        MOVZ 0,0,SZR    ;BUT THIS INSTRUCTION SKIPED.
00123 101021        MOVZ 0,0,SKP    ;CHECK TO MAKE SURE AC0
00124 063077        HALT            ;IS REALLY NOT A ZERO.

00125 020043 A9:    LDA 0,KB14      ;CHECK ADDER=0
00126 101025        MOVZ 0,0,SNR    ;BIT 14
00127 063077        HALT

00130 020044 A10:   LDA 0,KB13      ;CHECK ADDER=0
00131 101025        MOVZ 0,0,SNR    ;BIT 13
00132 063077        HALT

00133 020045 A11:   LDA 0,KB12      ;CHECK ADDER=0
00134 101025        MOVZ 0,0,SNR    ;BIT 12
00135 063077        HALT

00136 020046 A12:   LDA 0,KB11      ;CHECK ADDER=0
00137 101025        MOVZ 0,0,SNR    ;BIT 11
00140 063077        HALT
```

```
00141 020047 A13:    LDA 0,KB10        ;CHECK ADDER=0
00142 101025         MOVZ 0,0,SNR      ;BIT 10
00143 063077         HALT

00144 020050 A14:    LDA 0,KB9         ;CHECK ADDER=0
00145 101025         MOVZ 0,0,SNR      ;BIT 9
00146 063077         HALT

00147 020051 A15:    LDA 0,KB8         ;CHECK ADDER=0
00150 101025         MOVZ 0,0,SNR      ;BIT 8
00151 063077         HALT

00152 020052 A16:    LDA 0,KB7         ;CHECK ADDER=0
00153 101025         MOVZ 0,0,SNR      ;BIT 7
00154 063077         HALT

00155 020053 A17:    LDA 0,KB6         ;CHECK ADDER=0
00156 101025         MOVZ 0,0,SNR      ;BIT 6
00157 063077         HALT

00160 020054 A18:    LDA 0,KB5         ;CHECK ADDER=0
00161 101025         MOVZ 0,0,SNR      ;BIT 5
00162 063077         HALT

00163 020055 A19:    LDA 0,KB4         ;CHECK ADDER=0
00164 101025         MOVZ 0,0,SNR      ;BIT 4
00165 063077         HALT

00166 020056 A20:    LDA 0,KB3         ;CHECK ADDER=0
00167 101025         MOVZ 0,0,SNR      ;BIT 3
00170 063077         HALT

00171 020057 A21:    LDA 0,KB2         ;CHECK ADDER=0
00172 101025         MOVZ 0,0,SNR      ;BIT 2
00173 063077         HALT

00174 020060 A22:    LDA 0,KB1         ;CHECK ADDER=0
00175 101025         MOVZ 0,0,SNR      ;BIT 1
00176 063077         HALT

00177 020061 A23:    LDA 0,KB0         ;CHECK ADDER=0
00200 101045         MOVO 0,0,SNR      ;BIT 0
00201 063077         HALT

00202 020042 A24:    LDA 0,KB15        ;CHECK ADDER=0
00203 101045         MOVO 0,0,SNR      ;BIT 15
00204 063077         HALT

00205 020042 A26:    LDA 0,KB15        ;FALSE SHR SIGNEL TO
00206 101025         MOVZ 0,0,SNR      ;ADDER=0 BIT 15 AND GATE.
00207 063077         HALT

00210 020061 A27:    LDA 0,KB0         ;FALSE SHL SIGNEL TO
00211 101025         MOVZ 0,0,SNR      ;ADDER=0 BIT 0 AND GATE.
00212 063077         HALT
```

```
00213 020041 A28:   LDA 0,K0          ;FAIL TO SKIP ON ZERO WHEN
00214 101044        MOVO 0,0,SZR      ;CARRY IS SET. CHECK (ALC
00215 063077        HALT              ;SKIP ENAB) LEVEL.

00216 020221 A29:   LDA 0,.+3         ;INSTRUCTION SZR TESTED,
00217 101044        MOVO 0,0,SZR      ;WHEN NO ROTATE IS PRESENT.
00220 101041        MOVO 0,0,SKP
00221 063077        HALT

00222 020041 A30:   LDA 0,K0          ;INSTRUCTION SNR FULLY
00223 101045        MOVO 0,0,SNR      ;TESTED WHEN NO
00224 101001        MOV 0,0,SKP       ;ROTATE IS PRESENT!
00225 063077        HALT

00226 102041 A32:   ADCO 0,0,SKP      ;SKIP TESTED WITH ALL
00227 063077        HALT              ;CASES OF CARRY AND ADDER=0!

00230 102420 A33:   SUBZ 0,0          ;TEST FOR NOT
00231 101041        MOVO 0,0,SKP      ;SKIP WITH ALL
00232 063077        HALT              ;CASES OF CARRY AND ADDER=0!

00233 102440 A34:   SUBO 0,0
00234 101021        MOVZ 0,0,SKP
00235 063077        HALT

00236 102040 A35:   ADCO 0,0
00237 101021        MOVZ 0,0,SKP
00240 063077        HALT

00241 020041 A36:   LDA 0,K0          ;IF CARRY=1 CHECK
00242 101120        MOVZL 0,0         ;PREV CARRY AND CARRY
00243 103422        ANDZ 0,0,SZC      ;SET LOGIC. IF C(CARRY)=0
00244 063077        HALT              ;CHECK ENABLE CARRY AND
                                      ;ALC SKIP LOGIC.

00245 020041 A37:   LDA 0,K0          ;CHECK ENAB CARRY
00246 101120        MOVZL 0,0         ;INPUTS TO ALC SKIP
00247 103423        ANDZ 0,0,SNC      ;LOGIC.
00250 101001        MOV 0,0,SKP
00251 063077        HALT

00252 020042 A38:   LDA 0,KB15        ;ENAB CARRY FAILED.
00253 105120        MOVZL 0,1         ;CHECK AND OF (ADDER=15,
00254 103422        ANDZ 0,0,SZC      ;SHR). SHR SHOULD NOT BE
00255 063077        HALT              ;PRESENT DURING THE AND INST.

00256 020061 A39:   LDA 0,KB0         ;ENAB CARRY FAILED.
00257 105220        MOVZR 0,1         ;CHECK AND OF (ADDER=0
00260 103422        ANDZ 0,0,SZC      ;SHL). SHL SHOULD NOT BE
00261 063077        HALT              ;PRESENT DURING THE AND INST.
```

```
00262 020061 A40:   LDA 0,KB0       ;PREV CRY LOGIC FAILED.
00263 101120        MOVZL 0,0       ;CARRY FAILED WHEN OLD
00264 103422        ANDZ 0,0,SZC    ;STATE WAS ONE! AND GATE
00265 063077        HALT            ;(IR11,CARRY(1)).

00266 020041 A41:   LDA 0,K0        ;PREV CRY LOGIC FAILED.
00267 101120        MOVZL 0,0       ;CARRY FAIL TO SET WHEN
00270 103443        ANDO 0,0,SNC    ;OLD STATE WAS ZERO, ALSO
00271 063077        HALT            ;CHECK CARRY SET,ENAB CARRY.

00272 020061 A42:   LDA 0,KB0       ;PREV CRY LOGIC FAILED.
00273 101120        MOVZL 0,0       ;CARRY FAIL TO SET WHEN
00274 103443        ANDO 0,0,SNC    ;OLD STATE ONE.
00275 063077        HALT

00276 101020 A43:   MOVZ 0,0        ;CHECK COMP CARRY
00277 103463        ANDC 0,0,SNC    ;FROM ZERO TO ONE.
00300 063077        HALT            ;CHECK CARRY FLOP,
                                    ;LOAD CARRY ETC.

00301 101040 A44:   MOVO 0,0        ;CHECK COMP CARRY
00302 103462        ANDC 0,0,SZC    ;FROM ONE TO ZERO STATE
00303 063077        HALT            ;CHECK CARRY FLOP ETC.

00304 103420 A46:   ANDZ 0,0        ;WITH BITS 10-11 ZERO
00305 103402        AND 0,0,SZC     ;CARRY SHOULD NOT
00306 063077        HALT            ;CHANGE FROM ZERO.

00307 103440 A47:   ANDO 0,0        ;WITH BITS 10-11 ZERO
00310 103403        AND 0,0,SNC     ;CARRY SHOULD NOT
00311 063077        HALT            ;CHANGE FROM ONE.

00312 020060 A48:   LDA 0,KB1
00313 103442        ANDO 0,0,SZC
00314 103403        AND 0,0,SNC
00315 063077        HALT

00316 020041 A49:   LDA 0,K0
00317 103442        ANDO 0,0,SZC
00320 103443        ANDO 0,0,SNC
00321 063077        HALT

00322 020060 A50:   LDA 0,KB1       ;THE CARRY SKIP
00323 103422        ANDZ 0,0,SZC    ;LOGIC IS TESTED!
00324 063077        HALT
```

```
00325 103441 A51:    ANDO 0,0,SKP     ;LOAD CARRY WAS PRESENT
00326 103430         ANDZ# 0,0        ;DURING THIS INSTRUCTION.
00327 103403         AND 0,0,SNC      ;CHECK LOAD CARRY LOGIC
00330 063077         HALT             ;ON REGISTER LOAD CONTROL
                                      ;PRINT. (AND GATE WITH PTG5)

00331 103421 A52:    ANDZ 0,0,SKP     ;SAME AS ABOVE BUT
00332 103450         ANDO# 0,0        ;CHECK INPUTS TO CARRY
00333 103402         AND 0,0,SZC      ;FLOP.
00334 063077         HALT

00335 103421 A53:    ANDZ 0,0,SKP     ;CHECK FOR FALSE LOAD
00336 103440         ANDO 0,0         ;OF CARRY WHEN INSTRUCTION
00337 103402         AND 0,0,SZC      ;IS SKIPED. AND GATE
00340 063077         HALT             ;(ALC,ALC SKIP,IR12)

00341 103440 A54:    ANDO 0,0         ;CHECK FOR FALSE LOAD
00342 103430         ANDZ# 0,0        ;CARRY WHEN INSTRUCTION
00343 103403         AND 0,0,SNC      ;CONTAINS A BIT 12 (#=NO LOAD).
00344 063077         HALT             ;AND GATE (ALC,ALC SKIP,IR12)

00345 020042 A55:    LDA 0,KB15       ;BIT 15 SHOULD SET CARRY.
00346 103623         ANDZR 0,0,SNC    ;CHECK ENAB CARRY LEVEL
00347 063077         HALT             ;AND GATE (ADDER 15,SHR)

00350 020061 A56:    LDA 0,KB0        ;BIT 0 SHOULD SET CARRY.
00351 103523         ANDZL 0,0,SNC    ;CHECK ENAB CARRY LEVEL
00352 063077         HALT             ;AND GATE (ADDER 0,SHL)

00353 020041 A57:    LDA 0,K0         ;BIT 0 SHOULD NOT SET CARRY.
00354 103522         ANDZL 0,0,SZC    ;CHECK ENAB CARRY LEVEL
00355 063077         HALT             ;AND GATE (ADDER 0,SHL).

00356 020041 A58:    LDA 0,K0         ;BIT 15 SHOULD NOT SET CARRY.
00357 103622         ANDZR 0,0,SZC    ;CHECK ENAB CARRY LEVEL
00360 063077         HALT             ;AND GATE (ADDER 15,SHR)

00361 020041 A59:    LDA 0,K0         ;CHECK ENAB CARRY LEVEL.
00362 103642         ANDOR 0,0,SZC    ;AND GATE (SHL,SHR,CARRY SET)
00363 063077         HALT             ;THE SHR INPUT.

00364 020041 A60:    LDA 0,K0         ;CHECK ENAB CARRY LEVEL.
00365 103542         ANDOL 0,0,SZC    ;AND GATE (SHL,SHR,CARRY SET)
00366 063077         HALT             ;THE SHL INPUT.

00367 020041 A61:    LDA 0,K0         ;CHECK ALC SKIP ENAB LEVEL.
00370 103621         ANDZR 0,0,SZR    ;AND OF (CARRY SET,SHR).
00371 063077         HALT

00372 020041 A62:    LDA 0,K0         ;CHECK ALC SKIP ENAB LEVEL
00373 103521         ANDZL 0,0,SZR    ;AND OF (CARRY SET,SHL).
00374 063077         HALT
```

```
00375 020041 A63:   LDA 0,K0          ;CHECK ALC SKIP ENAB LEVEL
00376 103545        ANDOL 0,0,SNR     ;AND OF (CARRY SET,SHL)
00377 063077        HALT

00400 020041 A64:   LDA 0,K0          ;CHECK ALC SKIP ENAB LEVEL
00401 103645        ANDOR 0,0,SNR     ;AND OF (CARRY SET,SHR).
00402 063077        HALT

00403 020042 A65:   LDA 0,KB15        ;BIT 15 SHOULD BE SHIFTED.
00404 103624        ANDZR 0,0,SZR     ;SHR FAIL TO INHIBIT ADDER=15
00405 063077        HALT              ;FROM ADDER=0 LEVEL.

00406 020061 A66:   LDA 0,KB0         ;BIT 0 SHOULD BE SHIFTED.
00407 103524        ANDZL 0,0,SZR     ;SHL FAIL TO INHIBIT ADDER=0
00410 063077        HALT              ;FROM ADDER=0 LEVEL.

00411 020041 A67:   LDA 0,K0          ;CHECK CG=CR SIGNELS
00412 101023        MOVZ 0,0,SNC      ;TO CRY OUT LOGIC.
00413 101043        MOVO 0,0,SNC
00414 063077        HALT

00415 102000 A68:   ADC 0,0           ;CHECK CG=CR SIGNELS
00416 101023        MOVZ 0,0,SNC      ;TO CRY OUT LOGIC.
00417 101043        MOVO 0,0,SNC
00420 063077        HALT

00421 020041 A69:   LDA 0,K0          ;BIT 12 OF THE ADC INST
00422 102010        ADC# 0,0          ;DID NOT PREVENT THE
00423 101004        MOV 0,0,SZR       ;STORE OF AC.
00424 063077        HALT

00425 020041 A70:   LDA 0,K0          ;CHECK ACD0=ACC0.
00426 024062        LDA 1,M1          ;LOADING AC1 CHANGED
00427 101004        MOV 0,0,SZR       ;C(AC0)
00430 063077        HALT

00431 020041 A71:   LDA 0,K0          ;CHECK IR DECODE OF ACD
00432 024062        LDA 1,M1          ;LOADING OF AC 1=2=3
00433 030062        LDA 2,M1          ;CHANGED CONTENTS OF
00434 034062        LDA 3,M1          ;AC0.
00435 101004        MOV 0,0,SZR
00436 063077        HALT

00437 024041 A72:   LDA 1,K0          ;LOADING OF AC 2=3=0
00440 030062        LDA 2,M1          ;CHANGED THE CONTENTS
00441 034062        LDA 3,M1          ;OF AC1. CHECK ACD
00442 020062        LDA 0,M1          ;ACS DECODE.
00443 125004        MOV 1,1,SZR
00444 063077        HALT
```

```
00445 030041 A73:    LDA 2,K0          ;LOADING OF AC 3-0-1
00446 034062         LDA 3,M1          ;CHANGED THE CONTENTS
00447 020062         LDA 0,M1          ;OF AC2.
00450 024062         LDA 1,M1
00451 151004         MOV 2,2,SZR
00452 063077         HALT

00453 034041 A74:    LDA 3,K0          ;LOADING OF AC 0-1-2
00454 020062         LDA 0,M1          ;CHANGED THE CONTENTS
00455 024062         LDA 1,M1          ;OF AC3.
00456 030062         LDA 2,M1
00457 175004         MOV 3,3,SZR
00460 063077         HALT

00461 020062 A75:    LDA 0,M1          ;ACD0 FAIL TO ASSERT
00462 024041         LDA 1,K0
00463 030041         LDA 2,K0
00464 034041         LDA 3,K0
00465 101005         MOV 0,0,SNR
00466 063077         HALT

00467 020041 A76:    LDA 0,K0          ;ACD OR ACS1 FAIL TO
00470 024062         LDA 1,M1          ;ASSERT.
00471 125005         MOV 1,1,SNR
00472 063077         HALT

00473 024041 A77:    LDA 1,K0          ;ACD2 OR ACS2 FAIL TO
00474 030062         LDA 2,M1          ;ASSERT.
00475 151005         MOV 2,2,SNR
00476 063077         HALT

00477 030041 A78:    LDA 2,K0          ;ACD3 OR ACS3 FAIL TO
00500 034062         LDA 3,M1          ;ASSERT.
00501 175005         MOV 3,3,SNR
00502 063077         HALT
00503 034041         LDA 3,K0

00504 034041 A79:    LDA 3,K0          ;AC LOAD ENABLE FAILED
00505 176010         ADC# 3,3          ;TO PREVENT LOADING OF
00506 175004         MOV 3,3,SZR       ;AC3. CHECK AC3 ENAB
00507 063077         HALT              ;(A42) BOTTEM AND GATE.

00510 030041 A80:    LDA 2,K0          ;AC LOAD ENABLE FAILED
00511 152010         ADC# 2,2          ;TO PREVENT LOADING OF
00512 151004         MOV 2,2,SZR       ;AC2. CHECK AND GATE
00513 063077         HALT              ;(ACD2+ACC2,AC LOAD ENAB)

00514 024041 A81:    LDA 1,K0          ;AC LOAD ENABLE FAILED
00515 126010         ADC# 1,1          ;TO PREVENT LOADING OF
00516 125004         MOV 1,1,SZR       ;AC1. CHECK AND GATE
00517 063077         HALT              ;(ACD1+ACC1,AC LOAD ENAB)
```

```
00520 020041 A82:   LDA 0,K0          ;INC AC0 FROM +0 TO +1.
00521 101405        INC 0,0,SNR       ;CHECK ADD ONE LOGIC.
00522 063077        HALT              ;AND GATE (IR7,ALC)

00523 020041 A83:   LDA 0,K0          ;COMP AC0 FROM +0 TO -1
00524 100005        COM 0,0,SNR       ;CHECK ACS COMP ENAB LEVEL.
00525 063077        HALT              ;AND OF (ALC,IR6)

00526 020041 A84:   LDA 0,K0          ;MISC SKIP TEST FOR SKIP
00527 101027        MOVZ 0,0,SBN      ;IF EITHER AC OR CARRY IS
00530 101026        MOVZ 0,0,SEZ      ;ZERO, OR BOTH NOT ZERO.
00531 063077        HALT

00532 020041 A85:   LDA 0,K0
00533 101047        MOVO 0,0,SBN
00534 101046        MOVO 0,0,SEZ
00535 063077        HALT

00536 020062 A86:   LDA 0,M1
00537 101027        MOVZ 0,0,SBN
00540 101026        MOVZ 0,0,SEZ
00541 063077        HALT

00542 020062 A87:   LDA 0,M1          ;ALC SKIP FUNCTIONS
00543 101046        MOVO 0,0,SEZ      ;ARE NOW FULLY TESTED.
00544 101047        MOVO 0,0,SBN
00545 063077        HALT

00546 020041 A88:   LDA 0,K0          ;TRY TO COMP ALL ONES
00547 024062        LDA 1,M1          ;TO ALL ZEROS. CHECK
00550 120004        COM 1,0,SZR       ;ADDER INPUTS FROM AC1,
00551 063077        HALT              ;ADDER ETC.

00552 020062 A89:   LDA 0,M1          ;PERHAPS ACD OUT ASSERTED
00553 124004        COM 1,1,SZR       ;WITHOUT BIT 5 IN IR.
00554 063077        HALT              ;SEE PREVIOUS TEST.

00555 030062 A90:   LDA 2,M1          ;TRY TO COMP ALL ONES
00556 140004        COM 2,0,SZR       ;TO ALL ZEROS. CHECK
00557 063077        HALT              ;ADDER INPUTS FROM AC2.

00560 034062 A91:   LDA 3,M1          ;TRY TO COMP ALL ONES
00561 160004        COM 3,0,SZR       ;TO ALL ZEROS. CHECK
00562 063077        HALT              ;ADDER INPUTS FROM AC3.

00563 020041 A92:   LDA 0,K0          ;ZERO NEGATED IS STILL ZERO!
00564 100404        NEG 0,0,SZR       ;CHECK ADDER INPUTS
00565 063077        HALT              ;FROM GATE SELECTORS.
```

```
00566 024041 A93:    LDA 1,K0          ;SEE PREVIOUS TEST.
00567 124404         NEG 1,1,SZR
00570 063077         HALT

00571 030041 A94:    LDA 2,K0          ;SEE PREVIOUS TEST.
00572 150404         NEG 2,2,SZR
00573 063077         HALT

00574 034041 A95:    LDA 3,K0          ;SEE PREVIOUS TEST.
00575 174404         NEG 3,3,SZR
00576 063077         HALT

00577 020041 A96:    LDA 0,K0          ;CHECK THE GATES FEEDING
00600 101120         MOVZL 0,0         ;THE AC BUS. THE SHIFT
00601 101004         MOV 0,0,SZR       ;LEFT INPUT HAS NOT BEEN
00602 063077         HALT              ;USED BEFORE.

00603 020041 A97:    LDA 0,K0          ;CHECK THE GATES FEEDING
00604 101220         MOVZR 0,0         ;THE AC BUS. THE SHIFT
00605 101004         MOV 0,0,SZR       ;RIGHT INPUT HAS NOT BEEN
00606 063077         HALT              ;USED BEFOR.

00607 020041 A98:    LDA 0,K0          ;CHECK THE GATES FEEDING
00610 101300         MOVS 0,0          ;THE AC BUS. THE SWAP
00611 101004         MOV 0,0,SZR       ;INPUT WAS NOT BEEN USED
00612 063077         HALT              ;BEFOR.

00613 034062 A99:    LDA 3,M1          ;CHECK THE GATES FEEDING
00614 175300         MOVS 3,3          ;THE AC BUS.
00615 160004         COM 3,0,SZR       ;TEST FOR ALL ONES BY
00616 063077         HALT              ;USING COM TO ALL ZEROS.

                     .EOT
```

```
00617 030062 B00:   LDA 2,M1          ;SHIFT RIGHT INPUT TO
00620 151240        MOVOR 2,2         ;AC BUS FOR ANY BIT
00621 140004        COM 2,0,SZR       ;ZERO IN C(2).
00622 063077        HALT

00623 024062 B01:   LDA 1,M1          ;SHIFT LEFT INPUT TO
00624 125140        MOVOL 1,1         ;AC BUS FOR ANY BIT
00625 134004        COM 1,3,SZR       ;ZERO IN C(1).
00626 063077        HALT

00627 020041 B02:   LDA 0,K0          ;0+0=0
00630 103004        ADD 0,0,SZR
00631 063077        HALT

00632 030041 B03:   LDA 2,K0          ;0-0=0
00633 152404        SUB 2,2,SZR
00634 063077        HALT

00635 024062 B04:   LDA 1,M1          ;(-1)+(-1)=-2
00636 127005        ADD 1,1,SNR
00637 063077        HALT

00640 020041 B05:   LDA 0,K0          ;0+(-1)=-1
00641 024062        LDA 1,M1
00642 107005        ADD 0,1,SNR
00643 063077        HALT

00644 020062 B06:   LDA 0,M1          ;(-1)-(-1)=0
00645 102404        SUB 0,0,SZR
00646 063077        HALT

00647 020041 B07:   LDA 0,KR0
00650 103004        ADD 0,0,SZR
00651 063077        HALT

00652 024060 B08:   LDA 1,MB1         ;CHECK CARRY GENARATE
00653 127005        ADD 1,1,SNR       ;BIT 1
00654 063077        HALT
00655 126404        SUB 1,1,SZR       ;CARRY STOP BIT 0
00656 063077        HALT

00657 030057 B09:   LDA 2,MB2         ;CHACK CARRY GENERATE
00660 153005        ADD 2,2,SNR       ;BIT 2
00661 063077        HALT
00662 152404        SUB 2,2,SZR       ;CARRY STOP BIT 1
00663 063077        HALT
```

```
00664 034056 B10:   LDA 3,KB3       ;CHECK CARRY GENERATE
00665 177005        ADD 3,3,SNR     ;BIT 3
00666 063077        HALT
00667 176404        SUB 3,3,SZR     ;CARRY STOP BIT 2
00670 063077        HALT

00671 020055 B11:   LDA 0,KB4       ;CHECK CARRY GENERATE
00672 103005        ADD 0,0,SNR
00673 063077        HALT
00674 102404        SUB 0,0,SZR     ;CARRX STOP BIT 3
00675 063077        HALT

00676 024054 B12:   LDA 1,KB5       ;CHECK CARRY GENERATE
00677 127005        ADD 1,1,SNR
00700 063077        HALT
00701 126404        SUB 1,1,SZR     ;CARRY STOP BIT 4
00702 063077        HALT

00703 030053 B13:   LDA 2,KB6       ;CHECK CARRY GENERATE
00704 153005        ADD 2,2,SNR     ;BIT 6
00705 063077        HALT
00706 152404        SUB 2,2,SZR     ;CARRY STOP BIT 5
00707 063077        HALT

00710 034052 B14:   LDA 3,KB7       ;CHECK CARRY GENERATE
00711 177005        ADD 3,3,SNR     ;BIT 7
00712 063077        HALT
00713 176404        SUB 3,3,SZR     ;CARRY STOP BIT 6
00714 063077        HALT

00715 020051 B15:   LDA 0,KB8       ;CHECK CARRY GENERATE
00716 103005        ADD 0,0,SNR     ;BIT 8
00717 063077        HALT
00720 102404        SUB 0,0,SZR     ;CARRY STOP BIT 7
00721 063077        HALT

00722 024050 B16:   LDA 1,KB9       ;CHECK CARRY GENERATE
00723 127005        ADD 1,1,SNR     ;BIT 9
00724 063077        HALT
00725 126404        SUB 1,1,SZR     ;CARRY STOP BIT 8
00726 063077        HALT

00727 030047 B17:   LDA 2,KB10      ;CHECK CARRY GENERATE
00730 153005        ADD 2,2,SNR     ;BIT 10
00731 063077        HALT
00732 152404        SUB 2,2,SZR     ;CARRY STOP BIT 9
00733 063077        HALT

00734 034046 B18:   LDA 3,KB11      ;CHECK CARRY GENERATE
00735 177005        ADD 3,3,SNR     ;BIT 11
00736 063077        HALT
00737 176404        SUB 3,3,SZR
00740 063077        HALT            ;CARRY STOP BIT 10
```

```
00741 020045 B19:    LDA 0,KB12        ;CHECK CARRY GENERATE
00742 103005         ADD 0,0,SNR       ;BIT 12
00743 063077         HALT
00744 102404         SUB 0,0,SZR       ;CARRY STOP BIT 11
00745 063077         HALT

00746 024044 B20:    LDA 1,KB13        ;CHECK CARRY GENERATE
00747 127005         ADD 1,1,SNR       ;BIT 13.
00750 063077         HALT
00751 126404         SUB 1,1,SZR       ;CHECK CARRY STOP BIT 12
00752 063077         HALT

00753 030043 B21:    LDA 2,KB14        ;CHECK CARRY GENERATE
00754 153005         ADD 2,2,SNR       ;BIT 14
00755 063077         HALT
00756 152404         SUB 2,2,SZR
00757 063077         HALT              ;CARRY STOP BIT 13

00760 034042 B22:    LDA 3,KB15        ;CHECK CARRY GENERATE
00761 177005         ADD 3,3,SNR       ;BIT 15
00762 063077         HALT
00763 176404         SUB 3,3,SZR       ;CARRY STOP BIT 14
00764 063077         HALT

00765 034041 B23:    LDA 3,K0          ;CHECK THE ADD OF
00766 176000         ADC 3,3 ;ALL ONES TO ALL
00767 160004         COM 3,0,SZR       ;ZEROS.
00770 063077         HALT

00771 030062 B24:    LDA 2,M1          ;CHECK ATE ADD OF
00772 152000         ADC 2,2,          ;ALL ZEROS TO ALL
00773 140004         COM 2,0,SZR       ;ONES.
00774 063077         HALT

00775 024062 B25:    LDA 1,M1          ;CHECK ADD OF ALL ONES
00776 127000         ADD 1,1           ;TO ALL ONES. C(1)=RESULT
00777 121400         INC 1,0           ;OF ADD, C(0)=RESULT OF INC,
01000 114004         COM 0,3,SZR       ;C(3)=FINAL RESULT.
01001 063077         HALT

01002 020041 B26:    LDA 0,K0          ;ZERO NEGATED SHOULD PRODUCE
01003 100442         NEGO 0,0,SZC      ;CR CARRY SIGNEL TO
01004 063077         HALT              ;CARRY SET LOGIC. CHECK
                                       ;BOTTEM AND GATE TO CARRY SET,
                                       ;CR SIGNEL.
```

```
01005 020041 B27:   LDA 0,K0          ;ZEROS NEGATED SHOULD SET
01006 100423        NEGZ 0,0,SNC      ;CARRY. CHECK AND GATE TO PRODUE
01007 063077        HALT              ;CRY OUT AND ITS INPUT TO
                                      ;CARRY SET LOGIC.

01010 020041 B28:   LDA 0,K0          ;CHECK FOR FALSE CG-CR SIGNEL
01011 102022        ADCZ 0,0,SZC      ;TO CARRY SET LOGIC.
01012 063077        HALT

01013 020062 B29:   LDA 0,M1          ;SEE ABOVE.
01014 102022        ADCZ 0,0,SZC
01015 063077        HALT

01016 020041 B30:   LDA 0,K0          ;SEE ABOVE.
01017 102043        ADCO 0,0,SNC
01020 063077        HALT

01021 020062 B31:   LDA 0,M1          ;SEE ABOVE.
01022 102043        ADCO 0,0,SNC
01023 063077        HALT

01024 024061 B32:   LDA 1,KB0         ;CHECK CR CARRY SIGNEL
01025 127042        ADDO 1,1,SZC      ;FROM ADDER TO CARRY SET
01026 063077        HALT              ;LOGIC.

01027 024061 B33:   LDA 1,KB0         ;CHECK CG SIGNEL INTO
01030 127023        ADDZ 1,1,SNC      ;CRY OUT,CARRY SET LOGIC.
01031 063077        HALT

01032 020063 B34:   LDA 0,K125252     ;CHECK SELX-SELY AND
01033 101140        MOVOL 0,0         ;GATE INPUTS TO AC BUS.
01034 024064        LDA 1,KS2525
01035 106414        SUB# 0,1,SZR
01036 063077        HALT              ;LEFT SHIFT FAIL

01037 020064 B35:   LDA 0,KS2525      ;AS ABOVE. CHECK GATE
01040 101120        MOVZL 0,0         ;LEFT INPUTS TO AC BUS.
01041 024063        LDA 1,K125252
01042 106414        SUB# 0,1,SZR
01043 063077        HALT

01044 020064 B36:   LDA 0,KS2525      ;CHECK SELX-SELY AND
01045 101240        MOVOR 0,0         ;GATE INPUTS TO AC BUS.
01046 024063        LDA 1,K125252
01047 106414        SUB# 0,1,SZR
01050 063077        HALT              ;RIGHT SHIFT FAIL
```

```
01051 020063 B37:   LDA 0,K125252   ;CHECK SELX-SELY AND
01052 101220        MOVZR 0,0       ;GATE INPUTS TO AC BUS.
01053 024064        LDA 1,K52525
01054 106414        SUB# 0,1,SZR
01055 063077        HALT            ;RIGHT SHIFT FAIL.

01056 020062 B38:   LDA 0,M1        ;CHECK THE SWAP INPUTS
01057 101300        MOVS 0,0        ;TO THE GATES FEEDING THE
01060 104004        COM 0,1,SZR     ;AC BUS
01061 063077        HALT

01062 020064 B39:   LDA 0,K52525    ;THIS NUMBER SWAPPED SHOULD
01063 105300        MOVS 0,1        ;REMAIN UNCHANGED.
01064 106414        SUB# 0,1,SZR
01065 063077        HALT

01066 020063 B40:   LDA 0,K125252   ;SWAPING THIS NUMBER
01067 105320        MOVZS 0,1       ;SHOULD NOT CHANGE IT.
01070 106414        SUB# 0,1,SZR
01071 063077        HALT

01072 034041 B41:   LDA 3,K0        ;0+0=0
01073 177004        ADD 3,3,SZR     ;CHECK AC3 TO ADDER
01074 063077        HALT

01075 034062 B42:   LDA 3,M1        ;CHECK AC3 TO ADDER.
01076 176000        ADC 3,3
01077 160004        COM 3,0,SZR
01100 063077        HALT

01101 030041 B43:   LDA 2,K0        ;AC2 TO ADDER
01102 153004        ADD 2,2,SZR
01103 063077        HALT

01104 024041 B44:   LDA 1,K0        ;AC1  TO ADDER
01105 127004        ADD 1,1,SZR
01106 063077        HALT

01107 020041 B45:   LDA 0,K0        ;AC0 TO ADDER
01110 103004        ADD 0,0,SZR
01111 063077        HALT

01112 020062 B46:   LDA 0,M1        ;AC0 TO ADDER
01113 102000        ADC 0,0
01114 104004        COM 0,1,SZR
01115 063077        HALT

01116 024062 B47:   LDA 1,M1        ;AC1 TO ADDER
01117 126000        ADC 1,1
01120 120004        COM 1,0,SZR
01121 063077        HALT
```

```
01122 030062 B48:    LDA 2,M1          ;AC2 TO ADDER
01123 152000         ADC 2,2
01124 140004         COM 2,0,SZR
01125 063077         HALT

01126 020042 B49:    LDA 0,KB15        ;(+1)+(-1)=0
01127 024062         LDA 1,M1          ;CHECK FOR AND ASSERTING
01130 107004         ADD 0,1,SZR       ;IF RESULT IS +1,
01131 063077         HALT              ;(IR5,IR6,IR7) GATE.

01132 020042 B50:    LDA 0,KB15        ;CHECK FOR NOT AND ASSERT.
01133 102404         SUB 0,0,SZR
01134 063077         HALT

01135 020042 B51:    LDA 0,KB15        ;CHECK FOR NOT AND ASSERT.
01136 101405         INC 0,0,SNR
01137 063077         HALT

01140 020041 B52:    LDA 0,KC          ;0&0=0
01141 103404         AND 0,0,SZR
01142 063077         HALT

01143 020062 B53:    LDA 0,M1          ;AND INST DID NOT
01144 103422         ANDZ 0,0,SZC      ;SUPPRESS CARRY.
01145 063077         HALT

01146 020062 B54:    LDA 0,M1          ;AND OF ALL BITS TO
01147 103400         AND 0,0           ;ALL BITS SHOULD SET
01150 104004         COM 0,1,SZR       ;ALL BITS.
01151 063077         HALT

01152 102000 B55:    ADC 0,0           ;1&0=0
01153 126400         SUB 1,1
01154 107404         AND 0,1,SZR
01155 063077         HALT

01156 126000 B56:    ADC 1,1           ;0&1=0
01157 102400         SUB 0,0
01160 107404         AND 0,1,SZR
01161 063077         HALT
```

```
01162 020064 B57:   LDA 0,K52525    ;AND FAILED
01163 126000        ADC 1,1         ;C(0)=CORRECT
01164 107400        AND 0,1         ;C(1)=ERROR
01165 106414        SUB# 0,1,SZR
01166 063077        HALT

01167 020063 B58:   LDA 0,K125252   ;AND FAILED
01170 126000        ADC 1,1         ;C(0)=CORRECT
01171 107400        AND 0,1         ;C(1)=ERROR
01172 106414        SUB# 0,1,SZR
01173 063077        HALT

01174 010403 B59:   ISZ .+3         ;TEST TO INSURE
01175 101000        MOV 0,0         ;ISZ IS NOT A JMP
01176 101001        MOV 0,0,SKP     ;INSTRUCTION.
01177 063077        HALT

01200 152400 B60:   SUB 2,2         ;CHECK IR DECODE OF
01201 010000        ISZ 0           ;ISZ.
01202 101010        MOV# 0,0
01203 151004        MOV 2,2,SZR
01204 063077        HALT
01205 176520        SUBZL 3,3

01206 054000 B61:   STA 3,0         ;STA STORES DIFFIRENT
01207 020000        LDA 0,0         ;THINGS. PERHAPS
01210 054000        STA 3,0         ;DSZ IS ASSERTED.
01211 024000        LDA 1,0         ;CHECK AND GATE
01212 106414        SUB# 0,1,SZR    ;(IR0,IR1,IR2) ON IR
01213 063077        HALT            ;DECODING PRINT.

01214 102400 B62:   SUB 0,0         ;TRY TO STORE ZEROS
01215 040000        STA 0,0         ;IN LOCATION 0
01216 024000        LDA 1,0         ;CHECK IR DECODE (STA)
01217 125004        MOV 1,1,SZR     ;AND GATE (IR0,IR1,IR2)
01220 063077        HALT            ;ON IR=DECODING PRINT.

01221 102000 B63:   ADC 0,0         ;TRY TO STORE ALL
01222 040000        STA 0,0         ;ONES IN LOCATION 0.
01223 024000        LDA 1,0
01224 106414        SUB# 0,1,SZR
01225 063077        HALT

01226 102401 B64:   SUB 0,0,SKP     ;THIS INSTRUCTION SETS
01227 020777        LDA 0,.-1       ;ALC SKIP. ALC SKIP
01230 101004        MOV 0,0,SZR     ;SHOULD PREVENT EXEC
01231 063077        HALT            ;LEVEL FROM ASSERTING.
                                    ;SEE (STATES CPU-2 PRINT).
```

```
01232 102000 B65:    ADC 0,0          ;IR 7(1) OF LDA INSTRUCTION
01233 040000         STA 0,0          ;SHOULD MAKE IT LOAD THE
01234 020400         LDA 0,0          ;INSTRUCTION INTO AC0 NOT
01235 104005         COM 0,1,SNR      ;LOCATION 0.
01236 063077         HALT

01237 126621 R66:    SUBZR 1,1,SKP    ;IR BITS 8-9 OF THE
01240 100000         100000           ;LDA INSTRUCTION ARE SET.
01241 020777         LDA 0,.-1        ;DATA LOADED WAS SHIFTED
01242 125220         MOVZR 1,1        ;RIGHT. CHECK AND OF
01243 106415         SUB# 0,1,SNR     ;(ALC-IR8) TO PRODUCE SELX.
01244 063077         HALT

01245 126520 B67:    SUBZL 1,1        ;IR BITS 8-9 OF THE
01246 101041         MOVO 0,0,SKP     ;LDA INSTRUCTION ARE SET.
01247 000000         0                ;DATA LOADED WAS SHIFTED
01250 020777         LDA 0,.-1        ;LEFT. CHECK AND OF
01251 106415         SUB# 0,1,SNR     ;(ALC-IR9) TO PRODUCE SELY.
01252 063077         HALT

01253 102000 R68:    ADC 0,0          ;CHECK FOR FALSE
01254 103760         ANDCS 0,0        ;(ENAB ACD RIGHT MSB COMP)
01255 104014         COM# 0,1,SZR     ;LEVEL ON AND INSTRUCTION.
01256 063077         HALT

01257 176400 R69:    SUB 3,3
01260 020402         LDA 0,.+2        ;CHECK EFA FOR LDA INSTRUCTION.
01261 126621         SUBZL 1,1,SKP    ;IF ACO=0 PERHAPS NON-EXIST
01262 000001         1                ;MEMORY REFFERANCED VIA
01263 106414         SUB# 0,1,SZR     ;FALSE ENAB ACD RIGHT MSB COMP
01264 063077         HALT             ;SIGNEL. PERHAPS INDEXED ON AC-3

01265 126621 B70:    SUBZR 1,1,SKP    ;CHECK EFA FOR LDA.
01266 100000         100000           ;IF WORD IN ACO IS SAME AS THE
01267 020777         LDA 0,.-1        ;WORD AT THIS LOCATION +377 THEN
01270 106414         SUB# 0,1,SZR     ;SIGN BIT 8 WAS NOT EXTENDED
01271 063077         HALT             ;DURING THE EFFECTIVE ADDRESS TR

01272 176521 R71:    SUBZL 3,3,SKP    ;THE INSTRUCTION INDEXED ON
01273 123456         123456           ;AC3 OBTAINED THE SAME
01274 025777         LDA 1,.-1,3      ;RESULT AS THE INSTRUCTION
01275 030776         LDA 2,.-2        ;INDEXED ON PROGRAM COUNTER.
01276 132415         SUB# 1,2,SNR     ;PERHAPS IR4 TO ACSX
01277 063077         HALT             ;LEFT HI SELECT LOGIC FAILED.
```

```
01300 176000 B72:   ADC 3,3          ;LOAD LOCATION 0 VIA INDEX
01301 021401        LDA 0,1,3        ;ON AC3 AND DIRECT FROM
01302 024000        LDA 1,0          ;PAGE 0. RESULTS NOT SAME!
01303 106414        SUB# 0,1,SZR     ;CHECK ACSX LEFT HI SEL, ACSX
01304 063077        HALT             ;LEFT LOW SELECT AT EFA TIME.

01305 176400 B73:   SUB 3,3          ;LOAD LOCATION 0 VIA INDEX
01306 152000        ADC 2,2          ;ON AC2 AND DIRECT FROM
01307 021001        LDA 0,1,2        ;PAGE 0. RESULTS NOT THE SAME.
01310 024000        LDA 1,0          ;SEE ABOVE.
01311 106414        SUB# 0,1,SZR
01312 063077        HALT

01313 020400 B74:   LDA 0,.          ;INDEX ON AC2 SHOULD
01314 040000        STA 0,0          ;CAUSE ENAB ACD RIGHT MSB
01315 152520        SUBZL 2,2        ;COMP TO ASSERT. CHECK
01316 021377        LDA 0,-1,2       ;OR GATE (IR6,IR7).
01317 024000        LDA 1,0          ;(-1)+(+1) SHOULD PRODUCE LOC 0
01320 106414        SUB# 0,1,SZR     ;C(1)=LOC 0 DIRECT.
01321 063077        HALT

01322 034065 B75:   LDA 3,K377       ;DIRECT LOAD OF LOC 377
01323 020377        LDA 0,377        ;AND INDEX LOAD OF 377 NOT SAME!
01324 025400        LDA 1,0,3        ;CHECK AND GATE (EFA,IR6,
01325 106414        SUB# 0,1,SZR     ;(OR OR IR6-IR7)) TO ENAB
01326 063077        HALT             ;ACD RIGHT MSB COMP.

01327 034065 B76:   LDA 3,K377       ;THE NUMBER 377 AFTER
01330 054000        STA 3,0          ;A ISZ INSTRUCTION =0.
01331 010000        ISZ 0            ;CHECK IR DECODE.
01332 101010        MOV# 0,0         ;OR GATE (ISZ,DSZ,LDA) THE
01333 020000        LDA 0,0          ;ISZ INPUT.
01334 101005        MOV 0,0,SNR
01335 063077        HALT

01336 034065 B77:   LDA 3,K377       ;DECREMENT OF THE
01337 054000        STA 3,0          ;NUMBER 377 RESULTED IN 0.
01340 014000        DSZ 0            ;CHECK IR DECODE.
01341 101010        MOV# 0,0         ;OR GATE(ISZ,DSZ,LDA) THE
01342 020000        LDA 0,0          ;DSZ INPUT.
01343 101005        MOV 0,0,SNR
01344 063077        HALT
```

```
01345 102400 B78:   SUB 0,0          ;LOCATION 0 IS SET TO
01346 040000        STA 0,0          ;ALL ZEROS. THE ISZ INSTRUCTION
01347 010000        ISZ 0            ;SHOULD SET IT TO +1.
01350 101010        MOV# 0,0         ;THE VALUE IN LOC 0 IS STILL
01351 020000        LDA 0,0          ;ZERO HOWEVER. CHECK ADD ONE
01352 101005        MOV 0,0,SNR      ;LOGIC. AND GATE(TS0,EXEC,ISZ)
01353 063077        HALT

01354 102400 B79:   SUB 0,0          ;LOCATION 0 SHOULD BE
01355 040000        STA 0,0          ;INCREMENTED VIA THE ISZ
01356 010000        ISZ 0            ;INSTRUCTION FROM +0 TO +1.
01357 101010        MOV# 0,0         ;CHECK ISZ?
01360 020000        LDA 0,0          ;C(0)=RESULT OF ISZ
01361 126520        SUBZL 1,1        ;AC1=+1
01362 106414        SUB# 0,1,SZR
01363 063077        HALT

01364 102000 B80:   ADC 0,0          ;LOCATION 0 SHOULD BE
01365 040000        STA 0,0          ;INCREMENTED VIA THE ISZ
01366 010000        ISZ 0            ;INSTRUCTION FROM (-1) TO +0.
01367 101010        MOV# 0,0         ;CHECK ISZ?
01370 020000        LDA 0,0
01371 101004        MOV 0,0,SZR
01372 063077        HALT

01373 102020 B81:   ADCZ 0,0         ;A ISZ INSTRUCTION CHANGED
01374 040000        STA 0,0          ;THE STATE OF CARRY.
01375 010000        ISZ 0            ;CHECK AND GATE
01376 101010        MOV# 0,0         ;(ALC,ALC SKIP,IR12) THE
01377 101002        MOV 0,0,SZC      ;ALC INPUT.
01400 063077        HALT

01401 102400 B82:   SUB 0,0          ;INCREMENTING A ZERO TO +1
01402 040000        STA 0,0          ;CAUSED ISZ TO SKIP. CHECK AND
01403 010000        ISZ 0            ;(EXEC-B,ADDER=0 SAVE,PTG3,ISZ+B
01404 101001        MOV 0,0,SKP      ;TO THE INC PC FLOP. ADDER=0
01405 063077        HALT             ;SAVE SHOULD NOT BE TRUE.

01406 102000 B83:   ADC 0,0          ;INCREMENT OF (-1) TO +0 DID NOT
01407 040000        STA 0,0          ;SKIP. CHECK AND GATE (EXEC-B,
01410 010000        ISZ 0            ;ADDER=0 SAVE,PTG3,ISZ+DSZ)
01411 063077        HALT             ;TO THE INC PC FLOP.

01412 102120 B84:   ADCZL 0,0        ;ISZ SHOULD NOT SKIP THE
01413 040000        STA 0,0          ;FIRST TIME BUT SHOULD SKIP
01414 010000        ISZ 0            ;THE SECOND. CHECK (ADDER=0
01415 010000        ISZ 0            ;SAVE) ETC,ETC.
01416 063077        HALT
```

```
01417 102400 B85:    SUB 0,0          ;DSZ FAILED TO DECREMENT
01420 040000         STA 0,0          ;LOCATION 0. CHECK
01421 014000         DSZ 0            ;ACS COMP ENAB LEVEL.
01422 101010         MOV# 0,0         ;AND GATE (DSZ,TSO.EXEC)
01423 020000         LDA 0,0
01424 101005         MOV 0,0,SNR
01425 063077         HALT

01426 102400 B86:    SUB 0,0          ;DECREMENT OF 0 SHOULD
01427 040000         STA 0,0          ;PRODUCE 177777. DSZ
01430 014000         DSZ 0            ;FAILED.
01431 101010         MOV# 0,0
01432 020000         LDA 0,0
01433 104004         COM 0,1,SZR
01434 063077         HALT

01435 102400 B87:    SUB 0,0          ;DSZ SKIPED?
01436 040000         STA 0,0          ;C(LOC 0) SHOULD BE=(-1)
01437 014000         DSZ 0
01440 101001         MOV 0,0,SKP
01441 063077         HALT

01442 102520 B88:    SUBZL 0,0        ;DECREMENT OF +1
01443 040000         STA 0,0          ;SHOULD PRODUCE +0.
01444 014000         DSZ 0            ;DSZ FAILED.
01445 101000         MOV 0,0
01446 024000         LDA 1,0
01447 125004         MOV 1,1,SZR
01450 063077         HALT

01451 102520 B89:    SUBZL 0,0        ;DSZ FAIL TO SKIP
01452 040000         STA 0,0          ;DECREMENT OF +1 TO +0.
01453 014000         DSZ 0            ;CHECK IR DECODE OF
01454 063077         HALT             ;(ISZ+DSZ) LEVEL.

01455 102400 B90:    SUB 0,0          ;IF "LDA" HANGS UP (IN DEFER)
01456 040000         STA 0,0          ;CHECK STATES CPU# PRINT.
01457 022000         LDA 0,@0         ;AND GATE (DEFER,MEMO FLOP).

01460 102001 B91:    ADC 0,0,SKP      ;ALC SKIP LEVEL FAILED
01461 022000         LDA 0,@0         ;TO PREVENT DEFER CYCLE.
01462 104004         COM 0,1,SZR      ;CHECK AND GATE (EFA,IRS,
01463 063077         HALT             ;ALC SKIP).

01464 102400 B92:    SUB 0,0          ;A INDIRECT REFFERANCE TO
01465 040000         STA 0,0          ;LOCATION 0 SHOULD NOT
01466 022000         LDA 0,@0         ;CAUSE LOCATION 0 TO CHANGE.
01467 020000         LDA 0,0          ;CHECK AND GATE (MA12,
01470 101004         MOV 0,0,SZR      ;DEFER MEM MOD) TO AUTO
01471 063077         HALT             ;INC LEVEL.
```

```
01472 102400 B93:   SUB 0,0          ;A INDIRECT REFFERANCE TO
01473 040010        STA 0,10         ;LOCATION 10 SHOULD NOT
01474 022010        LDA 0,@10        ;CAUSE LOCATION 10 TO CHANGE.
01475 020010        LDA 0,10         ;CHECK AND GATE (MA12,
01476 101004        MOV 0,0,SZR      ;DEFER MEM MOD) TO AUTO
01477 063077        HALT             ;DEC LEVEL.

01500 102400 B94:   SUB 0,0          ;A REFFERANCE TO LOCATION
01501 040020        STA 0,20         ;20 WITHOUT DEFER SHOULD
01502 020020        LDA 0,20         ;NOT CAUSE AUTO INCREMENT.
01503 024020        LDA 1,20         ;CHECK AND GATE (MA ZERO BUS,
01504 107014        ADD# 0,1,SZR     ;MA11,DEFER,TS0) THE
01505 063077        HALT             ;DEFER INPUT.

01506 102400 B95:   SUB 0,0          ; A INDIRECT REFFERANCE TO
01507 040020        STA 0,20         ;LOCATION 20 SHOULD CAUSE IT
01510 022020        LDA 0,@20        ;TO INCREMENT TO +1. (CHECK ADDS
01511 020020        LDA 0,20         ;AC0=RESULT OF INCREMENT
01512 126520        SUBZL 1,1        ;AC1=+1. CHECK AUTO INC
01513 106414        SUB# 0,1,SZR     ;LEVEL. CHECK DEFER MEM MOD
01514 063077        HALT             ;LEVEL ETC. PERHAPS AUTO DEC ASC

01515 102520 R96:   SUBZL 0,0        ;A INDIRECT REFFERANCE TO
01516 040030        STA 0,30         ;LOCATION 30 SHOULD CAUSE IT
01517 022030        LDA 0,@30        ;TO DECREMENT FROM +1 TO +0.
01520 020030        LDA 0,30         ;AC0=RESULD OF DECREMENT.
01521 101004        MOV 0,0,SZR      ;CHECK AND GATE (DEFER MEM MOD,
01522 063077        HALT             ;MA12 TO AUTO DEC LEVEL.
                                     ;ALSO ACS COMP ENAB LEVEL.

01523 102520 B97:   SUBZL 0,0        ;SET C(LOC 0&LOC 1) TO +1.
01524 040000        STA 0,0          ;INDIRECT REFFERANCE TO LOC 0
01525 040001        STA 0,1          ;SHOULD LOAD AC1 WITH THE
01526 026000        LDA 1,@0         ;CONTENTS OF LOC 1, NAMELY
01527 106414        SUB# 0,1,SZR     ;A +1.
01530 063077        HALT

01531 020065 B98:   LDA 0,K377       ;LOC 0=377, LOC 1=+0.
01532 040000        STA 0,0          ;A INDIRECT REFFERANCE TO
01533 126400        SUB 1,1          ;LOCATION 1 SLOULD LOAD THE
01534 044001        STA 1,1          ;CONTENTS OF LOCATION 0.
01535 026001        LDA 1,@1
01536 106414        SUB# 0,1,SZR
01537 063077        HALT

01540 102400 B99:   SUB 0,0          ;ISZ WITH DEFER SKIPED.
01541 040000        STA 0,0          ;CHECK THE EXEC-B INPUT TO
01542 012000        ISZ @0           ;(EXEC-B,ADDER=0 SAVE,PTG3,
01543 101001        MOV 0,0,SKP      ;ISZ,DSZ) FEEDING INC PC FLOP.
01544 063077        HALT
```

```
01545 101001 C00:    MOV 0,0,SKP      ;TEST INDIRECT WITH PC
01546 000065         K377             ;INDEX
01547 022777         LDA 0,@.-1       ;C(AC0)=WORD VIA DEFER.
01550 024065         LDA 1,K377       ;C(AC1)=WORD PAGE 0 DIRECT.
01551 106414         SUB# 0,1,SZR     ;THEY SHOULD BE THE SAME!
01552 063077         HALT

01553 020065 C01:    LDA 0,K377       ;THIS PROGRAM PERFORMS A
01554 040001         STA 0,1          ;ANDIRECT REFFERANCE TO
01555 126520         SUBZL 1,1        ;LOCATION 0.
01556 044000         STA 1,0          ;IR BITS 5-6-7 ARE ALL
01557 176000         ADC 3,3          ;SET IN THE LDA INSTRUCTION.
01560 033401         LDA 2,@1,3       ;IF THE AND LEVEL ASSERTS
01561 112414         SUB# 0,2,SZR     ;NO CARRYS MAY OCCURE AND
01562 063077         HALT             ;THE WRONG LOCATION WILL BE
                                      ;REFFERANCED. CHECK GENERATION
                                      ;OF AND.

01563 102520 C02:    SUBZL 0,0        ;LOC 0=+1, LOC 1=@0
01564 040000         STA 0,0          ;TWO LEVEL DEFER. IF BIT 0
01565 126620         SUBZR 1,1        ;OF LOC 1 IS NOT DETECTED
01566 044001         STA 1,1          ;LOCATION 0 NOT 1 WILL BE
01567 036001         LDA 3,@1         ;LOADED INTO AC3. CHECK
01570 136414         SUB# 1,3,SZR     ;MEMO LEVEL TO FLOP, AND GATE
01571 063077         HALT             ;(DEFER,FLOP) ON STATES CPU2 PRE

01572 102220 C03:    ADCZR 0,0        ;LOC 0=77777
01573 040000         STA 0,0          ;LOC 1=0
01574 126400         SUB 1,1          ;LOC 2=@1
01575 044001         STA 1,1          ;LOC 3=@2
01576 125240         MOVOR 1,1        ;THREE LEVEL INDIRECT SHOULD
01577 125400         INC 1,1          ;LOAD LOCATION 0 INTO AC2.
01600 044002         STA 1,2          ;SEE CPU2 STATES PRINT.
01601 125400         INC 1,1
01602 044003         STA 1,3
01603 032003         LDA 2,03
01604 112414         SUB# 0,2,SZR
01605 063077         HALT

01606 000402 C04:    JMP .+2          ;FIRST TIME "JMP" USED.
01607 063077         HALT             ;CHECK LOAD PC(B15) AND GATE
                                      ;(C4,TS3,JMP,INST.DEFER.ALC-SKIP

01610 101001 C06:    MOV 0,0,SKP      ;LOAD PC SIGNEL WAS ASSERTED
01611 000402         JMP .+2          ;WHEN ALC SKIP WAS SET.
01612 101001         MOV 0,0,SKP      ;CHECK AND GATE (C4,JMP,TS3,
01613 063077         HALT             ;INST  .DEFER SET.ALC SKIP).
```

```
01614 101001 C07:    MOV 0,0,SKP      ;JMP WITH DEFER
01615 001621         .+4
01616 002777         JMP @.-1
01617 063077         HALT
01620 063077         HALT

01621 102000 C08:    ADC 0,0          ;JMP SHOULD NOT CHANGE
01622 114400         NEG 0,3          ;ACS.
01623 000401         JMP .+1
01624 117014         ADD# 0,3,SZR
01625 063077         HALT

01626 101001 C09:    MOV 0,0,SKP      ;FIRST TEST OF JSR
01627 002001         2001             ;SET UP LOCATION FOR
01630 020777         LDA 0,.-1        ;A RETURN TO THE HALT.
01631 040000         STA 0,0          ;PERHAPS ADDER OUT (B83)
01632 101001         MOV 0,0,SKP      ;FAILED TO ASSERT.
01633 001643         C09A             ;CHECK AND GATE
01634 020777         LDA 0,.-1        ;(EXEC SET,JSR,TS3) ETC.
01635 040001         STA 0,1
01636 101001         MOV 0,0,SKP
01637 001642         .+3
01640 034777         LDA 3,.-1
01641 004401         JSR .+1
01642 101001         MOV 0,0,SKP
01643 063077 C09A:   HALT

01644 101001 C10:    MOV 0,0,SKP      ;AC3 ENAB FAIL TO
01645 001651         .+4              ;ASSERT ON JSR. CHECK
01646 034777         LDA 3,.-1        ;AND GATE (INST.DEFER.ALC SKIP,
01647 004401         JSR .+1          ;JSR) TO AC3 ENAB.
01650 101001         MOV 0,0,SKP
01651 063077         HALT

01652 004402 C11:    JSR .+2          ;CHECK LOAD PC VIA JSR.
01653 063077         HALT             ;AND (C4,TS0,EXECB,JSR)

01654 176000 C12:    ADC 3,3          ;MA OUT FAILED TO
01655 004403         JSR .+3          ;ASSERT ON JSR. CHECK
01656 174015         COM# 3,3,SNR     ;AND GATE (JSR,EXEC,TS3).
01657 063077         HALT

01660 101001 C13:    MOV 0,0,SKP      ;PC TO MA AT FETCH TIME
01661 001663         .+2              ;OF JSR FAILED?
01662 020777         LDA 0,.-1
01663 004401         JSR .+1
01664 114415         SUB# 0,3,SNR
01665 063077         HALT
```

```
01666 004402 C14:   JSR .+2             ;AC3 AHOULD BE LOADED
01667 001667        .                   ;WITH ADDRESS C14+1.
01670 020777        LDA 0,.-1           ;JSR FAILED.
01671 116414        SUB# 0,3,SZR
01672 063077        HALT

01673 006402 C15:   JSR @.+2            ;CHECK AC3 VALUE WITH
01674 001674        .                   ;MULTI DEFERED JSR.
01675 101676        @.+1
01676 101677        @.+1
01677 001700        .+1
01700 020774        LDA 0,C15+1
01701 116414        SUB# 0,3,SZR
01702 063077        HALT

01703 176401 C16:   SUB 3,3,SKP     ;AC3 CHANGED VIA JSR
01704 004401        JSR .+1 ;WHEN ALC SKIP WAS SET?
01705 175004        MOV 3,3,SZR
01706 063077        HALT

01707 034047 C18:   LDA 3,KB10      ;CHECK THE MA=ZERO BUS.
01710 021400        LDA 0,0,3       ;AC3= THE LOCATION BEING
01711 101112        MOVL# 0,0,SZC   ;TESTED. IF THERE IS A
01712 000405        JMP .+5         ;DEFER BIT IN THE WORD
01713 027400        LDA 1,@0,3      ;THE TEST WILL NOT BE
01714 025400        LDA 1,0,3       ;PERFORMED.
01715 106414        SUB# 0,1,SZR    ;FOR THE BIT SPECIFIED IN
01716 063077        HALT            ;AC3, CHECK THE O.C. GATE
01717 175120        MOVZL 3,3       ;TO THE MA ZERO BUS.
01720 175113        MOVL# 3,3,SNC   ;RELOAD PROGRAM AFTER
01721 000767        JMP C18+1       ;MALFUNCTION IS CORRECTED.

01722 000403 C19:   JMP .+3
01723 000403        JMP .+3
01724 063077        HALT
01725 000776        JMP .-2         ;A JMP FOLLOWED BY JMP TEST.
```

```
01726 063500 D00:   SKPBZ 0          ;IF MACHINE HANGS UP ON THIS
01727 101010        MOV# 0,0         ;I/O INSTRUCTION CHECK IO PAI
                                     ;LEVEL, IO RLS FLOP, ETC.

01730 061477 D01:   INTA 0           ;IF MACHINE HANGS UP ON THIS
01731 101010        MOV# 0,0         ;I/O INSTRUCTION CHECK AND GATE
                                     ;(IOTG2,IOTG3) TO IO RLS SET
                                     ;LOGIC.

01732 061777 D02:   DIBP 0,CPU       ;A DIB INSTRUCTION SKIPED!
01733 101001        MOV 0,0,SKP      ;CHECK AND (IR5,IR6) TO ANOTHER
01734 063077        HALT             ;AND OF  (IR7(IO),IO,(IR5,IR6))@
                                     ;IO SKIP LEVEL.

01735 062777 D03:   DICP 0,CPU       ;A DIC INSTRUCTION SKIPED!
01736 101001        MOV 0,0,SKP      ;CHECK AND (IR5,IR6). SEE
01737 063077        HALT             ;ABOVE.

01740 103700 D04:   ANDS 0,0         ;AND ASSERTS IR 5,6,7. BUT
01741 101001        MOV 0,0,SKP      ;FOR THE FACT THAT IO IS NOT
01742 063077        HALT             ;TRUE A IO SKIP WOULD OCCURE.
                                     ;SEE ABOVE 2 TEST.

01743 103077 D05:   ADDC# 0,0,SBN    ;IF RUN(0) CHECK INPUT TO AND
01744 101000        MOV 0,0          ;GATE PRODUCING HALT LEVEL.
                                     ;ALL INPUTS ARE TRUE EXCEPT
                                     ;IO AND EXEC.

01745 102400 D06:   SUB 0,0          ;IF RUN(0), ALL INPUTS TO
01746 040000        STA 0,0          ;HALT LEVEL ARE TRUE EXCEPT
01747 030066        LDA 2,M77        ;AND OF (IO,EXEC) WHERE IO
01750 023077        LDA 0,@77,2      ;IS NOT TRUE.
01751 101000        MOV 0,0

01752 063577 D07:   SKPBZ CPU        ;IF RUN (0), ALL  INPUTS TO
01753 101000        MOV 0,0          ;HALT LEVEL ARE TRUE EXCEPT
01754 101000        MOV 0,0          ;IR7 IS (1), CHECK THIS 4 LEG
01755 000421        JMP D08          ;AND GATE.

01756 063077        HALT             ;THE PREVIOUS IO SKIP INST
01757 063077        HALT             ;SKIPED TO FAR. CHECK
01760 063077        HALT             ;INPUTS TO INC PC FLOP.
01761 063077        HALT
01762 063077        HALT
01763 063077        HALT
01764 063077        HALT
01765 063077        HALT
01766 063077        HALT
01767 063077        HALT
```

```
01770 063077            HALT
01771 063077            HALT
01772 063077            HALT
01773 063077            HALT
01774 063077            HALT
01775 063077            HALT

01776 102400 D08:       SUB 0,0         ;IF RUN (0), ALL INPUTS TO
01777 063000            DOC 0,0         ;HALT LEVEL ARE TRUE EXCEPT
02000 101000            MOV 0,0         ;CODE 77. CHECK THIS INPUT TO
                                        ;THE 4 INPUT AND GATE. SEE
                                        ;CPU1 INPUT OUTPUT PRINT.

02001 060077 D09:       NIO CPU         ;IF RUN(0), ALL INPUTS TO HALT
02002 101000            MOV 0,0         ;LEVEL AND GATE ARE TRUE EXCEPT
                                        ;THE AND OF (IR5,IR6). SEE
                                        ;CPU1 INPUT OUTPUT PRINT.

02003 102400 D10:       SUB 0,0         ;THE "DOC" INSTRUCTION SKIPED
02004 063300            DOCP 0,0        ;BECAUSE ALL CONDITIONS EXCEPT
02005 101001            MOV 0,0,SKP     ;IR7 WAS NOT SET. CHECK AND
02006 063077            HALT            ;(IR7(IO),IO,(IR5,IR6)) TO THE
                                        ;IO SKIP LOGIC.

02007 063700 D11:       SKPDZ 0         ;IO SKIP FAILED TO SKIP
02010 101001            MOV 0,0,SKP     ;WHEN BOTH THE ZERO AND NON
02011 000403            JMP .+3         ;ZERO CASES WERE TESTED. CHECK
02012 063600            SKPDN 0         ;IO SKIP LEVEL (A88). AND GATE
02013 063077            HALT            ;(FETCH,PTG5,IO SKIP) TO INC
02014 101000            MOV 0,0         ;PC FLOP. ALSO CHECK SELD LINE.

02015 063700 D12:       SKPDZ 0         ;IO SKIP CAUSED A SKIP OF
02016 101000            MOV 0,0         ;2 INSTRUCTIONS. THE FETCH
02017 101001            MOV 0,0,SKP     ;INPUT TO AND GATE (FETCH,PTG5,
02020 063077            HALT            ;IO SKIP) FEEDING INC PC FLOP FB

02021 063700 D13:       SKPDZ 0         ;CHECK IR9 TO I/O SKIP LOGIC.
02022 063077            HALT

02023 063600 D14:       SKPDN 0         ;CHECK IR9(0) TO IO SKIP LOGIC,
02024 101001            MOV 0,0,SKP     ;ETC. 2 INPUT OR GATE (IR9,SKIP
02025 063077            HALT            ;LEVEL) ETC, ETC.
```

```
02026 063777 D15:    SKPDZ CPU          ;CHECK THE PWR LOW LOGIC
02027 063077         HALT               ;TO THE SKIP AREA.

02030 063500 D16:    SKPBZ 0            ;CHECK SELB LINE, ITS INPUT
02031 063077         HALT               ;AND FOLLOWING GATES IN IO
                                        ;SKIP LOGIC.

02032 063400 D17:    SKPBN 0            ;SHOULD WORK?
02033 101001         MOV 0,0,SKP        ;SEE ABOVE TEST.
02034 063077         HALT

02035 062677 D18:    IORST              ;CLEAR ION. IF ION=0(PANNEL LIGE
02036 063577         SKPBZ CPU          ;CHECK AND OF (ION,CODE 77)
02037 063077         HALT               ;TO SKIP LOGIC. IF ION=1 CHECK
                                        ;THE CROSS COUPLED GATES
                                        ;WHICH MAKE UP THE ION FLOP.

02040 060277 D19:    NIOC CPU           ;ION SHOULD BE CLEAR AND
02041 063477         SKPBN CPU          ;NO IO SKIP SHOULD OCCURE.
02042 101001         MOV 0,0,SKP
02043 063077         HALT

02044 060100 D20:    NIOS 0             ;A START PULSE TO A DEVICE
02045 063577         SKPBZ CPU          ;OTHER THAN CPU SHOULD NOT
02046 063077         HALT               ;SET ION FLOP. CHECK THE "C"
                                        ;INPUT TO DECODER PRODUCING
                                        ;STRT,CLR,AND IO PLS.

02047 060377 D21:    NIOP CPU           ;A "P" PULSE TO THE CPU
02050 063577         SKPBZ CPU          ;SHOULD NOT SET THE ION FLOP.
02051 063077         HALT               ;CHECK IR8-IR9 INPUTS TO THE
                                        ;DECODER PRODUCING STRT,CLR,IO P
                                        ;CHECK DECODER OUTPUT PIN 4.

02052 060077 D22:    NIO CPU            ;SEE ABOVE.
02053 063577         SKPBZ CPU
02054 063077         HALT

02055 102520 D23:    SUBZL 0,0          ;SETTING ION CHANGED THE
02056 040001         STA 0,1            ;CONDITIONS IN THE SKIP LOGIC
02057 060177         NIOS CPU           ;WHEN TESTING THE SELB LINE.
02060 063500         SKPBZ 0            ;AND GATE (ION,CODE 77) THE
02061 063077         HALT               ;CODE 77 INPUT SHOULD NOT BE
                                        ;TRUE. CHEKK OUTPUT THIS GATE.
```

```
02062 060177 D24:   NIOS CPU        ;A CLR PULSE TO DEVICE
02063 060277        NIOC CPU        ;CPU FAILED TO CLEAR THE
02064 063577        SKPBZ CPU       ;ION FLOP.
02065 063077        HALT

02066 062677 D25:   IORST           ;RESET THE WORLD THEN
02067 102400        SUB 0,0         ;TURN ON INTERRUPT SYSTEM.
02070 060177        NIOS CPU        ;NO INTERRUPT SHOULD OCCURE.
02071 040000        STA 0,0         ;C(0) CHANGED! IE INTERRUPT OCCR
02072 020000        LDA 0,0         ;CHECK INT PEND LEVEL (B81)
02073 101004        MOV 0,0,SZR     ;CHECK PWR LOW AND INTR
02074 063077        HALT            ;LINE (B29).

02075 060177 D26:   NIOS CPU        ;IF ION LIGHT (1). CHECK
02076 063577        SKPBZ CPU       ;ION INPUT TO SKIP LOGIC
02077 101001        MOV 0,0,SKP     ;IF ION LIGHT (0) CHECK
02100 063077        HALT            ;ION FLOP,DECODER OF
                                    ;STRT , CLR,  IO PLS.

02101 060177 D27:   NIOS CPU        ;CHECK IO SKIP INVERSION
02102 063477        SKPBN CPU       ;LOGIC VIA IR9(0). INTERRUPTS
02103 063077        HALT            ;SHOULD BE ON.

02104 060177 D28:   NIOS CPU        ;SETTING ION EFFECTED
02105 063777        SKPDZ CPU       ;THE SKIP CONDITION OF
02106 063077        HALT            ;POWER LOW.  CHECK IR8
                                    ;INPUT TO IO SKIP LOGIC.

02107 060177 D29:   NIOS CPU        ;SKPDN INSTRUCTION MANAGED
02110 063677        SKPDN CPU       ;TO TURN OF THE INTERRUPT
02111 101010        MOV# 0,0        ;ON FLOP(ION). CHECK THE TOP
02112 063477        SKPBN CPU       ;LEG OF THE 3 INPUT AND GATE
02113 063077        HALT            ;FEEDING THE "D" INPUT TO
                                    ;DECODER GENERATING STRT,
                                    ;CLR, IO PLS.

02114 060177 D30:   NIOS CPU        ;A "P" PULSE  CLEARED ION
02115 060377        NIOP CPU        ;FLOP. CHECK DECODER
02116 063477        SKPBN CPU       ;PRODUCING STRT,CLR,IO PLS.
02117 063077        HALT
```

```
02120 060177 D31:    NIOS CPU          ;A "C" PULSE TO DEVICE 37
02121 060237         NIOC 37           ;CLEARED ION FLOP.
02122 063477         SKPBN CPU         ;CHECK CODE 77 LEVEL
02123 063077         HALT

02124 060177 D32:    NIOS CPU          ;A "C" PULSE TO DEVICE 57
02125 060257         NIOC 57           ;CLEARED ION FLOP.
02126 063477         SKPBN CPU         ;CHECK CODE 77 LEVEL.
02127 063077         HALT

02130 060177 D33:    NIOS CPU          ;A "C" PULSE TO DEVICE 67
02131 060267         NIOC 67           ;CLEARED ION FLOP.
02132 063477         SKPBN CPU         ;CHECK CODE 77 LEVEL
02133 063077         HALT

02134 060177 D34:    NIOS CPU          ;A "C" PULSE TO DEVICE 73
02135 060273         NIOC 73           ;CLEARED ION FLOP.
02136 063477         SKPBN CPU         ;CHECK CODE 77 LEVEL.
02137 063077         HALT

02140 060177 D35:    NIOS CPU          ;A "C" PULSE TO DEVICE 75
02141 060275         NIOC 75           ;CLEARED ION FLOP.
02142 063477         SKPBN CPU         ;CHECK CODE 77 LEVEL.
02143 063077         HALT

02144 060177 D36:    NIOS CPU          ;A "C" PULSE TO DEVICE 76
02145 060276         NIOC 76           ;CLEARED ION FLOP.
02146 063477         SKPBN CPU         ;CHECK CODE 77 LEVEL
02147 063077         HALT

02150 020061 D37:    LDA 0,KB0         ;A NIO INSTRUCTION CHANGED
02151 060877         NIOC CPU          ;THE CONTENTS OF AC0.
02152 024061         LDA 1,KB0         ;AC LOAD ENABLE SHOULD NOT
02153 106414         SUB# 0,1,SZR      ;BE ASSERTED DURING IO INST.
02154 063077         HALT              ;CHECK AND GATE (IO IN, IO TIME)

02155 020067 D38:    LDA 0,K65432      ;THE DIA/DIB INSTRUCTIONS
02156 060477         DIA 0,CPU         ;SHOULD CHANGE THE CONTENTS
02157 061400         DIB 0,0           ;OF AC0. CHECK FOR ASSERTION
02160 024067         LDA 1,K65432      ;OF IO IN LEVEL (A79). ALSO
02161 106415         SUB# 0,1,SNR      ;CHECK AND OF IO IN AND IO TIME
02162 063077         HALT              ;TO AC LOAD ENABLE LOGIC.

02163 020067 D39:    LDA 0,K65432      ;A DOB 0,CPU(MSK0) INSTRUCTION
02164 105000         MOV 0,1           ;CHANGED AC0 CONTENTS. PERHAPS
02165 062077         DOB 0,CPU         ;IT GOT DECODED AS (INTA). CHECK
02166 106414         SUB# 0,1,SZR      ;IR7(IO) TO AND GATE FEEDING
02167 063077         HALT              ;"D" INPUT OF DECODER PRODUCING
                                       ;INTA.
```

```
02170 102400 D40:    SUB 0,0          ;PERHAPS READS WAS
02171 061077         DOA 0,CPU        ;ASSERTED ON DOA. CHECK
02172 101004         MOV 0,0,SZR      ;IR7(IO) TO THE "D" INPUT OF
02173 063077         HALT             ;DECODER PRODUCING READS.

02174 102000 D41:    ADC 0,0          ;SEE ABOVE.
02175 061077         DOA 0,CPU
02176 104004         COM 0,1,SZR
02177 063077         HALT

02200 020067 D42:    LDA 0,K65432     ;DIA INSTRUCTION DID NOT
02201 105000         MOV 0,1          ;CHANGE THE CONTENTS OF AC0.
02202 060400         DIA 0,0          ;CHECK DATIA INPUT TO
02203 106415         SUB# 0,1,SNR     ;OR GATE FORMING THE IO IN
02204 063077         HALT             ;LEVEL.

02205 020067 D43:    LDA 0,K65432     ;A DIB INSTRUCTION DID NOT
02206 105000         MOV 0,1          ;CHANGE THE CONTENTS OF AC0.
02207 061400         DIB 0,0          ;CHECK DATIB INPUT TO OR
02210 106415         SUB# 0,1,SNR     ;GATE FORMING THE IO IN
02211 063077         HALT             ;LEVEL.

02212 020067 D44:    LDA 0,K65432     ;A IORST INSTRUCTION (DIC 0,CPU)
02213 105000         MOV 0,1          ;CHANGED THE CONTENTS OF AC0.
02214 062677         IORST            ;CHECK THE A-B-C INPUTS TO THE
02215 106414         SUB# 0,1,SZR     ;DECODER FORMING IORST LEVEL.
02216 063077         HALT             ;A PULSE SHOULD BE OBSERVED ON
                                      ;(ATO) IF DIC 0,CPU IS DECODED
                                      ;PROPERLY.

02217 020067 D45:    LDA 0,K65432     ;A DIC INSTRUCTION FAILED TO
02220 105000         MOV 0,1          ;CHANGE THE CONTENTS OF AC0.
02221 062400         DIC 0,0          ;CHECK THE "C" INPUT TO THE
02222 106415         SUB# 0,1,SNR     ;DECODER PRODUCING DATIC.
02223 063077         HALT             ;PERHAPS IORST IS PRODUCED NOT
                                      ;DATIC. ALSO THE DATIC INPUT
                                      ;TO OR GATE FORMING THE IO IN LE

02224 020067 D46:    LDA 0,K65432     ;READS (DIA 0,CPU) DID NOT CHANE
02225 105000         MOV 0,1,         ;THE CONTENTS OF AC0. CHECK THE
02226 060477         READS 0          ;DECODE OF READS AND
02227 106415         SUB# 0,1,SNR     ;ITS INPUT TO OR GATE FORMING
02230 063077         HALT             ;THE IO IN LEVEL.
```

```
02231 020067 D47:   LDA 0,K65432    ;INTA (DIB 0,CPU) DID NOT
02232 105000        MOV 0,1         ;CHANGE THE CONTENTS OF
02233 061477        INTA 0          ;AC0. CHECK DECODE OF INTA
02234 106415        SUB# 0,1,SNR    ;AND ITS INPUT TO OR GATE
02235 063077        HALT            ;FORMING THE IO IN LEVEL.

02236 020067 D48:   LDA 0,K65432    ;A DATA OUT INSTRUCTION
02237 105000        MOV 0,1         ;SHOULD NOT CHANGE THE
02240 061000        DOA 0,0
02241 062077        DOB 0,CPU
02242 063076        DOC 0,76
02243 106414        SUB# 0,1,SZR    ;VALUE OF AC0!
02244 063077        HALT

02245 020067 D49:   LDA 0,K65432    ;INC FAILED CHECK FOR
02246 105400        INC 0,1         ;FALSE IO IN LEVEL.
02247 106014        ADC# 0,1,SZR
02250 063077        HALT

02251 102400 D50:   SUB 0,0         ;SEND ZEROS TO TESTER
02252 061000        DOA 0,0         ;OR TO FREE SPACE. READING
02253 060400        DIA 0,0         ;BACK THE BUS SHOULD PRODUCE
02254 101004        MOV 0,0,SZR     ;ZEROS. CHECK IO BUS.
02255 063077        HALT

02256 060277 D51:   NIOC CPU        ;THE ALC SKIP LEVEL
02257 101001        MOV 0,0,SKP     ;SHOULD PREVENT THE SKIP
02260 063577        SKPBZ CPU       ;FROM THE SKPBZ INSTRUCTION
02261 101001        MOV 0,0,SKP     ;FOR NO IO LEVEL(B6).
02262 063077        HALT            ;AND GATE (PROTECT, CODE 0),
                                    ;ALC SKIP, POT IO INST).

02263 102400 D52:   SUB 0,0         ;THE (LDA 0,@177,3) HAS IR
02264 040000        STA 0,0         ;BITS 5-6-7 SET. IF THE
02265 034070        LDA 3,M177      ;IO LEVEL (B6) IS ASSERTED
02266 023577        LDA 0,@177,3    ;THE INSTRUCTION WILL SKIP.
02267 101001        MOV 0,0,SKP     ;CHECK AND (IR0,IR1,IR2,INST)
02270 063077        HALT            ;TO PRODUCE POT IO INST LEVEL.

02271 176400 D53:   SUB 3,3         ;IF (STA 0,@177,3) SKIPS SEE
02272 054000        STA 3,0         ;ABOVE. IR1 SHOULD PREVENT
02273 034070        LDA 3,M177      ;IO LEVEL FROM ASSERTING.
02274 043577        STA 0,@177,3
02275 101001        MOV 0,0,SKP
02276 063077        HALT
```

```
02277 102000 D54:    ADC 0,0          ;SOME HIGH ORDER BITS
02300 061477         INTA 0           ;WERE READ BACK ON INTA.
02301 024071         LDA 1,K77        ;PERHAPS ACD OUT WAS ASSERTED.
02302 122032         ADCZ# 1,0,SZC
02303 063077         HALT

02304 010072 XXX:    ISZ PASS         ;END OF A PROGRAM PASS
02305 101000         MOV 0,0
02306 000100         JMP A1

                .END
```