4001

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

PROGRAM

Logic Test

TAPES

Binary: 095-000005

ABSTRACT

This routine is the NOVA central processor diagnostic.

097-000001-00

```
;                       LOGIC TEST

;1.     ABSTRACT
;               LOGIC TEST IA A MAINTENANCE PROGRAM DESIGNED
;               TO TEST THE NOVA PROCESSOR. IT IS A GATE BY
;               GATE CHECK OF THE PROCESSOR LOGIC. THE
;               PROGRAM DOES NOT ASSUME ANY I/O EQUIPMENT
;               AND DOES NOT THEREFOR TEST MOST FEATURES
;               OF THE PROCESSOR I/O LOGIC.

;2.     MACHINE REQUIREMENTS
;2.1            STANDRED NOVA PROCESSOR
;2.2            1K READ/WRITE MEMORY

;3.     SWITCH SETTINGS
;               STARTING ADDRESS=000040

;4.     OPERATING PROCEEDURE
;4.1            LOAD THE PROGRAM VIA THE BINARY LOADER
;4.2            SET THE SWITCHES TO 000040
;4.3            PRESS START
;4.4            THE PROGRAM SHOULD EXECUTE THE HALT AL LOC
;               EXAMINE AC3. IT SHOULD CONTAIN 40(THE SWITCHS)
;4.6            PRESS CONTINUE
;4.7            THE PROGRAM SHOULD RUN.OTHER HALTS ARE ERRORS.

;5.     PROGRAM OUTPUT/ERROR DISCRIPTION
;5.1            THE ERROR INDICATION OF THIS PROGRAM IS
;               THE HALT INSTRUCTION. THERE IS ONLY ONE
;               LEGAL HALT AT LOCATION 42. THIS HALT IS
;               EXECUTED ONCE WHEN THE PROGRAM IS STARTED.
;               THIS IS THE ONLY TEST OF THE HALT INSTRUCTION.
;5.2            WHEN A ERROR HALT OCCURS RECORD THE STATE
;               OF THE MACHINE WITH PENCIL AND PAPER.
;5.3            CONSULT THE LISTING TO DETERMINE THE CAUSE
;               OF THE ERROR.
;5.4            CONSTRUCT A LOOP WHICH WILL REPRODUCE THE
;               THE ERROR.
;5.5            SCOPE THE FAILURE.

;6.     PROGRAM DISCRIPTION/THEORY OF OPERATION
;               THIS PROGRAM IS A COLLECTION OF SMALL RCUTINES
;               EACH DESIGNED TO TEST A PORTION OF THE
;               PROCESSOR LOGIC. EACH ROUTINE IS DESIGNED TO
;               TEST AS SMALL A PART OF THE LOGIC AS POSSIBLE.
;               EACH TEST IN THE SEQUENCE IS BASED ON
;               PREVIOUS TEST WORKING.

;7.     RESTRICTIONS / MISC
;               THE TIME FOR ONE COMPETE PASS IS MEASURED
;               IN MILLISECONDS.
```

```
;TEST READ SWITCHES OF RANDOM SETTING.TEST HALT
;INSTRUCTION
;IF C(3)=-1 SWITCHS NOT STORED,CHECK WRITE ENABLE
;INPUTS (E79 2-3). IF C(3)=-1 THE ADC INSTRUCTION
;MAY HAVE SKIPPED ALSO. IF C(3)=0 CS ENABLE INPUTS
;(E74-1,A88) REGISTER CONTROL PRINT ETC,ETC.
;IF C(3)=OTHER CHECK ADDER ENABLE LEVELS,ETC.
;IF PC DOES NOT ADVANCE CHECK ONE ENABLE (A87)
;ETC,ETC.
;IF MACHINE FAILS TO HALT,READ SWITHCES SKIPPED,
;I/O INSTRUCTION FAILED TO STOP IT,ETC,ETC.

      000000          .LOC 0

00000 063077          HALT
00001 063077          HALT
00002 063077          HALT
00003 000002          JMP .-1

      000040          .LOC 40

00040 176040 A00:     ADCO 3,3          ;C(3)=-1
00041 074477          READS 3           ;C(3)=SWITCHES
00042 063077          HALT              ;EXAMINE C(3) MANUALLY!

00043 102421 A02:     SUBZ 0,0,SKP      ;TEST SKIP ABILITY. SEE
00044 063077          HALT              ;SKIP LEVEL SHOULD BE HIGH
                                        ;AT ADDER,TS4 OF (SUB)
                                        ;INSTRUCTION.

00045 102420 A04:     SUBZ 0,0          ;C(0)=0. A ZERO RESULT AND
00046 101001          MOV 0,0,SKP       ;NO SKIP FUNCTION SHOULD NOT
00047 063077          HALT              ;SKIP. "SUBZ" DID SKIP!
                                        ;CHECK SKIP LOGIC AT E79
                                        ;1-3-13-11-12. ALSO E53
                                        ;E60-9 (A78).

00050 101020 A10:     MOVZ 0,0          ;THIS IS THE FIRST TIME CARRY
00051 101001          MOV 0,0,SKP       ;IS 0 AND MOVZ SKIPPED.
00052 063077          HALT              ;CHECK SKIP LOGIC E53-5.
                                        ;MB14 NOT BEING SET SHOULD
                                        ;PREVENT THE SKIP

00053 101100 A12:     MOVL 0,0          ;MOVL CONTAINS A BIT 9,IT
00054 101001          MOV 0,0,SKP       ;ALSO SKIPPED. PERHAPS THE
00055 063077          HALT              ;SKIP LOGIC WAS INVERTED BY
                                        ;THE I/O SKIP LEVEL FAILING.
                                        ;CHECK E92 5-6 AND
                                        ;E41 9-10
```

```
00056 020061 A13:    LDA 0,.+3
00057 101000         MOV 0,0
00060 101001         MOV 0,0,SKP       ;THE LDA INSTRUCTION
00061 063077         HALT              ;THINKS ITS A "JMP".

00062 040065 A131:   STA 0,.+3
00063 101000         MOV 0,0
00064 101001         MOV 0,0,SKP       ;THE STA INSTRUCTION
00065 063077         HALT              ;THINKS ITS A JMP.

00066 040072 A132:   STA 0,.+4
00067 010072         ISZ .+3
00070 101000         MOV 0,0
00071 101001         MOV 0,0,SKP       ;THE ISZ INSTRUCTION
00072 063077         HALT              ;THINKS ITS A "JMP".

00073 040077 A133:   STA 0,.+4
00074 014077         DSZ .+3
00075 101000         MOV 0,0
00076 101001         MOV 0,0,SKP       ;THE DSZ INSTRUCTION
00077 063077         HALT              ;THINKS ITS A "JMP".

00100 020107 A134:   LDA 0,C0          ;A HALT INDICATES THAT
00101 101001         MOV 0,0,SKP       ;SKIPPED. CHECK SKIP LOGIC
00102 063077         HALT              ;E80-10. OR PC ENABLE GOT
                                       ;TURNED ON AT TS4 OF THE
                                       ;FETCH CYCLE

00103 040000 A08:    STA 0,0           ;C(0)=0 STORE IN REG 0
00104 101001         MOV 0,0,SKP       ;A HALT INDICATES "STA"
00105 063077         HALT              ;SKIPPED. CHECK SKIP
                                       ;LOGIC E80-11
                                       ;NOT STA SHOULD PREVENT
                                       ;THE SKIP
00106 101001         MOV 0,0,SKP
00107 000000 C0:     0                 ;A CONSTANT


00110 102000 A14:    ADC 0,0           ;BOTH C(0) AND C(1) ARE NOT
00111 024230         LDA 1,C7700       ;ZERO. CHECK THE ONES REMEMBER
00112 107045         ADDO 0,1,SNR      ;FLOP IN THE SKIP LOGIC.
00113 063077         HALT              ;E68,E67 1-13-8,E78 10-12
                                       ;-13-9-8 AND E53 2-3

00114 020232 A16:    LDA 0,CM1         ;C(0)=-1
00115 101025         MOVZ 0,0,SNR      ;IF C(0) WAS -1 AFTER LDA
00116 063077         HALT              ;AND IS NOW 0,MAB ENABLE
                                       ;LEVEL FAILED.CHECK E52 1-3
                                       ;AND E54 1-2-12-13 AND
                                       ;ACS ENABLE (A86) SEE INST REG
```

```
---
***

00117 024232 A18:    LDA 1,CM1          ;C(1)=-1
00120 125025         MOVZ 1,1,SNR       ;IF C(1) WAS -1 AFTER LDA
00121 063077         HALT               ;AND IS NOW 0,CHECK ACS1 ON
                                        ;THE INST REGISTER+AC PRINT.
                                        ;IF C(1) NOT LOADED BY LDA
                                        ;CHECK ACD1

00122 030232 A20:    LDA 2,CM1          ;C(2)=-1
00123 151025         MOVZ 2,2,SNR       ;IF C(2) WAS -1 AFTER THE LDA
00124 063077         HALT               ;AND IS NOW 0,CHECK ACS2 ON
                                        ;THE INSTRUCTION REGISTER.
                                        ;CHECK ACS2,ADC2 ON THE AC
                                        ;PRINT

00125 034232 A22:    LDA 3,CM1          ;C(3)=-1
00126 175025         MOVZ 3,3,SNR       ;SAME AS ABOVE EXCEPT
00127 063077         HALT               ;AC 3 USED.


00130 020132 A24:    LDA 0,.+2          ;TRY TO LOAD C(0)FROM
00131 101011         MOV# 0,0,SKP       ;CONSTANT. A NUMBER
00132 000000         0                  ;OF THINGS COULD FAIL
00133 101024         MOVZ 0,0,SZR       ;BUT LOOK AT ACD
00134 063077         HALT               ;AT ACD AROUND THE
                                        ;INSTRUCTION REGISTER

00135 020137 A26:    LDA 0,.+2          ;CHECK THAT ACD IS NOT
00136 126521         SUBZL 1,1,SKP      ;ENABLED ON A MOV INST.
00137 177777         177777             ;IF C(1)=0 CHECK AC ENABLE
00140 105025         MOVZ 0,1,SNR       ;LOGIC. E73 2-12-1-8
00141 063077         HALT

00142 024144 A27:    LDA 1,.+2          ;C(1) SHOULD BE SET TO 0
00143 101011         MOV# 0,0,SKP
00144 000000         0                  ;CONSTANT
00145 125024         MOVZ 1,1,SZR       ;CHECK ACD1,ETC,ETC
00146 063077         HALT

00147 131000 A28:    MOV 1,2            ;C(1) SHOULD GO TO C(2)
00150 151024         MOVZ 2,2,SZR       ;C(1)=0
00151 063077         HALT               ;CHECK ACD2,ETC,ETC
```

```
---
↑↑↑

00152 155000 A30:    MOV 2,3          ;C(2) SHOULD BE=0
00153 175024         MOVZ 3,3,SZR     ;CHECK ACD2 AT THE
00154 063077         HALT             ;AC FLOPS AND IR
                                      ;REGISTER DECODE

00155 020234 A32:    LDA 0,C17        ;CHECK THE REMEMBER"1"
00156 101025         MOVZ 0,0,SNR     ;LOGIC TO THE SKIP.
00157 063077         HALT             ;IF C(0)=17 CHECK E78
                                      ;E67 8-9-10

00160 020236 A34:    LDA 0,C2         ;CHECK THE REMEMBER "1"
00161 101025         MOVZ 0,0,SNR     ;LOGIC TH THE SKIP LEVEL.
00162 063077         HALT             ;IF C(0)=2,CHECK E78 8-13

00163 020240 A36:    LDA 0,C4         ;CHECK THE REMEMBER "1"
00164 101025         MOVZ 0,0,SNR     ;LOGIC TO THE SKIP.
00165 063077         HALT             ;IF C(0)=4,CHECK E78 8-12

00166 020242 A38:    LDA 0,C10C       ;CHECK THE REMEMBER "1"
00167 101025         MOVZ 0,0,SNR     ;LOGIC TO THE SKIP LEVEL
00170 063077         HALT             ;IF C(0)=10 CHECK E78 8-10

00171 020244 A40:    LDA 0,C1         ;CHECK THE REMEMBER "1"
00172 101025         MOVZ 0,0,SNR     ;LOGIC TO THE SKIP .
00173 063077         HALT             ;IF C(0)=1 CHECK E78 8-9
```

```
00174 020246 A42:     LDA 0,AF00        ;CONSTANT 0,TEST ALC
00175 100025          COMZ 0,0,SNR      ;BIT 6(0),USE COMP OF ACS.
00176 063077          HALT              ;CHECK NOT (MA ENABLE) AT
                                        ;A83. E50 6-5-8-9-10.
                                        ;ALSO E69 1-2

00177 020220 A44:     LDA 0,C167356     ;NOT MA ENABLE GATES THE
00200 126400          SUB 1,1           ;ZEROS INTO THE ADDER.ONLY
00201 104025          COMZ 0,1,SNR      ;ZERO PRESENT IS MA15
00202 063077          HALT              ;IF C(1)=0 E20 2-3 FAIL

00203 020222 A46:     LDA 0,C156735     ;NOT MA ENABLE GATES THE
00204 126400          SUB 1,1           ;ZEROS INTO THE ADDER.ONLY
00205 104025          COMZ 0,1,SNR      ;ZERO PRESENT IS MA14
00206 063077          HALT              ;IF C(1)=0 E8 2-3 FAIL

00207 020224 A48:     LDA 0,C135673     ;NOT MA ENABLE GATES THE
00210 126400          SUB 1,1           ;ZEROS INTO THE ADDER.ONLY
00211 104025          COMZ 0,1,SNR      ;ZEROS PRESENT IS MA13
00212 063077          HALT              ;IF C(1)=0 E46 2-3
                                        ;FAILED

00213 020226 A50:     LDA 0,C073567     ;NOT MA ENABLE GATES THE
00214 126400          SUB 1,1           ;ZEROS INTO THE ADDER.ONLY
00215 104025          COMZ 0,1,SNR      ;ZERO PRESENT IS MA12
00216 063077          HALT              ;IF C(1)=0 E33 2-3 FAIL

00217 101011          MOV# 0,0,SKP
00220 167356 C167356:167356
00221 101011          MOV# 0,0,SKP
00222 156735 C156735:156735
00223 101011          MOV# 0,0,SKP
00224 135673 C135673:135673
00225 101011          MOV# 0,0,SKP
00226 073567 C073567:073567
00227 101001          MOV 0,0,SKP
00230 007700 C7700:   7700
00231 101001          MOV 0,0,SKP
00232 177777 CM1:     -1
00233 101001          MOV 0,0,SKP
00234 000017 C17:     17
00235 101001          MOV 0,0,SKP
00236 000002 C2:      2
00237 101001          MOV 0,0,SKP
00240 000004 C4:      4
00241 101001          MOV 0,0,SKP
00242 000010 C10C:    10
00243 101001          MOV 0,0,SKP
00244 000001 C1:      1
00245 101001          MOV 0,0,SKP
00246 000000 AF00:    0
```

```
---
↑↑↑

00247 020247 A52:   LDA 0,.          ;RANDOM BITS TO C(0)
00250 126400        SUB 1,1          ;CLEAR DESTINATION IF COMP
00251 104020        COMZ 0,1         ;A RANDOM NUMBER OF BITS
00252 125415        INC# 1,1,SNR     ;PRODUCES (-1) PERHAPS
00253 063077        HALT             ;MAB ENABLE IS ASSERTED.
                                     ;CHECK A85,E52 12-13
                                     ;E50 1-2-3

00254 020256 A54:   LDA 0,.+2        ;THE COM INST GATES MA ZEROS
00255 126401        SUB 1,1,SKP      ;TO ADDER. AT TS2 MA SHOULD
00256 177777        177777           ;CONTAIN NO ZEROS.SUGGEST
00257 104024        COMZ 0,1,SZR     ;MA0 INPUT TO ADDER IS ENABLED
00260 063077        HALT             ;IF C(1)=010421 MA15
                                     ;IF C(1)=021042 MA14
                                     ;IF C(1)=042104 MA13
                                     ;IF C(1)=104210 MA12

00261 020263 A56:   LDA 0,.+2        ;C(0)=-1
00262 126001        ADC 1,1,SKP      ;PREVIOUSLY COM HAS COMP
00263 177777        177777           ;ALL ONES TO ALL ZEROS. THE
00264 104024        COMZ 0,1,SZR     ;DESTINATION AC WAS ZERO
                                     ;HOWEVER
00265 063077        HALT             ;PERHAPS ACD WAS ENABLED?
                                     ;CHECK E71 2-12-11-8 IF
                                     ;C(1)=-2,CHECK MISC NOISE
                                     ;AROUND THE ADDER

00266 020270 A58:   LDA 0,.+2        ;C(0)=0
00267 101421        INCZ 0,0,SKP     ;INC SHOULD PRODUCE +1
00270 000000        0                ;IF C(0)=0
00271 101015        MOV# 0,0,SNR     ;CHECK ONE ENABLE LOGIC
00272 063077        HALT             ;AT E51 8-9-10-11-5

00273 020275 A60:   LDA 0,.+2        ;INCREMENTION THE VALUE 17
00274 101421        INCZ 0,0,SKP     ;SHOULD PRODUCE 20 IF THE
00275 000017        17               ;CARRY SAVE LOGIC WORKS.
00276 101005        MOV 0,0,SNR
00277 063077        HALT



00300 020302 A62:   LDA 0,.+2        ;CHECK IF INC
00301 101421        INCZ 0,0,SKP     ;CAN PRODUCE A CARRY
00302 177777        177777           ;DOWN 16 BITS
00303 101004        MOV 0,0,SZR      ;(-1)+(+1)=0
00304 063077        HALT

00305 020307 A64:   LDA 0,.+2        ;NEGATIOG A ZERO WILL
00306 100401        NEG 0,0,SKP      ;PRODUCE 0. IE
00307 000000        0                ;0 TO 177777 +1=0
00310 101004        MOV 0,0,SZR
00311 063077        HALT
```

```
00312 020314 A66:    LDA 0,.+2          ;SUBTRACTING 0 FROM 0
00313 102401         SUB 0,0,SKP        ;SHOULD PRODUCE 0
00314 000000         0                  ;IF NOT SUBTRACT FAIL?
00315 101004         MOV 0,0,SZR
00316 063077         HALT

00317 020321 A68:    LDA 0,.+2          ;CHECK ABILITY OF ADC
00320 102001         ADC 0,0,SKP        ;TO SET ALL ONES
00321 000000         0                  ;DONT KNOW WHY THIS FAILED.
00322 100014         COM# 0,0,SZR       ;ALL SHOULD WORK BY
00323 063077         HALT               ;THIS TIME.

00324 020326 A70:    LDA 0,.+2          ;ANOTHER TEST OF ADC
00325 102001         ADC 0,0,SKP
00326 177777         177777             ;IF C(0)=0,ACD WAS NOT
00327 100014         COM# 0,0,SZR       ;USED. CHECK  E71
                                        ;1-2-12-13-11-8
00330 063077         HALT               ;THE AC ENABLE FAILED.


00331 020333 A72:    LDA 0,.+2          ;TEST THAT THE
00332 105001         MOV 0,1,SKP        ;DESTINATION AC IS USED
00333 000001         1                  ;BY SUB.
00334 106404         SUB 0,1,SZR        ;IF C(1)=(-1) THEN
00335 063077         HALT               ;AND ENABLE PREVENTED
                                        ;AC ENABLE. CHECK E70 8-10

00336 102400 A74:    SUB 0,0            ;C(0)=0
00337 103004         ADD 0,0,SZR        ;0+0 SHOULD = 0
00340 063077         HALT               ;?CARRY LEVEL?

00341 102400 A76:    SUB 0,0            ;ADDITION OF 0 AND -1
00342 126000         ADC 1,1            ;SHOULD PRODUCE NON ZERO
00343 107005         ADD 0,1,SNR        ;SUGGEST AND ENABLE
00344 063077         HALT               ;PREVENTED AC ENABLE
                                        ;CHECK E70 12-8

00345 102000 A80:    ADC 0,0            ;INC A (-1) TO
00346 126400         SUB 1,1            ;PRODUCE 0
00347 105424         INCZ 0,1,SZR       ;IF C(1)=+1
00350 063077         HALT               ;AND ENABLE ASSERTED
                                        ;CHECK E70 8-9

00351 102400 A82:    SUB 0,0            ;C(0)=0
00352 103404         AND 0,0,SZR        ;0 AND 0 SHOULD = 0
00353 063077         HALT               ;CHECK THE ADDER INPUTS
                                        ;FOR THE AND CONDITION

00354 102000 A84:    ADC 0,0            ;CHECK AND OF (-1) TO (-1)
00355 103405         AND 0,0,SNR        ;SHOULD GET (-1) RESULT
00356 063077         HALT               ;BUT GOT 0
                                        ;PERHAPS ONE ENABLE ASSERTED
                                        ;CHECK E51 5-6-8-10
```

```
00357 102400 A86:    SUB 0,0          ;AND 0 TO -1
00360 126000         ADC 1,1          ;SHOULD PRODUCE 0
00361 107400         AND 0,1          ;IT PRODUCED (-1) HOWEVER
00362 124015         COM# 1,1,SNR     ;PERHAPS AC ENABLE
00363 063077         HALT             ;ASSERTED. CHECK
                                      ;E71 12-13

00364 126000 A88:    ADC 1,1          ;AND (-1) TO 0
00365 102400         SUB 0,0          ;SHOULD PRODUCE 0
00366 123400         AND 1,0          ;IT PRODUCED (-1) HOWEVER
00367 100015         COM# 0,0,SNR     ;PERHAPS MAB ENABLE
00370 063077         HALT             ;ASSERTED. CHECK
                                      ;E52 1-12

00371 102000 A90:    ADC 0,0          ;(-1) AND (-1)
00372 103400         AND 0,0          ;SHOULD PRODUCE (-1)
00373 100014         COM# 0,0,SZR     ;A BIT DROPED IS IN ERROR
00374 063077         HALT             ;CHECK ADDER INPUT
                                      ;GATES FOR AND

00375 102400 A92:    SUB 0,0          ;AND (0) TO (-1)
00376 126000         ADC 1,1          ;SHOULD PRODUCE 0
00377 107404         AND 0,1,SZR      ;CHECK ADDER GATES FOR AND
00400 063077         HALT

00401 102000 A94:    ADC 0,0          ;AND (-1) TO 0
00402 126400         SUB 1,1          ;SHOULD PRODUCE 0
00403 107404         AND 0,1,SZR      ;CHECK ADDER GATES
00404 063077         HALT             ;FOR AND


00405 101040 A96:    MOVO 0,0         ;SET CARRY THEN CLEAR
00406 101022         MOVZ 0,0,SZC     ;IF C(CARRY)=0 CHECK
00407 063077         HALT             ;E53 4-5-6 TO SKIP LOGIC
                                      ;IF CARRY=1 CHECK CARRY
                                      ;SAVE LOGIC

00410 101020 A98:    MOVZ 0,0         ;CLEAR CARRY AND THEN SET
00411 101043         MOVO 0,0,SNC     ;IF CARRY=1,CHECK E53-4
00412 063077         HALT             ;IF CARRY=0.CHECK CARRY
                                      ;SAVE LOGIC

00413 101040 A99:    MOVO 0,0         ;SET CARRY ,SEE IF IT
00414 101003         MOV 0,0,SNC      ;WAS REMEMBERED
00415 063077         HALT             ;CHECK E80,E28,E42


                     .EOT
```

```
00416 101020 B00:    MOVZ 0,0          ;CLEAR CARRY THEN CHECK
00417 101002         MOV 0,0,SZC       ;THAT MOV DOES NOT SET IT.
00420 063077         HALT              ;IF C(CARRY)=1 CHECK
                                       ;E44 1-6-8-9
                                       ;OR E28 CARRY REMEMBER

00421 101020 B02:    MOVZ 0,0          ;CLEAR CARRY THEN CHECK
00422 020060         LDA 0,60          ;THAT LDA DOES NOT SET IT
00423 101002         MOV 0,0,SZC       ;CHECK E44 4-6
00424 063077         HALT

00425 101020 B04:    MOVZ 0,0          ;ADDER PRODUCES CARRY
00426 102423         SUBZ 0,0,SNC      ;CHECK CARRY E28 2-3
00427 063077         HALT              ;E42 5-6

00430 101040 B06:    MOVO 0,0
00431 102423         SUBZ 0,0,SNC
00432 063077         HALT              ;IF CARRY=0
                                       ;CHECK E44 -2
                                       ;ALSO THE CARRY FLOP

00433 101040 B08:    MOVO 0,0          ;CHECK E44 5-6
00434 102463         SUBC 0,0,SNC      ;INPUT TO CARRY LEVEL
00435 063077         HALT              ;CAN IT REMEMBER A "1"
                                       ;IN THE CARRY FLOP

00436 102000 B10:    ADC 0,0           ;CHECK FOR
00437 040000         STA 0,0           ;NO CHANGE IN
00440 101040         MOVO 0,0          ;CARRY FROM A
00441 010000         ISZ 0             ;ISZ INSTRUCTION
00442 101010         MOV# 0,0          ;CHECK E44 8-3
00443 101003         MOV 0,0,SNC       ;FOR OPEN ETCH OR
00444 063077         HALT              ;FAIL CHIP


00445 101020 B12:    MOVZ 0,0          ;CHECK CARRY REMEMBER
00446 101050         MOVO# 0,0         ;LOGIC MB12(0) (#) SHOULD
00447 101002         MOV 0,0,SZC       ;PREVENT CARRY FROM CHANGE
00450 063077         HALT              ;CHECK E28 1-8-13,E80 2-12

00451 101040 B14:    MOVO 0,0          ;CHECK CARRY REMEMBER
00452 101050         MOVO# 0,0         ;LOGIC E28 13 IS HIGH
00453 101003         MOV 0,0,SNC       ;OR CHIP FAIL
00454 063077         HALT

00455 101020 B16:    MOVZ 0,0          ;CHECK CARRY REMEMBER
00456 101030         MOVZ# 0,0         ;LOGIC E28 9-10
00457 101002         MOV 0,0,SZC       ;E42 8-9
00460 063077         HALT
```

```
00461 101040 B18:    MOVO 0,0          ;CHECK CARRY REMEMBER
00462 101030         MOVZ# 0,0         ;LOGIC E28 9-10
00463 101003         MOV 0,0,SNC       ;E42 8-9
00464 063077         HALT

00465 101020 B20:    MOVZ 0,0          ;CAN IT REMEMBER
00466 101000         MOV 0,0           ;AT ALL ABOUT CARRY?
00467 101002         MOV 0,0,SZC
00470 063077         HALT

00471 101040 B22:    MOVO 0,0          ;CHECK THAT
00472 020000         LDA 0,0           ;CARRY CHANGES
00473 101003         MOV 0,0,SNC       ;ONLY AT TS3 TIME
00474 063077         HALT              ;E80


00475 102022 B24:    ADCZ 0,0,SZC      ;JUST TRYING SOME STUF
00476 063077         HALT
00477 102063         ADCC 0,0,SNC
00500 063077         HALT
00501 102032         ADCZ# 0,0,SZC
00502 063077         HALT
00503 102032         ADCZ# 0,0,SZC
00504 063077         HALT
00505 102053         ADCO# 0,0,SNC
00506 063077         HALT
00507 102053         ADCO# 0,0,SNC
00510 063077         HALT
00511 102062         ADCC 0,0,SZC
00512 063077         HALT


00513 102400 B26:    SUB 0,0           ;TEST FOR MB12(1)
00514 102010         ADC# 0,0          ;PREVENTING AC WRITE
00515 101004         MOV 0,0,SZR       ;ENABLE (B48),E77 9-10
00516 063077         HALT

00517 020400 B28:    LDA 0,.           ;CHECK THAT
00520 040000         STA 0,0           ;AC WRITE ENABLE
00521 152400         SUB 2,2           ;IS NOT TURNED ON
00522 010000         ISZ 0,0           ;DURING A IS INSTRUCTION
00523 101000         MOV 0,0           ;E77 5-8-10
00524 151004         MOV 2,2,SZR
00525 063077         HALT
```

```
↑↑↑

00526 102000 B30:   ADC 0,0
00527 040000        STA 0,0
00530 102400        SUB 0,0          ;STA FAILED
00531 040000        STA 0,0          ;IT ASSERTED MEM
00532 020000        LDA 0,0          ;ENABLE
00533 101004        MOV 0,0,SZR      ;CHECK E68 8-9-10
00534 063077        HALT             ;E70 5-6

00535 102000 B32:   ADC 0,0          ;IF STA FAILED
00536 040000        STA 0,0          ;TO ASSERT AC ENABLE
00537 024000        LDA 1,0          ;IT WOULD STORE ZEROS
00540 125005        MOV 1,1,SNR      ;CHECK E71 8-9,E56 4-5-6
00541 063077        HALT

00542 102400 B34:   SUB 0,0          ;C(0)=0
00543 102300        ADCS 0,0         ;SHOULD SET ONES AND
00544 101005        MOV 0,0,SNR      ;STORE SWAP RESULT
00545 063077        HALT             ;E43-4 FAIL TO
                                     ;ASSERT NOSH+SWAP ENABLE

00546 102724 B36:   SUBZS 0,0,SZR    ;IF C(0)=+1 OR 100000
00547 063077        HALT             ;MAR OR MAL WAS ENABLED
                                     ;IN ERROR. CHECK
                                     ;DECODER AT E54

00550 020402 B38:   LDA 0,.+2        ;CHECK TO SEE IF
00551 105301        MOVS 0,1,SKP     ;DATA WAS ACTUALLY
00552 000377        377              ;SWAPPED. IF C(1)=C(0)
00553 106415        SUB# 0,1,SNR     ;E89-1 DECODER FOR
00554 063077        HALT             ;ACS FAILED


00555 102400 B40:   SUB 0,0          ;C(0)=0. AC STORE SWAP
00556 102300        ADCS 0,0         ;STORE USING THE OTHER
00557 100014        COM# 0,0,SZR     ;GATES OF THE SEQUENCER.
00560 063077        HALT             ;CHECK THE SECTION
                                     ;OF E90 WHOS BITS ARE
                                     ;MISSING

00561 102440 B42:   SUBO 0,0         ;C(0) AND C(CARRY)=0
00562 101202        MOVR 0,0,SZC     ;CARRY SAVE AND CARRY
00563 063077        HALT             ;GOT SET. CHECK E29 2-3-8
                                     ;C(0) SHOULD STILL = 0
                                     ;CHECK ADDER INPUTS ETC
                                     ;IF C(0) NOT 0,LOOK THERE FIRST

00564 102440 B44:   SUBO 0,0         ;C(0) AND C(CARRY)=0
00565 101102        MOVL 0,0,SZC     ;CARRY SAVE AND CARRY GOT
00566 063077        HALT             ;SET. CHECK E29 4-5-8
                                     ;C(0) SHOULD =0 CHECK
                                     ;ADDER INPUTS ETC.IF C(0) NOT 0
                                     ;LOOK INTO THAT FIRST
```

```
00567 102223 B46:    ADCZR 0,0,SNC     ;CHECK THE SET TO CARRY
00570 063077         HALT              ;VIA MA8(1) AND MAR ENABLE
                                       ;CHECK E29 2-3-8.C(0) SHOULD
                                       ;=(-1). IF NOT CHECK THAT FIRST
                                       ;ADDER GATES ETC

00571 102123 B48:    ADCZL 0,0,SNC     ;CHECK THE SET TO CARRY
00572 063077         HALT              ;VIA MA12(1) AND MAL ENABLE
                                       ;CHECK E29 4-5-6

00573 102420 B50:    SUBZ 0,0          ;C(0)=0,C(CARRY)=1
00574 101105         MOVL 0,0,SNR      ;CARRY SHOULD MOVE TO
00575 063077         HALT              ;BIT 15 PRODUCING +1
                                       ;IN C(0). CHECK CARRY
                                       ;SAVE INPUT TO ADDER
                                       ;E20 9-10 ETC,ETC

00576 102440 B52:    SUBO 0,0          ;C(0)=0,C(CARRY)=0
00577 101100         MOVL 0,0          ;A ZERO CARRY ON
00600 104400         NEG 0,1           ;MOVL WAS INSERTED
00601 124005         COM 1,1,SNR       ;AS A ONE
00602 063077         HALT              ;CHECK ADDER INPUT
                                       ;AT E20 9-10


00603 102440 B54:    SUBO 0,0          ;C(0)=0,C(CARRY)=0
00604 101104         MOVL 0,0,SZR      ;ZEROS MOVED LEFT
00605 063077         HALT              ;SHOULD PRODUCE ZEROS

00606 020402 B56:    LDA 0,.+2         ;C(0)=1
00607 101121         MOVZL 0,0,SKP     ;C(0) SHOULD=2 NOW
00610 000001         1
00611 101015         MOV# 0,0,SNR      ;ADDER INPUT 2 FAIL
00612 063077         HALT              ;CHECK E8 9-10

00613 020402 B58:    LDA 0,.+2         ;C(0)=2
00614 101121         MOVZL 0,0,SKP     ;C(0) SHOULD=4 NOW
00615 000002         2
00616 101015         MOV# 0,0,SNR      ;ADDER INPUT 1 FAIL
00617 063077         HALT              ;CHECK E46 2-3

00620 020402 B60:    LDA 0,.+2
00621 101121         MOVZL 0,0,SKP     ;C(0)=4
00622 000004         4                 ;SHOULD NOW=10
00623 101015         MOV# 0,0,SNR      ;ADDER INPUT 0 FAIL
00624 063077         HALT              ;CHECK E33 9-10

00625 102040 B62:    ADCO 0,0          ;C(0)=(-1),C(CARRY)=1
00626 101100         MOVL 0,0          ;MOVING ALL ONES
00627 100014         COM# 0,0,SZR      ;SHOULD LEAVE ALL ONES
00630 063077         HALT              ;DON'T KNOW WHY THIS
                                       ;SHOOLD FAIL?

00631 102245 B64:    ADCOR 0,0,SNR     ;MAR ENABLE FAIL
00632 063077         HALT              ;TO ASSERT. CHECK
                                       ;E63 12-13
```

```
---
***

00633 102520 B66:    SUBZL 0,0          ;C(0)=+1 RIGHT SHIFT
00634 105203         MOVR 0,1,SNC       ;FAIL TO PUT BIT 15
00635 063077         HALT               ;INTO CARRY. CHECK
                                        ;E56 1-13 INPUT TO NA
                                        ;ON RIGHT SHIFT, ETC, ETC

00636 102642 B68:    SUBOR 0,0,SEC      ;RIGHT SHIFT A 0 INTO
00637 063077         HALT               ;CARRY POSITION.

00640 102625 B70:    SUBZR 0,0,SNR      ;SET CARRY THEN MOVE
00641 063077         HALT               ;IT INTO BIT 0

00642 102640 B72:    SUBOR 0,0          ;CLEAR C(0) AND C(CARRY)
00643 105124         MOVZL 0,1,SZR      ;GET THE SIGN BIT OUT
00644 063077         HALT               ;SUBOR FAILED. CHECK
                                        ;ADDER GATES

00645 102644 B74:    SUBOR 0,0,SZR      ;C(CARRY)=0 AT TS2 TIME
00646 063077         HALT               ;CARRY GOT INSERTED
                                        ;AS A 1. CHECK E56 2-3

00647 020402 B76:    LDA 0,.+2          ;SHIFT A 2 RIGHT
00650 101221         MOVZR 0,0,SKP
00651 000002         2
00652 105005         MOV 0,1,SNR        ;ADDER INPUT FAIL
00653 063077         HALT
                                        ;CHECK E19 12-1

00654 020402 B78:    LDA 0,.+2          ;SHIFT A 4 RIGHT
00655 101221         MOVZR 0,0,SKP
00656 000004         4
00657 105005         MOV 0,1,SNR        ;ADDER INPUT FAIL
00660 063077         HALT               ;CHECK E7 12-1


00661 020402 B80:    LDA 0,.+2          ;SHIFT A 10 RIGHT
00662 101221         MOVZR 0,0,SKP
00663 000010         10
00664 105005         MOV 0,1,SNR        ;ADDER INPUT FAIL
00665 063077         HALT               ;CHECK E45 13-1

00666 020402 B82:    LDA 0,.+2          ;SHIFT A 20 RIGHT
00667 101221         MOVZR 0,0,SKP
00670 000020         20
00671 105005         MOV 0,1,SNR        ;ADDER INPUT FAIL
00672 063077         HALT               ;CHECK E32 13-1


00673 020402 B84:    LDA 0,.+2          ;C(0)=ORIGIONAL NUMBER
00674 105221         MOVZR 0,1,SKP      ;C(1)=RIGHT SHIFT RESULT
00675 012345         12345              ;THE NUMBER
00676 131100         MOVL 1,2           ;C(2)=LEFT SHIFT RESULT
00677 112414         SUB# 0,2,SZR       ;EXAMINE AC TO SEE WHICH
00700 063077         HALT               ;INSTRUCTION FAILED
```

```
000701 062777 C00:    DICP 0,CPU        ;IO RESET AND A
000702 101011         MOV# 0,0,SKP      ;PULSE,CAUSED A SKIP
000703 063077         HALT              ;CHECK INST REGISTER
                                        ;DECODE I/O SKIP
                                        ;SHOULD NOT BE PRESENT

000704 060277 C02:    NIOC CPU          ;CLEAR ION FLOP,MISC
000705 101011         MOV# 0,0,SKP      ;REASONS.
000706 063077         HALT              ;IT SKIPPED!
                                        ;CHECK E41 9-10,E92
                                        ;TO THE SKIP LOGIC

000707 063677 C03:    SKPDN CPU         ;SKIP POWER FAIL,ASSUMING
000710 101001         MOV 0,0,SKP       ;POWER FAIL WORKS (ASD)
000711 063077         HALT              ;CHECK E81 5+6,AND
                                        ;E67 4-5-6 TO SKIP LOGIC
                                        ;ALSO SKIP INVERT LOGIC
                                        ;E41 8-9-10

000712 063777 C04:    SKPBZ CPU         ;SKIP POWER FAIL=0
000713 063077         HALT              ;SEE PREVIOUS TEST.
                                        ;CHECK E41 8-9-10
                                        ;NBS(1) FAILED TO INVERT
                                        ;SKIP LOGIC. E92 5-6,E53 1-13

000714 060277 C06:    NIOC CPU          ;CLEAR ION FLAG,IF ION LIGHT
000715 063577         SKPBZ CPU         ;IS OUT: CHECK I/O SKIP LOGIC
000716 063077         HALT              ;AT E53 8-1-13,E67 4-5-6
                                        ;E51 11-12-13,E81 5-6

000717 060277 C08:    NIOC CPU          ;CLEAR ION FLAG. IF
000720 060100         NIOS 0            ;ION IS SET BY PULSING
000721 063577         SKPBZ CPU         ;DEVICE 0 THEN CHECK
000722 063077         HALT              ;THE CPU INST LEVEL AT
                                        ;(A11),E15 3-4 ALSO THE ION
                                        ;FLOP E76-E79,GATES E27
                                        ;ETC,ETC

000723 062677 C10:    DICC 0,CPU        ;CLEAR I/O DEVICES,ION
000724 060176         NIOS 76           ;A START PULSE WITH
000725 063577         SKPBZ CPU         ;DEVICE NOT CPU TURNED
000726 063077         HALT              ;ION ON,CHECK E2-4
```

```
00727 062677 C12:   DICC 0,CPU      ;CLEAR I/O DEVICES, ION
00730 060175        NIOS 75         ;A START PULSE WITH
00731 063577        SKPBZ CPU       ;DEVICE NOT CPU TURNED
00732 063077        HALT            ;ION ON. CHECK E2-3

00733 062677 C14:   DICC 0,CPU      ;CLEAR I/O DEVICES, ION
00734 060173        NIOS 73         ;A START PULSE WITH
00735 063577        SKPBZ CPU       ;DEVICE NOT CPU TURNED
00736 063077        HALT            ;ION ON. CHECK E2-2

00737 062677 C16:   DICC 0,CPU      ;CLEAR I/O DEVICES, ION
00740 060167        NIOS 67         ;A START PULSE WITH
00741 063577        SKPBZ CPU       ;DEVICE NOT CPU TURNED
00742 063077        HALT            ;ION ON. CHECK E2-1

00743 062677 C18:   DICC 0,CPU      ;CLEAR I/O DEVICES, ION
00744 060157        NIOS 57         ;A START PULSE WITH
00745 063577        SKPBZ CPU       ;DEVICE NOT CPU TURNED
00746 063077        HALT            ;ION ON. CHECK E2-13

00747 062677 C20:   DICC 0,CPU      ;CLEAR I/O DEVICES, ION
00750 060137        NIOS 37         ;A START PULSE WITH
00751 063577        SKPBZ CPU       ;DEVICE NOT CPU TURNED
00752 063077        HALT            ;ION ON. CHECK E2-12

00753 102000 C22:   ADC 0,0         ;C(0)=-1
00754 063477        SKPBN CPU       ;SEE IF SKPBN WILL
00755 101010        MOV# 0,0        ;MODIVY C(0)
00756 100014        COM# 0,0,SZR    ;CHECK I/O INPUT LEVEL
00757 063077        HALT            ;(A13),E66-6


00760 102000 C24:   ADC 0,0         ;SEE IF DIC WILL
00761 062477        DIC 0,CPU       ;MODIFY C(0)
00762 100014        COM# 0,0,SZR    ;CHECK  E64 1-3
00763 063077        HALT            ;THE INPUT TO THE
                                    ;LEVEL IO INPUT

00764 102000 C26:   ADC 0,0         ;CHECK THAT DIC WILL
00765 062400        DIC 0,0         ;WRITE INTO C(0)
00766 100015        COM# 0,0,SNR    ;CHECK E64 1-2-3,
00767 063077        HALT            ;E66 5-6

00770 102000 C28:   ADC 0,0         ;C(0)=-1
00771 063400        SKPBN 0         ;CHECK THAT SKPBN DEVICE 0
00772 101010        MOV# 0,0        ;DOES NOT MODIFY C(0)
00773 100014        COM# 0,0,SZR    ;CHECK E64 2-3
00774 063077        HALT

00775 102000 C30:   ADC 0,0         ;C(0)=-1,SEE IF
00776 061477        DIB 0,CPU       ;INTERRUPT ACKNOWLEGE
00777 100015        COM# 0,0,SNR    ;WILL MODIFY THE C(0)
01000 063077        HALT            ;CHECK DTIB INPUT TO
                                    ;I/O INPUT LEVEL. E66 3-6
```

```
---
†††
   01001 102000 C32:    ADC 0,0          ;BECAUSE DEVICE 0
   01002 061400         DIB 0,0          ;DOES NOT EXIST,THEREFOR
   01003 101004         MOV 0,0,SZR      ;DATA LINES SHOULD
   01004 063077         HALT             ;CONTAIN ZEROS. CHECK
                                         ;I/O DATA LINES TO MA

   01005 062677 C34:    DICC 0,CPU       ;CLEAR I/O DEVICES
   01006 061477         DIB 0,CPU        ;INTERRUPT ACKNOWLEDGE
   01007 101014         MOV# 0,0,SZR     ;WE CLEARED THE DEVICES
   01010 063077         HALT             ;YE SOMETHING WANTS A
                                         ;INTERRUPT. CHECK THE BUSS



   01011 062677 CC22:   DICC 0,CPU       ;CLEAR I/O DEVICES,ION
   01012 020406         LDA 0,C222       ;POINT TO HALT
   01013 040001         STA 0,1          ;IF INTERRUPT OCCURES
   01014 060177         NIOS CPU         ;MACH WILL HALT
   01015 101001         MOV 0,0,SKP
   01016 063077 CC221:  HALT             ;INTERRUPT OCCURED WITH
                                         ;ION COMING ON
                                         ;IT SHOULD NOT HAVE!
                                         ;CHECK INT REQ
   01017 101001         MOV 0,0,SKP
   01020 001016 C222:   CC221

   01021 102620 CC23:   SUBZR 0,0        ;IF INTERRUPTS
   01022 040001         STA 0,1          ;OCCURE POINT THEM BACK

   01023 062677 CC24:   DICC 0,CPU       ;CLEAR DEVICES AND ION
   01024 060177         NIOS CPU         ;TURN ON INTERRUPT (ION)
   01025 063777         SKPDZ CPU        ;THIS INSTRUCTION WORKED
   01026 063077         HALT             ;WHEN INTERRUPT WAS OFF.
                                         ;CHECK E81 5-6,E67 4-5

   01027 062677 CC26:   DICC 0,CPU       ;CLEAR DEVICES AND ION
   01030 060177         NIOS CPU         ;TURN ON INTERRUPT (ION)
   01031 063500         SKPBZ 0
   01032 063077         HALT             ;CHECK CPU INST INPUT TO
                                         ;SKIP LOGIC. E51 11-13

   01033 062677 CC28:   DICC 0,CPU       ;CLEAR DEVICES AND ION
   01034 060177         NIOS CPU         ;TURN ON INTERRUPT (ION)
   01035 063477         SKPBN CPU        ;IF THE ION LIGHT IS ON CHECK
   01036 063077         HALT             ;E51 11-12-13,E67 4-5-6,TO SKIP
                                         ;IF THE ION LIGHT IS OFF
                                         ;CHECK ION FLOP

   01037 060177 CC30:   NIOS CPU         ;TURN ON INTERRUPT (ION)
   01040 060277         NIOC CPU         ;TURN OFF INTERRUPT
   01041 063577         SKPBZ CPU        ;IF (ION) LIGHT IS ON
   01042 063077         HALT             ;CHECK I/O CLEAR INPUT
                                         ;TO THE ION FLOP
```

```
---
↑↑↑

01043 060177 CC32:   NIOS CPU          ;TURN ON INTERRUPT (ION)
01044 060200         NIOC 0            ;PRODUCE A CLEAR PULSE
01045 063477         SKPBN CPU         ;BUT TO DEVICE 0
01046 063077         HALT              ;THE INTERRUPT WAS STILL
                                       ;CLEARED. CHECK E27 12-13
                                       ;CHECK E27 12-13 TO ION

01047 060277 CC34:   NIOC CPU          ;CLEAR INTERRUPT (ION FLOP)
01050 060100         NIOS 0            ;DOES I/O START PULSE
01051 063577         SKPBZ CPU         ;WITH OUT CPU INST TURN ON
01052 063077         HALT              ;THE ION FLOP? YES
                                       ;CHECK E27 8-9-10
                                       ;PART OF THE ION STUF

01053 060277 C36:    NIOC CPU          ;CLEAR ION FLOP
01054 063577         SKPBZ CPU         ;THIS SKIP INST HAS
01055 101010         MOV# 0,0          ;MB8(0),MB9(1) AND
01056 063577         SKPBZ CPU         ;IT TURNED ION ON
01057 063077         HALT              ;LOOK FOR A PHONEY
                                       ;I/O START PULSE.CHECK
                                       ;E91 8-9,E43,E55,ETC

01060 060177 C38:    NIOS CPU          ;SET ION FLOP
01061 103600         ANDR 0,0          ;CHECK FOR A PHONEY
01062 063477         SKPBN CPU         ;I/O CLEAR PULSE. THE I/O
01063 063077         HALT              ;INPUT TO E91 SHOULD
                                       ;PREVENT IT.

01064 063000 C40:    DOC 0,0           ;A DOC TO A DEVICE
                                       ;NOT CPU SHOULD NOT STOP
                                       ;THE MACHINE
                                       ;CHECK E27,8-9-10 TO RUN FLOP

01065 060277 C42:    NIOC CPU          ;CLEAR ION FLOP
01066 060377         NIOP CPU          ;GENERATE A "P" PULSE
01067 063577         SKPBZ CPU         ;CHECK THAT
01070 063077         HALT              ;NO CLEAR PULSE WAS
                                       ;GENERATED


01071 060177 C44:    NIOS CPU          ;SET ION FLOP
01072 060377         NIOP CPU          ;GENERATE A "P" PULSE
01073 063477         SKPBN CPU         ;CHECK THAT
01074 063077         HALT              ;NO START PULSE
                                       ;WAS GENERATED

01075 060177 C46:    NIOS CPU          ;SET ION FLOP
01076 060077         NIO CPU           ;CHECK THAT NO CLEAR
01077 063477         SKPBN CPU         ;PULSE WAS GENERATED
01100 063077         HALT              ;BY (NIO) INSTRUCTION

01101 060277 C48:    NIOC CPU          ;CLEAR ION FLOP
01102 060077         NIO CPU           ;CHECK THAT NO
01103 063577         SKPBZ CPU         ;START PULSE
01104 063077         HALT              ;WAS GENERATED BY

                     .EOT
```

```
01105 102400 D00:    SUB 0,0          ;C(0)=0
01106 040000         STA 0,0          ;STORE IN REGISTER 0
01107 010000         ISZ 0,0          ;INC TO +1
01110 101010         MOV# 0,0
01111 020000         LDA 0,0
01112 101015         MOV# 0,0,SNR     ;CHECK THEAT MB COUNT COUNTED
01113 063077         HALT             ;E74 8-10,E76 8-9-10
                                      ;INPUT TO ONE ENABLE
                                      ;LOGIC AT E68 2-3-6

01114 020000 D02:    LDA 0,0          ;LOAD REGISTER 0 INTO
01115 024000         LDA 1,0          ;AC0 AND AC1. IF C(1)=
01116 106414         SUB# 0,1,SZR     ;C(0)+1 CHECK MB COUNT
01117 063077         HALT             ;E76 8-9-10
                                      ;IF C(0)+1=C(1) CHECK
                                      ;MB DECREMENT E76 2-3-6

01120 020120 D04:    LDA 0,120        ;IF C(1)=C(0)+1
01121 024120         LDA 1,120        ;CHECK MB COUNT
01122 106414         SUB# 0,1,SZR     ;E76 1-8-13
01123 063077         HALT


01124 020130 D06:    LDA 0,130
01125 024130         LDA 1,130        ;IF C(0)+1=C(1)
01126 106414         SUB# 0,1,SZR     ;CHECK MB DECREMENT
01127 063077         HALT             ;E76 4-5-6

01130 020020 D08:    LDA 0,20         ;IF C(1)=C(0)+1
01131 024020         LDA 1,20         ;CHECK MB COUNT LOGIC
01132 106414         SUB# 0,1,SZR     ;THE DEFER INPUT TO E17
01133 063077         HALT

01134 102400 D10:    SUB 0,0
01135 040000         STA 0,0          ;STORE A 0 IN LOCATION 0
01136 014000         DSZ 0            ;DECREMENT TO 177777
01137 101000         MOV 0,0
01140 020000         LDA 0,0          ;IF C(0)=0 CHECK MB
01141 100014         COM# 0,0,SZR     ;DECREMENT LOGIC TO MAB ENABLE
01142 063077         HALT             ;AND NOT MA ENABLE
                                      ;E66 2-3-6,E75 10-11
                                      ;E76 2-3-6
```

```
01143 102000 D12:    ADC 0,0          ;C(0)=-1
01144 040000         STA 0,0          ;STORE IN REGISTER 0
01145 010000         ISZ 0            ;INCREMENTIOG SHOULD
01146 063077         HALT             ;PRODUCE 0 AND A SKIP
                                      ;CHECK E80 8-9-10-11
                                      ;IN SKIP LOGIC.LOOK AT
                                      ;REGISTER 0 ALSO

01147 102520 D14:    SUBZL 0,0        ;C(0)=1
01150 040000         STA 0,0          ;LOCATION 0=+1
01151 022000         LDA 0,@0         ;CHECK THAT DEFER DOES
01152 020000         LDA 0,0          ;NOT DECREMENT LOC 0
01153 101005         MOV 0,0,SNR      ;THERE SHOULD BE NO
01154 063077         HALT             ;MB DECREMENT LEVEL.CHECK
                                      ;E76 4-5-6

01155 102400 D16:    SUB 0,0          ;C(0)=0
01156 040000         STA 0,0          ;CHECK THAT DEFER DOES NOT
01157 022000         LDA 0,@0         ;INCREMENT LOCATION 0
01160 020000         LDA 0,0          ;THERE SHOULD BE NO
01161 101004         MOV 0,0,SZR      ;MB COUNT LEVEL. CHECK
01162 063077         HALT             ;E76 8-1-13

01163 102400 D18:    SUB 0,0          ;C(0)=0
01164 040020         STA 0,20         ;CHECK THAT INDIRECT
01165 022020         LDA 0,@20        ;ADDRESSING WILL INC
01166 020020         LDA 0,20         ;THE AUTO REGISTER 20.
01167 101005         MOV 0,0,SNR      ;CHECK E76 8-13-1,
01170 063077         HALT             ;E17-8,ETC. PERHAPS THE DEFER
                                      ;CYCLE WAS NOT TAKEN. USE
                                      ;MEMORY STEP TO CHECK

01171 102520 D20:    SUBZL 0,0        ;C(0)=1
01172 040030         STA 0,30         ;CHECK THAT INDIRECT
01173 022030         LDA 0,@30        ;ADDRESSING WILL
01174 020030         LDA 0,30         ;DECREMENT AUTO REGISTER
01175 101004         MOV 0,0,SZR      ;30. CHECK E76 4-5-6
01176 063077         HALT
                                      ;IN MB DECREMENT LOGIC
```

```
01177 102400 D22:    SUB 0,0          ;JUST TO CHECK THAT
01200 040000         STA 0,0          ;ISZ DOES NOT SKIP
01201 010000         ISZ 0
01202 101001         MOV 0,0,SKP
01203 063077         HALT

01204 102400 D24:    SUB 0,0
01205 040000         STA 0,0          ;C(LOCATION 0)=0
01206 012000         ISZ 00           ;WILL INCREMENT LOC 0
01207 020000         LDA 0,0          ;IF REGISTER=2
01210 101234         MOVZR# 0,0,SZR   ;MB COUNT OCCURED IN DEFER
01211 063077         HALT             ;CYCLE
                                      ;CHECK E76 8-9-10

01212 102400 D26:    SUB 0,0
01213 040001         STA 0,1          ;CHECK THAT MB
01214 016001         DSZ 01           ;DECREMENT OCCURES
01215 020001         LDA 0,1          ;ONLY IN THE EXECUTE
01216 101004         MOV 0,0,SZR      ;CYCLE OF A DSZ
01217 063077         HALT

01220 102000 D28:    ADC 0,0
01221 040001         STA 0,1
01222 126520         SUBZL 1,1
01223 044000         STA 1,0          ;LDA LOADS
01224 032000         LDA 2,00         ;LOC 0 INSTEAD OF
01225 147014         ADD# 2,1,SZR     ;LOC 1 CHECK E57 8-9-10
01226 063077         HALT             ;(MB ENABLE)

01227 020403 DD28:   LDA 0,D281       ;C(0)=12345
01230 040001         STA 0,1          ;LOCATION 1=12345
01231 126521         SUBZL 1,1,SKP    ;C(1)=1
01232 012345 D281:   12345
01233 044000         STA 1,0          ;C(0)=1
01234 026000         LDA 1,00         ;C(1) SHOULD = 12345
01235 122414         SUB# 1,0,SZR     ;JUST TO USE DEFER
01236 063077         HALT             ;WITH SOME REAL DATA
```

```
01237 102000 D30:   ADC 0,0
01240 040002        STA 0,2          ;LOCATION 2=(-1)
01241 102620        SUBZR 0,0
01242 101400        INC 0,0
01243 040000        STA 0,0          ;LOCATION 0=01
01244 101120        MOVZL 0,0
01245 040001        STA 0,1          ;LOCATION 1=2
01246 032000        LDA 2,00         ;GET LOC 2 USING
01247 150014        COM# 2,2,SZR     ;INDIRECT ADDRESSING
01250 063077        HALT             ;CHECK DEFER LOGIC
                                     ;E38,E39,E10

01251 102400 D32:   SUB 0,0          ;TEST RELATIVE ADDRESSING
01252 040000        STA 0,0          ;LOCATION 0=0, IF RELATIVE
01253 020400        LDA 0,.+0        ;TO PC C(0)WILL BE NOT 0
01254 101015        MOV# 0,0,SNR     ;IF PC NOT ADDED WILL LOAD
01255 063077        HALT             ;LOCATION 0,A ZERO

01256 102400 D34:   SUB 0,0          ;IF NEG EXTEND TURNS ON
01257 020377        LDA 0,377        ;WITHOUT IR7(1) WILL LOAD
01260 100015        COM# 0,0,SNR     ;NON EXIST MEMORY
01261 063077        HALT             ;CHECK E53 8-9,E55 10-11

01262 176400 D36:   SUB 3,3          ;C(3)=0. WITH IR 6-7
01263 021400        LDA 0,0,3        ;BOTH SET SHOULD NOT GET
01264 024777        LDA 1,.-1        ;MAA ENABLE ON FIRST LDA
01265 106415        SUB# 0,1,SNR     ;IF ASSERTED(LDA 0,0,3) WOULD
01266 063077        HALT             ;LOAD ITSELF. CHECK E52 4-6

01267 176400 D38:   SUB 3,3          ;C(3)=0. CHECK NEG
01270 020377        LDA 0,377        ;EXTEND LEVEL. IF NEG
01271 025777        LDA 1,-1,3       ;EXTEND DOES NOT WORK
01272 106415        SUB# 0,1,SNR     ;THE SECOND LDA INSTRUCTION
01273 063077        HALT             ;WILL LOAD LOCATION 377
                                     ;E54,E55 TO CHECK
```

```
---
***

01274 152400 D40:    SUB 2,2          ;C(2)=0. CHECK NEG
01275 020377         LDA 0,377        ;EXTEND LEVEL. IF NEG
01276 025377         LDA 1,-1,2       ;EXTEND DOES NOT ASSERT
01277 106415         SUB# 0,1,SNR     ;THE SECOND LDA INST WILL
01300 063077         HALT             ;LOAD LOCATION 377.
                                      ;CHECK E53 8-10

01301 020400 D42:    LDA 0,.+0        ;LOADS ITSELF
01302 024777         LDA 1,.-1        ;SHOULD LOAD PREVIOUS INST
01303 106414         SUB# 0,1,SZR     ;IF NEG EXTEND FAIL WILL
01304 063077         HALT             ;LOAD 377 LOCATIONS AWAY
                                      ;CHECK E55 10-11,E53 8-9

01305 176400 D44:    SUB 3,3          ;CHECK THAT NEG EXTEND
01306 021577         LDA 0,177,3      ;DOES NOT ASSERT WHEN IR8
01307 024177         LDA 1,177        ;IS ZERO
01310 106414         SUB# 0,1,SZR     ;IF C(0)=-1 CHECK E54 4-6
01311 063077         HALT

01312 176400 D46:    SUB 3,3
01313 054000         STA 3,0          ;CHECK THAT AC3
01314 152000         ADC 2,2          ;GETS ENABLED (EFA)
01315 021400         LDA 0,0,3        ;SHOULD LOAD LOC 0
01316 101004         MOV 0,0,SZR      ;IF C(0)=-1,CHECK
01317 063077         HALT             ;E67 8-9-10

01320 152400 D48:    SUB 2,2          ;CHECK THAT AC2
01321 176000         ADC 3,3          ;ONLY GETS ENABLED
01322 021000         LDA 0,0,2        ;ON EFA CALCULATION
01323 101004         MOV 0,0,SZR      ;IF C(0)=-1,CHECK
01324 063077         HALT             ;E67 4-5-6

01325 152000 D50:    ADC 2,2          ;C(2)=-1
01326 176520         SUBZL 3,3        ;C(3)=+1
01327 054001         STA 3,1          ;LOCATION 1=+1
01330 025400         LDA 1,0,3        ;USE INDEX 3 TO GET LOC 1
01331 136414         SUB# 1,3,SZR     ;CHECK E67 4-5-6
01332 063077         HALT             ;ON AC PRINT


01333 176000 D52:    ADC 3,3          ;C(3)=-1,C(2)=+1,LOC1=1
01334 152520         SUBZL 2,2        ;USE INDEX 2
01335 025000         LDA 1,0,2        ;TO GET LOC 1
01336 132414         SUB# 1,2,SZR     ;CHECK E67 8-9-10
01337 063077         HALT
```

```
01340 000401 E00:    JMP .+1          ;TEST JMP
01341 101001         MOV 0,0,SKP      ;SKIP GOT ADDED IN?
01342 063077         HALT

01343 000402 E02:    JMP .+2          ;TEST JMP
01344 063077         HALT             ;CHECK E42 8-10
                                      ;E11 1-2-4-5-6 LOGIC
                                      ;TO PRODUCE PC CLOCK

01345 102000 E04:    ADC 0,0          ;C(0)=-1
01346 000401         JMP .+1          ;THE JMP INSTRUCTION
01347 100014         COM# 0,0,SZR     ;SOMEHOW MODIFIED C(0)
01350 063077         HALT

01351 020431 E05:    LDA 0,E051       ;CONSTANT(JMP @1)
01352 040000         STA 0,0
01353 020431         LDA 0,E052       ;CONSTANT TO RETURN
01354 040001         STA 0,1
01355 004424         JSR E053+1       ;CHECK THAT MB
01356 000423         JMP E053+1       ;CLOCK WORKS. IF
01357 000422         JMP E053+1       ;IT DOESENT JSR WILL
      001400         .LOC .+20&7760   ;GO TO LOC 0
01400 063077 E053:   HALT             ;CHECK E28 8-9-10
                                      ;E47 8-9
01401 101001         MOV 0,0,SKP
01402 002001 E051:   JMP @1
01403 101001         MOV 0,0,SKP
01404 001400 E052:   E053

01405 004401 E06:    JSR .+1
01406 101001         MOV 0,0,SKP      ;JSR WENT 1 TO FAR
01407 063077         HALT

01410 102400 E08:    SUB 0,0          ;C(0)=0
01411 126400         SUB 1,1          ;C(1)=0
01412 004401         JSR .+1
01413 107014         ADD# 0,1,SZR     ;THE JSR INSTRUCTION
01414 063077         HALT             ;MODIFIED C(0) OR C(1)

01415 102400 E10:    SUB 0,0          ;JSR SHOULD STORE PC
01416 004401         JSR .+1          ;IN C(3). CHECK E67-13
01417 175005         MOV 3,3,SNR      ;THE PC LOAD TO AC LOGIC
01420 063077         HALT             ;E74-12 THE PC LOAD
                                      ;INPUT TO AC WRITE ENABLE

01421 004401 E12:    JSR .+1          ;JSR IS TESTED
01422 020404         LDA 0,E121       ;FOR STORING A RATIONAL
01423 116414         SUB# 0,3,SZR     ;PC VALUE
01424 063077         HALT             ;C(0)=CORRECT
                                      ;C(3)=VALUE STORED
01425 101001         MOV 0,0,SKP
01426 001422 E121:   E12+1
```

```
01427 004402 E14:    JSR .+2          ;CHECK PC STORE
01430 101000         MOV 0,0,         ;FEATURE OF THE JSR
01431 020404         LDA 0,E141       ;INSTRUCTION
01432 116414         SUB# 0,3,SZR     ;IF C(3)=E14+3(AS A ADDRESS)
01433 063077         HALT             ;THE PC WAS LOADED AT
                                      ;TS4 TIME, CHECK E11 5-6
                                      ;AND E9 11-12-13
01434 101001         MOV 0,0,SKP
01435 001430 E141:   E14+1

01436 004402 E16:    JSR .+2          ;PC ENABLE FAILED
01437 063077         HALT             ;TO ASSERT AT TS5 OF
                                      ;THE JSR INSTRUCTION


01440 020406 E18:    LDA 0,E182       ;CHECK THAT THE
01441 040000         STA 0,0          ;PC GETS LOADED IN
01442 006401         JSR @.+1         ;DEFER CYCLE. IF NOT
01443 100000         @0               ;PC IS SET TO FIRST
01444 063077 E181:   HALT             ;CALCULATED EFFECTIVE ADDRESS
                                      ;(@0) IS A ALC INSTRUCTION
                                      ;IF EXECUTED. CHECK
                                      ;E9 11-12-13
01445 101011         MOV# 0,0,SKP
01446 001445 E182:   E181+1

01447 020406 E20:    LDA 0,E202       ;SEE IF JMP
01450 040000         STA 0,0          ;DEFER WILL WORK
01451 002401         JMP @.+1         ;CHECK E11 4-6
01452 100000         @0
01453 063077 E201:   HALT
01454 101011         MOV# 0,0,SKP
01455 001454 E202:   E201+1

01456 032410 E22:    LDA 2,@E221      ;CHECK THAT BMA6(0)
01457 151112         MOVL# 2,2,SZC    ;PREVENTS AUTO INDEX AT
01460 000410         JMP E222+1       ;LOCATION 1020
01461 036406         LDA 3,@E222      ;CHECK E17 6-3
01462 026404         LDA 1,@E221
01463 132414         SUB# 1,2,SZR
01464 063077         HALT
01465 000403         JMP .+3
01466 001020 E221:   1020
01467 101020 E222:   01020

01470 032410 E24:    LDA 2,@E241      ;CHECK THAT BMA5(0)
01471 151112         MOVL# 2,2,SZC    ;PREVENTS AUTO INDEX AT
01472 000410         JMP E242+1       ;LOCATION 2020
01473 036406         LDA 3,@E242
01474 026404         LDA 1,@E241
01475 132414         SUB# 1,2,SZR
01476 063077         HALT
01477 000403         JMP .+3
01500 002020 E241:   2020
01501 102020 E242:   02020
```

```
01502 032410 E26:    LDA 2,@E261        ;CHECK THAT MBA4(@)
01503 151112         MOVL# 2,2,SZC      ;PREVENTS AUTO INDEX
01504 000407         JMP E262           ;AT LOCATION 4020
01505 036406         LDA 3,@E262
01506 026404         LDA 1,@E261
01507 132414         SUB# 1,2,SZR
01510 063077         HALT
01511 000403         JMP .+3
01512 004020 E261:   4020
01513 104020 E262:   @4020

01514 030414 E28:    LDA 2,E282         ;CHECK THAT AUTO
01515 025020         LDA 1,20,2         ;INCREMENT DOES NOT
01516 125112         MOVL# 1,1,SZC      ;WORK AT LOCATIONS
01517 000405         JMP E281           ;10020,20020,40020
01520 037020         LDA 3,@20,2
01521 021020         LDA 0,20,2
01522 106414         SUB# @,1,SZR
01523 063077         HALT               ;SET CONSOL SWITCH TO
                                        ;LOOP ERROR
                                        ;C(2)+20=FAIL LOCATION
                                        ;CHECK E17 11-12-13
01524 151120 E281:   MOVZL 2,2          ;TO MB COUNT LOGIC
01525 151133         MOVZL# 2,2,SNC
01526 000767         JMP E28+1
01527 101001         MOV @,@,SKP
01530 010000 E282:   10000

01531 102000 E30:    ADC @,@            ;CHECK LDA AC ENABLE
01532 020402         LDA @,.+2
01533 101001         MOV @,@,SKP
01534 100000         @@
01535 101134         MOVZL# @,@,SZR
01536 063077         HALT

01537 020400 E32:    LDA @,.+@          ;CHECK STA AC ENABLE
01540 105000         MOV @,1
01541 046402         STA 1,@.+2
01542 151011         MOV# 2,2,SKP       ;EXECTUTE(1) ONLY
01543 000017         17
01544 030017         LDA 2,17
01545 112414         SUB# @,2,SZR
01546 063077         HALT
```

```
01547 020407 E34:    LDA 0,E342        ;USE JSR
01550 176000         ADC 3,3           ;WITH BIT PATTERN
01551 040000         STA 0,0           ;5-6-7=1
01552 007401         JSR @1,3          ;TO CHECK A
01553 116414 E341:   SUB# 0,3,SZR      ;AND ENABLE GATE
01554 063077         HALT              ;AT E70
01555 101001         MOV 0,0,SKP
01556 001553 E342:   E341

01557 176000 E36:    ADC 3,3           ;TEST MEM ENABLE
01560 126520         SUBZL 1,1         ;IN THE DEFER CYCLE
01561 044001         STA 1,1           ;IF LOC 0=(-1)
01562 056001         STA 3,@1          ;CHECK E68 8-9-10
01563 030001         LDA 2,1
01564 150014         COM# 2,2,SZR
01565 063077         HALT

01566 020410 E38:    LDA 0,E381
01567 040000         STA 0,0
01570 024400         LDA 1,.
01571 044017         STA 1,17
01572 032000         LDA 2,@0          ;C(LOC 0)=17
01573 132414         SUB# 1,2,SZR      ;SHOULD BE SAME
01574 063077         HALT              ;JUST A CHECK OF DEFER
01575 101001         MOV 0,0,SKP
01576 000017 E381:   17


01577 101000         MOV 0,0
01600 002401         JMP @.+1          ;LOOP THE TEST
01601 000043         A02


                     .END
```