Vit :	110 /	300 /	600 /	1200
Ad: 1F85	57	76	A7	3D
87	04	01	00	00
94	7E	E5	45	76
96	08	02	01	00
B9	08	06	04	02
C4	FF	64	25	86
C6	17	06	03	01
D0	8A	F0	50	81
D2	08	03	01	00

SC/MP 1

FENARD. XAVIER
9 Rue Pierre Lescot 75001 Paris
tel: 233-39-70

NATIONAL SEMICONDUCTOR

SC/MP

CROSS ASSEMBLER

TITLE: NIBL 12/17/76 N-SC/MP

Ti Fernand le langage intermédiaire à la fin du bandeau

Ad:	110	300	600	1200
85	C3	29	8A	BB
87	08	03	01	00
94	45	11	D4	34
96	11	06	02	01
B9	11	06	03	01
C4	BB	6C	2D	99
C6	2F	06	03	01
D0	54	21	E5	44
D2	11	06	02	01

```
                .TITLE  NIBL,'12/17/76'
0001            .LIST   1

                ;*************************************************************
                ;*                                                           *
                ;*      WE ARE TIED DOWN TO A LANGUAGE WHICH                 *
                ;*      MAKES UP IN OBSCURITY WHAT IT LACKS                  *
                ;*      IN STYLE.                                            *
                ;*                      -- TOM STOPPARD                      *
                ;*                                                           *
                ;*************************************************************

0020    TSTBIT  =       020             ;I.L. INSTRUCTION FLAGS
0040    JMPBIT  =       040
0080    CALBIT  =       080
0001    P1      =       1               ;SC/MP POINTER ASSIGNMENTS
0002    P2      =       2
0003    P3      =       3
FF80    EREG    =       -128            ;THE EXTENSION REGISTER

                ; DISPLACEMENTS FOR RAM VARIABLES USED BY INTERPRETER

FFFF    DOPTR   =       -1              ;DO-STACK POINTER
FFFE    FORPTR  =       -2              ;FOR-STACK POINTER
FFFD    LSTK    =       -3              ;ARITHMETIC STACK POINTER
FFFC    SBRPTR  =       -4              ;GOSUB STACK POINTER
FFFB    PCLOW   =       -5              ;I.L. PROGRAM COUNTER
FFFA    PCHIGH  =       -6
FFF9    PCSTK   =       -7              ;I.L. CALL STACK POINTER
FFF8    LOLINE  =       -8              ;CURRENT LINE NUMBER
FFF7    HILINE  =       -9
FFF6    PAGE    =       -10             ;VALUE OF CURRENT PAGE
FFF5    LISTNG  =       -11             ;LISTING FLAG
FFF4    RUNMOD  =       -12             ;RUN/EDIT FLAG
FFF3    LABLLO  =       -13
FFF2    LABLHI  =       -14
FFF1    P1LOW   =       -15             ;SPACE TO SAVE CURSOR
FFF0    P1HIGH  =       -16
FFEF    LO      =       -17
FFEE    HI      =       -18
FFED    FAILLO  =       -19
FFEC    FAILHI  =       -20
FFEB    NUM     =       -21
FFEA    TEMP    =       -22
FFE9    TEMP2   =       -23
FFE8    TEMP3   =       -24
FFE7    CHRNUM  =       -25
FFE6    RNDF    =       -26
FFE5    RNDX    =       -27             ;SEEDS FOR RANDOM NUMBER
FFE4    RNDY    =       -28

                ; ALLOCATION OF RAM FOR NIBL VARIABLES, STACKS,
                ;  AND LINE BUFFER
```

```
/MP ASSEMBLER REV-A
BL  12/17/76


                        .=01000+28
0000                    .=.+52                  ;NIBL VARIABLES A-Z
101C    VARS:           .=.+26                  ;ARITHMETIC STACK
1050    AESTK:          .=.+16                  ;GOSUB STACK
106A    SBRSTK:         .=.+16                  ;DO/UNTIL STACK
107A    DOSTAK:         .=.+28                  ;FOR/NEXT STACK
108A    FORSTK:         .=.+48                  ;I.L. CALL STACK
10A6    PCSTAK:         .=.+74                  ;LINE BUFFER
10D6    LBUF:           .=0                     ;USER'S PROGRAM
1120    PGM:

                        .MACRO  LDPI,P,VAL
                        .MLOC   TEMP
                        .SET    TEMP,VAL
                        LDI     H(TEMP)
                        XPAH    P
                        LDI     L(TEMP)
                        XPAL    P
                        .ENDM


                ;*****************************************
                ;*      INITIALIZATION OF NIBL           *
                ;*****************************************


0000 08                 NOP
0001                    LDPI    P2,VARS         ;POINT P2 AT VARIABLES
0007                    LDPI    P1,PGM          ;POINT P1 AT PAGE ONE PROGRAM
000D C4FF               LDI     -1              ;STORE -1 AT START OF PROGRAM
000F C900               ST      0(P1)
0011 C901               ST      1(P1)
0013 C40D               LDI     0D              ;ALSO STORE A DUMMY
0015 C9FF               ST      -1(P1)          ;  CARRIAGE RETURN
0017 C402               LDI     2               ;POINT P2 AT PAGE 2,
0019 CAF6               ST      PAGE(P2)        ; INITIALLY SET PAGE TO 2
001B 31                 XPAL    P1
001C C420               LDI     020
001E 35                 XPAH    P1
001F B902               DLD     2(P1)           ;CHECK IF THERE IS REALLY
0021 01                 XAE                     ; A PROGRAM IN PAGE 2:
0022 C180               LD      EREG(P1)        ; IF FIRST LINE LENGTH
0024 E40D               XRI     0D              ; POINTS TO CARR. RETURN
0026 9802               JZ      $0              ; AT END OF LINE
0028 BAF6               DLD     PAGE(P2)        ;IF NOT, PAGE = 1
002A C420       $0:     LDI     020
002C 35         $LOOP:  XPAH    P1
002D C4FF               LDI     -1              ;STORE -1 IN 2 CONSECUTIVE
002F C900               ST      (P1)            ; LOCATIONS AT START OF PAGE
0031 C901               ST      1(P1)
0033 C40D               LDI     0D              ;ALSO PUT A DUMMY END-OF-LINE
0035 C9FF               ST      -1(P1)          ; JUST BEFORE TEXT
0037 35                 XPAH    P1              ;UPDATE P1 TO POINT TO
0038 02                 CCL                     ; NEXT PAGE (UNTIL PAGE=0)
0039 F410               ADI     010             ;REPEAT INITIALIZATION
```

```
109 003B E480           XRI     080             ; FOR PAGES 2-7
110 003D 9804           JZ      $1
111 003F E480           XRI     080
112 0041 90E9           JMP     $LOOP
113 0043 C400   $1:     LDI     0               ;CLEAR SOME FLAGS
114 0045 CAF4           ST      RUNMOD(P2)
115 0047 CAF5           ST      LISTNG(P2)
116 0049 C438           LDI     L(BEGIN)        ;INITIALIZE IL PC SO THAT
117 004B CAFB           ST      PCLOW(P2)       ; NIBL PROGRAM
118 004D C40C           LDI     H(BEGIN)        ; IS EXECUTED IMMEDIATELY
119 004F CAFA           ST      PCHIGH(P2)
120 0051 C400   CLEAR:  LDI     0
121 0053 CAEA           ST      TEMP(P2)
122 0055 01             XAE
123 0056 C400   CLEAR1: LDI     0               ;SET ALL VARIABLES
124 0058 CA80           ST      EREG(P2)        ; TO ZERO
125 005A AAEA           ILD     TEMP(P2)
126 005C 01             XAE
127 005D C434           LDI     52
128 005F 60             XRE
129 0060 9CF4           JNZ     CLEAR1
130 0062 C450           LDI     L(AESTK)        ;INITIALIZE SOME STACKS:
131 0064 CAFD           ST      LSTK(P2)        ; ARITHMETIC STACK,
132 0066 C47A           LDI     L(DOSTAK)
133 0068 CAFF           ST      DOPTR(P2)       ; DO/UNTIL STACK,
134 006A C46A           LDI     L(SBRSTK)
135 006C CAFC           ST      SBRPTR(P2)      ; GOSUB STACK,
136 006E C4A6           LDI     L(PCSTAK)
137 0070 CAF9           ST      PCSTK(P2)       ; I.L. CALL STACK,
138 0072 C48A           LDI     L(FORSTK)
139 0074 CAFE           ST      FORPTR(P2)      ; FOR/NEXT STACK
140
141
142             ;****************************************
143             ;*   INTERMEDIATE LANGUAGE EXECUTOR     *
144             ;****************************************
145
146 0076 C2FB   EXECIL: LD      PCLOW(P2)       ;SET P3 TO CURRENT
147 0078 33             XPAL    P3              ; IL PC.
148 0079 C2FA           LD      PCHIGH(P2)
149 007B 37             XPAH    P3
150 007C C701   CHEAT:  LD      @1(P3)
151 007E 01             XAE                     ;GET NEW I.L. INSTRUCTION
152 007F C701           LD      @1(P3)          ; INTO P3 THROUGH
153 0081 33             XPAL    P3              ; OBSCURE METHODS
154 0082 CAFB           ST      PCLOW(P2)       ;SIMULTANEOUSLY, INCREMENT
155 0084 40             LDE                     ; THE I.L. PC BY 2
156 0085 D40F           ANI     0F              ;REMOVE FLAG FROM INSTRUCTION
157 0087 DC00           ORI     ./256           ; TURN INTO ACTUAL ADDRESS,
158 0089 37             XPAH    P3              ; PUT BACK INTO P3
159 008A CAFA           ST      PCHIGH(P2)
160 008C 40             LDE
161 008D D4F0           ANI     0F0             ;CHECK IF I.L. INSTRUCTION
162 008F E420           XRI     TSTBIT          ; IS A 'TEST'
```

```
                        JZ      TST
                        XRI     CALBIT!TSTBIT   ;CHECK FOR I.L. CALL
                        JZ      ILCALL
                        XRI     JMPBIT!CALBIT   ;CHECK FOR I.L. JUMP
                        JZ      CHEAT           ;I.L. JUMP IS TRIVIAL
NOJUMP:                 XPPC    P3              ;MUST BE AN ML SUBROUTINE
                        JMP     EXECIL          ; IF NONE OF THE ABOVE


                ;*****************************************
                ;*      INTERMEDIATE LANGUAGE CALL       *
                ;*****************************************

ILCALL:         LD      PCSTK(P2)
                XRI     L(LBUF)             ;CHECK FOR STACK OVERFLOW
                JNZ     ILC1
                LDI     10
                JMP     E0A
ILC1:           XRI     L(LBUF)             ;RESTORE ACCUMULATOR
                XPAL    P3                  ;SAVE LOW BYTE OF NEW
                ST      TEMP(P2)            ; I.L. PC IN TEMP
                LDI     H(PCSTAK)           ;POINT P3 AT I.L.
                XPAH    P3                  ; SUBROUTINE STACK
                XAE                         ;SAVE NEW I.L. PC HIGH IN EX
                LD      PCLOW(P2)           ;SAVE OLD I.L. PC ON STACK
                ST      @1(P3)
                LD      PCHIGH(P2)
                ST      @1(P3)
                LD      TEMP(P2)            ;GET LOW BYTE OF NEW
                XPAL    P3                  ; I.L. PC INTO P3 LOW
                ST      PCSTK(P2)           ;UPDATE I.L. STACK POINTER
                LDE                         ;GET HIGH BYTE OF NEW
                XPAH    P3                  ; I.L. PC INTO P3 HIGH
CHEAT1:         JMP     CHEAT


                ;*****************************************
                ;*      I.L. 'TEST' INSTRUCTION          *
                ;*****************************************

                .LOCAL
TST:            ST      CHRNUM(P2)          ;CLEAR NUMBER OF CHARS SCANNE
$SCAN:          LD      @1(P1)              ;SLEW OFF SPACES
                XRI     ' '
                JZ      $SCAN
                LD      @-1(P1)             ;REPOSITION CURSOR
                LD      PCHIGH(P2)          ;POINT P3 AT I.L. TABLE
                XPAH    P3
                ST      FAILHI(P2)          ;OLD P3 BECOMES THE
                LD      PCLOW(P2)           ; TEST FAIL ADDRESS
                XPAL    P3
                ST      FAILLO(P2)
$LOOP:          LD      @1(P3)
                XAE                         ;SAVE CHAR FROM TABLE
```

```
/MP ASSEMBLER REV-A
BL  12/17/76


21  00D9 BAE7            DLD     CHRNUM(P2)      ;DECREMENT CHAR COUNT
218 00DB 40              LDE                     ;GET CHAR BACK
219 00DC D47F            ANI     07F             ;SCRUB OFF FLAG (IF ANY)
220 00DE E501            XOR     @1(P1)          ;IS CHAR EQUAL TO TEXT CHAR?
221 00E0 9C07            JNZ     $NEQ            ;NO - END TEST
                                                 ;YES - BUT IS IT LAST CHAR?
222 00E2 40              LDE                     ;IF NOT, CONTINUE TO COMPARE
223 00E3 94F1            JP      $LOOP           ;IF SO, GET NEXT I.L.
224 00E5 9095            JMP     CHEAT           ; INSTRUCTION
225 00E7 908D    XO:     JMP     EXECIL          ;RESTORE P1 TO
226 00E9 C2E7    $NEQ:   LD      CHRNUM(P2)      ; ORIGINAL VALUE
227 00EB 01              XAE
228 00EC C580            LD      @EREG(P1)
229 00EE C2ED            LD      FAILLO(P2)      ;LOAD TEST-FAIL ADDRESS
230 00F0 33              XPAL    P3              ; INTO P3
231 00F1 C2EC            LD      FAILHI(P2)
232 00F3 37              XPAH    P3
233 00F4 90CA            JMP     CHEAT1          ;GET NEXT I.L. INSTRUCTION
234
23
236                      ;****************************************
237                      ;*        I.L. SUBROUTINE RETURN        *
238                      ;****************************************
239
240 00F6 C410    RTN:    LDI     H(PCSTAK)       ;POINT P3 AT I.L. PC STACK
241 00F8 37              XPAH    P3
242 00F9 C2F9            LD      PCSTK(P2)
243 00FB 33              XPAL    P3
244 00FC C7FF            LD      @-1(P3)         ;GET HIGH PART OF OLD PC
245 00FE 01              XAE
246 00FF C7FF            LD      @-1(P3)         ;GET LOW PART OF OLD PC
247 0101 33              XPAL    P3
248 0102 CAF9            ST      PCSTK(P2)       ;UPDATE IL STACK POINTER
249 0104 40              LDE
250 0105 37              XPAH    P3              ;P3 NOW HAS OLD IL PC
251 0106 90B8            JMP     CHEAT1
252 0108 9041    EOA:    JMP     EO
253                      ,'
25
255                      ;****************************************
256                      ;*     SAVE GOSUB RETURN ADDRESS        *
257                      ;****************************************
258
259 010A C2FC    SAV:    LD      SBRPTR(P2)
260 010C E47A            XRI     L(DOSTAK)       ;CHECK FOR MORE
261 010E 981C            JZ      SAV2            ; THAN 8 SAVES
262 0110 AAFC            ILD     SBRPTR(P2)
263 0112 AAFC            ILD     SBRPTR(P2)
264 0114 33              XPAL    P3              ;SET P3 TO
265 0115 C410            LDI     H(SBRSTK)       ; SUBROUTINE STACK TOP.
266 0117 37              XPAH    P3
267 0118 C2F4            LD      RUNMOD(P2)      ;IF IMMEDIATE MODE,
268 011A 980A            JZ      SAV1            ; SAVE NEGATIVE ADDRESS.
269 011C 35              XPAH    P1              ;SAVE HIGH PORTION
270 011D CBFF            ST      -1(P3)          ; OF CURSOR
```

```
271 011F 35             XPAH    P1
272 0120 31             XPAL    P1                  ;SAVE LOW PORTION
273 0121 CBFE           ST      -2(P3)              ; OF CURSOR
274 0123 31             XPAL    P1
275 0124 90C1           JMP     X0                  ;RETURN
276 0126 C4FF   SAV1:   LDI     -1                  ;IMMEDIATE MODE
277 0128 CBFF           ST      -1(P3)              ; RETURN ADDRESS IS
278 012A 90BB           JMP     X0                  ; NEGATIVE.
279 012C C40A   SAV2:   LDI     10                  ;ERROR: MORE THAN
280 012E 901B           JMP     E0                  ; 8 GOSUBS
281
282
283             ;****************************************
284             ;*      CHECK STATEMENT FINISHED        *
285             ;****************************************
286
287 0130 C501   DONE:   LD      @1(P1)              ;SKIP SPACES
288 0132 E420           XRI     ' '
289 0134 98FA           JZ      DONE
290 0136 E42D           XRI     ' ' ! 0D            ;IS IT CARRIAGE RETURN?
291 0138 9804           JZ      DONE1               ;YES - RETURN
292 013A E437           XRI     037                 ;IS CHAR A ':' ?
293 013C 9C01           JNZ     DONE2               ;NO - ERROR
294 013E 3F     DONE1:  XPPC    P3                  ;YES - RETURN
295 013F C404   DONE2:  LDI     4
296 0141 9008           JMP     E0
297
298
299             ;****************************************
300             ;*          RETURN FROM GOSUB           *
301             ;****************************************
302
303 0143 C2FC   RSTR:   LD      SBRPTR(P2)
304 0145 E46A           XRI     L(SBRSTK)           ;CHECK FOR RETURN
305 0147 9C04           JNZ     RSTR1               ; W/O GOSUB
306 0149 C409           LDI     9
307 014B 9040   E0:     JMP     E1                  ;REPORT THE ERROR
308 014D BAFC   RSTR1:  DLD     SBRPTR(P2)
309 014F BAFC           DLD     SBRPTR(P2)          ;POP GOSUB STACK,
310 0151 33             XPAL    P3                  ; PUT PTR INTO P3
311 0152 C410           LDI     H(SBRSTK)
312 0154 37             XPAH    P3
313 0155 C301           LD      1(P3)               ;IF ADDRESS NEGATIVE,
314 0157 9406           JP      RSTR2               ; SUBROUTINE WAS CALLED
315 0159 C400           LDI     0                   ; FROM EDIT MODE,
316 015B CAF4           ST      RUNMOD(P2)          ; SO RETURN TO EDITING
317 015D 9088   X1:     JMP     X0
318 015F 35     RSTR2:  XPAH    P1                  ;RESTORE CURSOR HIGH
319 0160 C300           LD      0(P3)
320 0162 31             XPAL    P1                  ;RESTORE CURSOR LOW
321 0163 C401           LDI     1                   ;SET RUN MODE
322 0165 CAF4           ST      RUNMOD(P2)
323 0167 90F4           JMP     X1
324
```

```
825
826                 ;****************************************
827                 ;*     TRANSFER TO NEW STATEMENT        *
828                 ;****************************************
829
830 0169 C2F2   XFER:   LD      LABLHI(P2)      ;CHECK FOR NON-EXISTENT LINE
831 016B 9404         :  JP      XFER1
832 016D C408         :  LDI     8
833 016F 901C         :  JMP     E1
834 0171 C401   XFER1:  LDI     1               ;SET RUN MODE TO 1
835 0173 CAF4           ST      RUNMOD(P2)
836 0175 3F             XPPC    P3
837
838
839                 ;****************************************
840                 ;*     PRINT STRING IN TEXT             *
841                 ;****************************************
842
843 0176        PRS:     LDPI    P3,PUTC-1       ;POINT P3 AT PUTC ROUTINE
844 017C C501           LD      @1(P1)          ;LOAD NEXT CHAR
845 017E E422           XRI     '"'             ;IF ", END OF
846 0180 98DB           JZ      X1              ; STRING
847 0182 E42F           XRI     02F             ;IF CR, ERROR
848 0184 9805           JZ      PRS1
849 0186 E40D           XRI     0D              ;RESTORE CHAR
850 0188 3F             XPPC    P3              ;PRINT CHAR
851 0189 90EB           JMP     PRS             ;GET NEXT CHAR
852 018B C407   PRS1:   LDI     7               ;SYNTAX ERROR
853 018D 9035   E1:     JMP     E2
854
855
856                 ;****************************************
857                 ;*     PRINT NUMBER ON STACK            *
858                 ;****************************************
859
860                 ; THIS ROUTINE IS BASED ON DENNIS ALLISON'S BINARY TO DECIMAL
861                 ; CONVERSION ROUTINE IN VOL. 1, #1 OF "DR. DOBB'S JOURNAL",
862                 ; BUT IS MUCH MORE OBSCURE BECAUSE OF THE STACK MANIPULATION.
863
864                       .LOCAL
865 018F C410   PRN:    LDI     H(AESTK)        ;POINT P3 AT A.E. STACK
866 0191 37             XPAH    P3
867 0192 AAFD           ILD     LSTK(P2)
868 0194 AAFD           ILD     LSTK(P2)
869 0196 33             XPAL    P3
870 0197 C40A           LDI     10              ;PUT 10 ON STACK (WE'LL BE
871 0199 CBFE           ST      -2(P3)          ; DIVIDING BY IT LATER)
872 019B C400           LDI     0
873 019D CBFF           ST      -1(P3)
874 019F C405           LDI     5               ;SET CHRNUM TO POINT TO PLACE
875 01A1 CAE7           ST      CHRNUM(P2)      ;.IN STACK WHERE WE STORE
876 01A3 C4FF           LDI     -1              ; THE CHARACTERS TO PRINT
877 01A5 CB05           ST      5(P3)           ;FIRST CHAR IS A FLAG (-1)
878 01A7 C3FD           LD      -3(P3)          ;CHECK IF NUMBER IS NEGATIVE
```

```
                        JP      $1
                        LDI     '-'             ;PUT '-' ON STACK, AND NEGATE
                        ST      4(P3)           ; THE NUMBER
                        LDI     0
                        SCL
                        CAD     -4(P3)
                        ST      -4(P3)
                        LDI     0
                        CAD     -3(P3)
                        ST      -3(P3)
                        JMP     X1              ;GO DO DIVISION BY 10
                $1:     LDI     ' '             ;IF POSITIVE, PUT ' ' ON
                        ST      4(P3)           ; STACK BEFORE DIVISION
                X4:     JMP     X1
                E2:     JMP     ERR1

                ; THE DIVISION IS PERFORMED, THEN CONTROL IS TRANSFERRED
                ; TO PRN1, WHICH FOLLOWS.

                PRN1:   ILD     LSTK(P2)        ;POINT P1 AT A.E. STACK
                        ILD     LSTK(P2)
                        XPAL    P1
                        LDI     H(AESTK)
                        XPAH    P1
                        ILD     CHRNUM(P2)      ;INCREMENT CHARACTER STACK
                        XAE                     ; POINTER, PUT IN EX. REG.
                        LD      1(P1)           ;GET REMAINDER FROM DIVIDE,
                        ORI     '0'
                        ST      EREG(P1)        ; PUT IT ON THE STACK
                        LD      -3(P1)          ;IS THE QUOTIENT ZERO YET?
                        OR      -4(P1)
                        JZ      $PRNT           ;YES - GO PRINT THE NUMBER
                        LDI     H(PRNUM1)       ;NO - CHANGE THE I.L. PC
                        ST      PCHIGH(P2)      ; SO THAT DIVIDE IS
                        LDI     L(PRNUM1)       ; PERFORMED AGAIN
                        ST      PCLOW(P2)
                        JMP     X4              ;GO DO DIVISION BY 10 AGAIN
                $PRNT:  LDPI    P3,PUTC-1       ;POINT P3 AT PUTC ROUTINE
                        LD      LISTNG(P2)      ;IF LISTING, SKIP PRINTING
                        JNZ     $2              ; LEADING SPACE
                        LD      4(P1)           ;PRINT EITHER '-'
                        XPPC    P3              ; OR LEADING SPACE
                        LD      CHRNUM(P2)      ;GET EX. REG. VALUE BACK
                        XAE
                $2:     LD      @EREG(P1)       ;POINT P3 AT FIRST CHAR
                        LD      (P1)            ; TO BE PRINTED
                $LOOP:  XPPC    P3              ;PRINT THE CHARACTER
                        LD      @-1(P1)         ;GET NEXT CHARACTER
                        JP      $LOOP           ;REPEAT UNTIL = -1
                        LDI     L(AESTK)
                        ST      LSTK(P2)        ;CLEAR THE A.E. STACK
                        LD      LISTNG(P2)      ;PRINT A TRAILING SPACE
                        JNZ     X4              ; IF NOT LISTING PROGRAM
                        LDI     ' '
```

```
433 020A 3F             XPPC    P3
434 020B 90B5           JMP     X4
435
436
437                     ;*****************************************
438                     ;*      CARRIAGE RETURN/LINE FEED        *
439                     ;*****************************************
440
441 020D        NLINE:  LDPI    P3,PUTC-1       ;POINT P3 AT PUTC ROUTINE
442 020F C40D           LDI     0D              ;CARRIAGE RETURN
443 0211 3F             XPPC    P3
444 0212 C40A           LDI     0A              ;LINE FEED
445 0214 3F             XPPC    P3
446 0215 90AB   X5:     JMP     X4
447
448
449                     ;*****************************************
450                     ;*      ERROR  ROUTINE                   *
451                     ;*****************************************
452
453                     .LOCAL
454 0217 C405   ERR:    LDI     5               ;SYNTAX ERROR
455 0219 CAEB   ERR1:   ST      NUM(P2)         ;SAVE ERROR #
456 021B C2EB   ERR2:   LD      NUM(P2)
457 021D CAEA           ST      TEMP(P2)
458 021F                LDPI    P3,PUTC-1       ;POINT P3 AT PUTC
459 0221 C40D           LDI     0D              ;PRINT CR/LF
460 0223 3F             XPPC    P3
461 0224 C40A           LDI     0A
462 0226 3F             XPPC    P3
463 0227                LDPI    P1,MESGS        ;P1 -> ERROR MESSAGES
464 0229 BAEB   $1:     DLD     NUM(P2)         ;IS THIS THE RIGHT MESSAGE?
465 022B 9806           JZ      $MSG            ;YES - GO PRINT IT
466 022D C501   $LOOP:  LD      @1(P1)          ;NO - SCAN THROUGH TO
467 022F 94FC           JP      $LOOP           ; NEXT MESSAGE
468 0231 90F6           JMP     $1
469 0233 C501   $MSG:   LD      @1(P1)          ;GET MESSAGE CHAR
470 0235 3F             XPPC    P3              ;PRINT IT
471 0236 C1FF           LD      -1(P1)          ;IS MESSAGE DONE?
472 0238 94F9           JP      $MSG            ;NO - GET NEXT CHAR
473 023A C2EA           LD      TEMP(P2)        ;WAS THIS A BREAK MESSAGE?
474 023C E40E           XRI     14
475 023E 980D           JZ      $3              ;YES - SKIP PRINTING 'ERROR'
476 0240                LDPI    P1,MESGS        ;NO - PRINT 'ERROR'
477 0246 C501   $2:     LD      @1(P1)          ;GET CHARACTER
478 0248 3F             XPPC    P3              ;PRINT IT
479 0249 C1FF           LD      -1(P1)          ;DONE?
480 024B 94F9           JP      $2              ;NO - REPEAT LOOP
481 024D C2F4   $3:     LD      RUNMOD(P2)      ;DON'T PRINT LINE #
482 024F 984D           JZ      FIN             ; IF IMMEDIATE MODE
483 0251 C420           LDI     ' '
484 0253 3F             XPPC    P3              ;SPACE
485 0254 C441           LDI     'A'             ;AT
486 0256 3F             XPPC    P3
```

```
 487 0263 C454          LDI     'T'
 488 0265 3F            XPPC    P3
 489 0266 C410          LDI     H(AESTK)        ;POINT P3 AT A.E. STACK
 490 0268 37            XPAH    P3
 491 0269 AAFD          ILD     LSTK(P2)
 492 026B AAFD          ILD     LSTK(P2)
 493 026D 33            XPAL    P3
 494 026E C2F7          LD      HILINE(P2)      ;GET HIGH BYTE OF LINE #
 495 0270 CBFF          ST      -1(P3)          ;PUT ON STACK
 496 0272 C2F8          LD      LOLINE(P2)      ;GET LOW BYTE OF LINE #
 497 0274 CBFE          ST      -2(P3)          ;PUT ON STACK
 498 0276 C431          LDI     L(ERRNUM)       ;GO TO PRN
 499 0278 CAFB          ST      PCLOW(P2)
 500 027A C40E          LDI     H(ERRNUM)
 501 027C CAFA          ST      PCHIGH(P2)
 502 027E 9099  X5A:    JMP     X5
 503
 504
 505                    ;****************************************
 506                    ;*      BREAK, NXT, FIN, & STRT         *
 507                    ;****************************************
 508
 509 0280 C40E  BREAK:  LDI     14              ;*** CAUSE A BREAK ***
 510 0282 9099  E3A:    JMP     ERR1
 511                                            ;*** NEXT STATEMENT ***
 512 0284 C2F4  NXT:    LD      RUNMOD(P2)      ;IF IN EDIT MODE,
 513 0286 9822          JZ      FIN             ; STOP EXECUTION
 514 0288 C100          LD      (P1)            ;IF WE HIT END OF FILE,
 515 028A D480          ANI     080             ; FINISH UP THINGS
 516 028C 9C1C          JNZ     FIN
 517 028E 06            CSA                     ;BREAK IF SOMEONE IS
 518 028F D420          ANI     020             ; TYPING ON THE CONSOLE
 519 0291 98ED          JZ      BREAK
 520 0293 C1FF          LD      -1(P1)          ;GET LAST CHARACTER SCANNED
 521 0295 E40D          XRI     0D              ;WAS IT CARRIAGE RETURN?
 522 0297 9C08          JNZ     NXT1            ;YES - SKIP FOLLOWING UPDAT
 523 0299 C501          LD      @1(P1)          ;GET HIGH BYTE OF NEXT LINE
 524 029B CAF7          ST      HILINE(P2)      ;SAVE IT
 525 029D C502          LD      @2(P1)          ;GET LOW BYTE OF LINE #, SK
 526 029F CAF8          ST      LOLINE(P2)      ; LINE LENGTH BYTE
 527 02A1 C40C  NXT1:   LDI     H(STMT)         ;GO TO 'STMT' IN IL TABLE
 528 02A3 CAFA          ST      PCHIGH(P2)
 529 02A5 C406          LDI     L(STMT)
 530 02A7 CAFB          ST      PCLOW(P2)
 531 02A9 3F            XPPC    P3
 532
 533 02AA C400  FIN:    LDI     0               ;*** FINISH EXECUTION ***
 534 02AC CAF4          ST      RUNMOD(P2)      ;CLEAR RUN MODE
 535 02AE C450          LDI     L(AESTK)        ;CLEAR ARITHMETIC STACK
 536 02B0 CAFD          ST      LSTK(P2)
 537 02B2 C41C          LDI     L(START)        ;MODIFY I.L. PC TO RETURN
 538 02B4 CAFB          ST      PCLOW(P2)       ; TO PROMPT FOR COMMAND
 539 02B6 C40C          LDI     H(START)
 540 02B8 CAFA          ST      PCHIGH(P2)
```

```
541 02BA C4A6           LDI     L(PCSTAK)
542 02BC CAF9           ST      PCSTK(P2)
543 02BE 90BE           JMP     X5A
544                                         ;*** START EXECUTION ***
545 02C0 AAF4   STRT:   ILD     RUNMOD(P2)  ;RUN MODE = 1
546 02C2 C2E9           LD      TEMP2(P2)   ;POINT CURSOR TO
547 02C4 35             XPAH    P1          ; START OF NIBL PROGRAM
548 02C5 C2E8           LD      TEMP3(P2)
549 02C7 31             XPAL    P1
550 02C8 C46A           LDI     L(SBRSTK)   ;EMPTY SOME STACKS:
551 02CA CAFC           ST      SBRPTR(P2)  ; GOSUB STACK,
552 02CC C48A           LDI     L(FORSTK)
553 02CE CAFE           ST      FORPTR(P2)  ; FOR STACK
554 02D0 C47A           LDI     L(DOSTAK)
555 02D2 CAFF           ST      DOPTR(P2)   ; & DO/UNTIL STACK
556 02D4 3F             XPPC    P3          ;RETURN
557 02D5 90A7   X6:     JMP     X5A
558 02D7 90A9   E4:     JMP     E3A
559
560
561                     ;*************************************
562                     ;*        LIST NIBL PROGRAM          *
563                     ;*************************************
564
565 02D9 C100   LST:    LD      (P1)        ;CHECK FOR END OF FILE
566 02DB E480           XRI     080
567 02DD 9418           JP      LST2
568 02DF C410           LDI     H(AESTK)    ;GET LINE NUMBER ONTO STACK
569 02E1 37             XPAH    P3
570 02E2 AAFD           ILD     LSTK(P2)
571 02E4 AAFD           ILD     LSTK(P2)
572 02E6 33             XPAL    P3
573 02E7 C501           LD      @1(P1)
574 02E9 CBFF           ST      -1(P3)
575 02EB C501           LD      @1(P1)
576 02ED CBFE           ST      -2(P3)
577 02EF C501           LD      @1(P1)      ;SKIP OVER LINE LENGTH
578 02F1 C401           LDI     1
579 02F3 CAF5           ST      LISTNG(P2)  ;SET LISTING FLAG
580 02F5 90DE           JMP     X6          ;GO PRINT LINE NUMBER
581 02F7 C400   LST2:   LDI     0
582 02F9 CAF5           ST      LISTNG(P2)  ;CLEAR LISTING FLAG
583 02FB C402           JS      P3,NXT      ;GO TO NXT
584 0302 90D1   X6A:    JMP     X6
585 0304 90D1   E5:     JMP     E4
586 0306        LST3:   LDPI    P3,PUTC-1   ;POINT P3 AT PUTC
587 030C 06     LST4:   CSA
588 030D D420           ANI     020
589 030F 98E6           JZ      LST2        ;IF TYPING, STOP
590 0311 C501           LD      @1(P1)      ;GET NEXT CHAR
591 0313 E40D           XRI     0D          ;TEST FOR CR
592 0315 9805           JZ      LST5
593 0317 E40D           XRI     0D          ;GET CHARACTER
594 0319 3F             XPPC    P3          ;PRINT CHARACTER
```

```
 595 031A 90F0          JMP     LST4
 596 031C C40D  LST5:   LDI     0D              ;CARRIAGE RETURN
 597 031E 3F            XPPC    P3
 598 031F C40A          LDI     0A              ;LINE FEED
 599 0321 3F            XPPC    P3
 600 0322 02            CCL
 601 0323 C44B          LDI     L(LIST3)
 602 0325 CAFB          ST      PCLOW(P2)
 603 0327 C40C          LDI     H(LIST3)
 604 0329 CAFA          ST      PCHIGH(P2)
 605 032B 90AC          JMP     LST             ;GET NEXT LINE
 606
 607
 608                    ;****************************************
 609                    ;*          ADD AND SUBTRACT            *
 610                    ;****************************************
 611
 612 032D C410  ADD:    LDI     H(AESTK)        ;SET P3 TO CURRENT
 613 032F 37            XPAH    P3              ; STACK LOCATION
 614 0330 BAFD          DLD     LSTK(P2)
 615 0332 BAFD          DLD     LSTK(P2)
 616 0334 33            XPAL    P3
 617 0335 02            CCL
 618 0336 C3FE          LD      -2(P3)          ;REPLACE TWO TOP ITEMS
 619 0338 F300          ADD     0(P3)           ; ON STACK BY THEIR SUM
 620 033A CBFE          ST      -2(P3)
 621 033C C3FF          LD      -1(P3)
 622 033E F301          ADD     1(P3)
 623 0340 CBFF          ST      -1(P3)
 624 0342 90BE  X7:     JMP     X6A
 625
 626 0344 C410  SUB:    LDI     H(AESTK)        ;SET P3 TO CURRENT
 627 0346 37            XPAH    P3              ; STACK LOCATION
 628 0347 BAFD          DLD     LSTK(P2)
 629 0349 BAFD          DLD     LSTK(P2)
 630 034B 33            XPAL    P3
 631 034C 03            SCL
 632 034D C3FE          LD      -2(P3)          ;REPLACE TWO TOP ITEMS
 633 034F FB00          CAD     0(P3)           ; ON STACK BY THEIR
 634 0351 CBFE          ST      -2(P3)          ; DIFFERENCE
 635 0353 C3FF          LD      -1(P3)
 636 0355 FB01          CAD     1(P3)
 637 0357 CBFF          ST      -1(P3)
 638 0359 90A7          JMP     X6A
 639
 640
 641                    ;****************************************
 642                    ;*             NEGATE                   *
 643                    ;****************************************
 644
 645 035B C410  NEG:    LDI     H(AESTK)        ;SET P3 TO CURRENT
 646 035D 37            XPAH    P3              ; STACK LOCATION
 647 035E C2FD          LD      LSTK(P2)
 648 0360 33            XPAL    P3
```

```
 650 0361 03              SCL
 650 0362 C400            LDI     0
 651 0364 FBFE            CAD     -2(P3)          ;NEGATE TOP ITEM ON STACK
 652 0366 CBFE            ST      -2(P3)
 653 0368 C400            LDI     0
 654 036A FBFF            CAD     -1(P3)
 655 036C CBFF            ST      -1(P3)
 656 036E 90D2    X8:     JMP     X7
 657 0370 9092    E6:     JMP     E5
 658
 659
 660              ;****************************************
 661              ;*          MULTIPLY                    *
 662              ;****************************************
 663
 664                      .LOCAL
 665 0372 C410    MUL:    LDI     H(AESTK)        ;SET P3 TO CURRENT
 666 0374 37              XPAH    P3              ; STACK LOCATION
 667 0375 C2FD            LD      LSTK(P2)
 668 0377 33              XPAL    P3              ;DETERMINE SIGN OF PRODUCT,
 669 0378 C3FF            LD      -1(P3)          ; SAVE IN TEMP(P2)
 670 037A E3FD            XOR     -3(P3)
 671 037C CAEA            ST      TEMP(P2)
 672 037E C3FF            LD      -1(P3)          ;CHECK FOR NEGATIVE
 673 0380 940D            JP      $1              ; MULTIPLIER
 674 0382 03              SCL
 675 0383 C400            LDI     0               ;IF NEGATIVE,
 676 0385 FBFE            CAD     -2(P3)          ; NEGATE
 677 0387 CBFE            ST      -2(P3)
 678 0389 C400            LDI     0
 679 038B FBFF            CAD     -1(P3)
 680 038D CBFF            ST      -1(P3)
 681 038F C3FD    $1:     LD      -3(P3)          ;CHECK FOR NEGATIVE
 682 0391 940D            JP      $2              ; MULTIPLICAND
 683 0393 03              SCL
 684 0394 C400            LDI     0               ;IF NEGATIVE,
 685 0396 FBFC            CAD     -4(P3)          ; NEGATE
 686 0398 CBFC            ST      -4(P3)
 687 039A C400            LDI     0
 688 039C FBFD            CAD     -3(P3)
 689 039E CBFD            ST      -3(P3)
 690 03A0 C400    $2:     LDI     0               ;CLEAR WORKSPACE
 691 03A2 CB00            ST      0(P3)
 692 03A4 CB01            ST      1(P3)
 693 03A6 CB02            ST      2(P3)
 694 03A8 CB03            ST      3(P3)
 695 03AA C410            LDI     16              ;SET COUNTER TO 16
 696 03AC CAEB            ST      NUM(P2)
 697 03AE C3FF    $LOOP:  LD      -1(P3)          ;ROTATE MULTIPLIER
 698 03B0 1F              RRL                     ; RIGHT ONE BIT
 699 03B1 CBFF            ST      -1(P3)
 700 03B3 C3FE            LD      -2(P3)
 701 03B5 1F              RRL
 702 03B6 CBFE            ST      -2(P3)
```

```
 703 03B8 06                CSA                 ;CHECK FOR CARRY BIT
 704 03B9 9411              JP      $3          ;IF NOT SET, DON'T DO ADD
 705 03BB 02                CCL
 706 03BC C302              LD      2(P3)       ;ADD MULTIPLICAND
 707 03BE F3FC              ADD     -4(P3)      ;  INTO WORKSPACE
 708 03C0 CB02              ST      2(P3)
 709 03C2 C303              LD      3(P3)
 710 03C4 F3FD              ADD     -3(P3)
 711 03C6 CB03              ST      3(P3)
 712 03C8 9002              JMP     $3
 713 03CA 90A4    E6A:      JMP     E6
 714 03CC 02      $3:       CCL
 715 03CD C303              LD      3(P3)       ;SHIFT WORKSPACE RIGHT BY 1
 716 03CF 1F                RRL
 717 03D0 CB03              ST      3(P3)
 718 03D2 C302              LD      2(P3)
 719 03D4 1F                RRL
 720 03D5 CB02              ST      2(P3)
 721 03D7 C301              LD      1(P3)
 722 03D9 1F                RRL
 723 03DA CB01              ST      1(P3)
 724 03DC C300              LD      0(P3)
 725 03DE 1F                RRL
 726 03DF CB00              ST      0(P3)
 727 03E1 BAEB              DLD     NUM(P2)     ;DECREMENT COUNTER
 728 03E3 90C9              JNZ     $LOOP       ;LOOP IF NOT ZERO
 729 03E5 9002              JMP     $4
 730 03E7 9085    X9:       JMP     X8
 731 03E9 C2EA    $4:       LD      TEMP(P2)    ;CHECK SIGN WORD
 732 03EB 940D              JP      $EXIT       ;IF BIT7 = 1, NEGATE PRODUCT
 733 03ED 03                SCL
 734 03EE C400              LDI     0
 735 03F0 FB00              CAD     0(P3)
 736 03F2 CB00              ST      0(P3)
 737 03F4 C400              LDI     0
 738 03F6 FB01              CAD     1(P3)
 739 03F8 CB01              ST      1(P3)
 740 03FA C300    $EXIT:    LD      0(P3)       ;PUT PRODUCT ON TOP
 741 03FC CBFC              ST      -4(P3)      ;  OF STACK
 742 03FE C301              LD      1(P3)
 743 0400 CBFD              ST      -3(P3)
 744 0402 BAFD              DLD     LSTK(P2)    ;SUBTRACT 2 FROM
 745 0404 BAFD              DLD     LSTK(P2)    ;  LSTK
 746 0406 90DF              JMP     X9
 747
 748
 749                ;*****************************************
 750                ;*              DIVIDE                   *
 751                ;*****************************************
 752
 753                        .LOCAL
 754 0408 C410    DIV:      LDI     H(AESTK)
 755 040A 37                XPAH    P3
 756 040B C2FD              LD      LSTK(P2)
```

```
 7[illegible] 040D 33          XPAL    P3
 758 040E C3FF          LD      -1(P3)          ;CHECK FOR DIVISION BY 0
 759 0410 DBFE          OR      -2(P3)
 760 0412 9C04          JNZ     $0
 761 0414 C40D          LDI     13
 762 0416 90B2          JMP     E6A
 763 0418 C3FD  $0:     LD      -3(P3)
 764 041A E3FF          XOR     -1(P3)
 765 041C CAEA          ST      TEMP(P2)        ;SAVE SIGN OF QUOTIENT
 766 041E C3FD          LD      -3(P3)          ;IS DIVIDEND POSITIVE?
 767 0420 9411          JP      $POS            ;YES - JUMP
 768 0422 C400          LDI     0
 769 0424 03            SCL
 770 0425 FBFC          CAD     -4(P3)          ;NO - NEGATE DIVIDEND,
 771 0427 CB03          ST      3(P3)           ; STORE IN RIGHT HALF
 772 0429 C400          LDI     0               ; OF 32-BIT ACCUMULATOR
 773 042B FBFD          CAD     -3(P3)
 774 042D CB02          ST      2(P3)
 7[illegible] 042F 900A          JMP     $1
 7[illegible] 0431 90B4  X9A:    JMP     X9
 777 0433 C3FD  $POS:   LD      -3(P3)          ;STORE NON-NEGATED DIVIDEND
 778 0435 CB02          ST      2(P3)           ; IN 32-BIT ACCUMULATOR
 779 0437 C3FC          LD      -4(P3)
 780 0439 CB03          ST      3(P3)
 781 043B C3FF  $1:     LD      -1(P3)          ;CHECK FOR NEGATIVE DIVISOR
 782 043D 940D          JP      $2
 783 043F C400          LDI     0               ;NEGATE DIVISOR
 784 0441 03            SCL
 785 0442 FBFE          CAD     -2(P3)
 786 0444 CBFE          ST      -2(P3)
 787 0446 C400          LDI     0
 788 0448 FBFF          CAD     -1(P3)
 789 044A CBFF          ST      -1(P3)
 790 044C C400  $2:     LDI     0               ;PUT ZERO IN:
 791 044E CB01          ST      1(P3)           ;  LEFT HALF OF 32-BIT ACC,
 792 0450 CB00          ST      0(P3)
 793 0452 CAEB          ST      NUM(P2)         ;  THE COUNTER, AND
 7[illegible] 0454 CBFD          ST      -3(P3)          ;  IN THE DIVIDEND, NOW USED
 7[illegible] 0456 CBFC          ST      -4(P3)          ;  STORE THE QUOTIENT
 796 0458 02    $LOOP:  CCL                     ;BEGIN MAIN DIVIDE LOOP:
 797 0459 C3FC          LD      -4(P3)          ;  SHIFT QUOTIENT LEFT,
 798 045B F3FC          ADD     -4(P3)
 799 045D CBFC          ST      -4(P3)
 800 045F C3FD          LD      -3(P3)
 801 0461 F3FD          ADD     -3(P3)
 802 0463 CBFD          ST      -3(P3)
 803 0465 02            CCL                     ;  SHIFT 32-BIT ACC LEFT,
 804 0466 C303          LD      3(P3)
 805 0468 F303          ADD     3(P3)
 806 046A CB03          ST      3(P3)
 807 046C C302          LD      2(P3)
 808 046E F302          ADD     2(P3)
 809 0470 CB02          ST      2(P3)
 810 0472 C301          LD      1(P3)
```

```
 0474 F301            ADD     1(P3)
 0476 CB01            ST      1(P3)
 0478 C300            LD      (P3)
 047A F300            ADD     (P3)
 047C CB00            ST      (P3)
 047E 03              SCL
 047F C301            LD      1(P3)           ;  SUBTRACT DIVISOR INTO
 0481 FBFE            CAD     -2(P3)          ;   LEFT HALF OF ACC,
 0483 CB01            ST      1(P3)
 0485 C300            LD      (P3)
 0487 FBFF            CAD     -1(P3)
 0489 CB00            ST      (P3)
 048B 9411            JP      $ENT1           ;  IF RESULT IS NEGATIVE,
 048D 02              CCL                     ;   RESTORE ORIGINAL CONTENTS
 048E C301            LD      1(P3)           ;   OF ACC BY ADDING DIVISOR
 0490 F3FE            ADD     -2(P3)
 0492 CB01            ST      1(P3)
 0494 C300            LD      (P3)
 0496 F3FF            ADD     -1(P3)
 0498 CB00            ST      (P3)
 049A 9008            JMP     $3
 049C 9093    X9B:    JMP     X9A
 049E C3FC    $ENT1:  LD      -4(P3)          ;  ELSE IF RESULT POSITIVE,
 04A0 DC01            ORI     1               ;   RECORD A 1 IN QUOTIENT
 04A2 CBFC            ST      -4(P3)          ;   W/O RESTORING THE ACC
 04A4 AAEB    $3:     ILD     NUM(P2)         ;  INCREMENT THE COUNTER
 04A6 E410            XRI     16              ;  ARE WE DONE?
 04A8 9CAE            JNZ     $LOOP           ;  LOOP IF NOT DONE
 04AA C2EA            LD      TEMP(P2)        ;CHECK THE QUOTIENT'S SIGN,
 04AC 940D            JP      $END            ; NEGATING IF NECESSARY
 04AE C400            LDI     0
 04B0 03              SCL
 04B1 FBFC            CAD     -4(P3)
 04B3 CBFC            ST      -4(P3)
 04B5 C400            LDI     0
 04B7 FBFD            CAD     -3(P3)
 04B9 CBFD            ST      -3(P3)
 04BB BAFD    $END:   DLD     LSTK(P2)        ;DECREMENT THE STACK POINTER,
 04BD BAFD            DLD     LSTK(P2)
 04BF 90DB            JMP     X9B             ; AND EXIT

              ;****************************************
              ;*            STORE VARIABLE            *
              ;****************************************

 04C1 C410    STORE:  LDI     H(AESTK)        ;SET P3 TO STACK
 04C3 37              XPAH    P3
 04C4 C2FD            LD      LSTK(P2)
 04C6 33              XPAL    P3
 04C7 C7FD            LD      @-3(P3)         ;GET VARIABLE INDEX
 04C9 01              XAE                     ;PUT IN E REG
 04CA C301            LD      1(P3)
 04CC CA80            ST      EREG(P2)        ;STORE LOWER 8 BITS
```

```
 8  04CE 02              CCL                     ; INTO VARIABLE
866 04CF 40              LDE                     ;INCREMENT INDEX
867 04D0 F401            ADI     1
868 04D2 01              XAE
869 04D3 C302            LD      2(P3)
870 04D5 CA80            ST      EREG(P2)        ;STORE UPPER 8 BITS
871 04D7 33              XPAL    P3              ; INTO VARIABLE
872 04D8 CAFD            ST      LSTK(P2)        ;UPDATE STACK POINTER
873 04DA C400    X10:    JS      P3,EXECIL
874
875
876              ;****************************************
877              ;*     TEST FOR VARIABLE IN TEXT        *
878              ;****************************************
879
880 04E1 C501    TSTVAR: LD      @1(P1)
881 04E3 E420            XRI     ' '             ;SLEW OFF SPACES
882 04E5 98FA            JZ      TSTVAR
 8  04E7 C1FF            LD      -1(P1)          ;GET CHARACTER IN QUESTION
884 04E9 03              SCL
885 04EA FC5B            CAI     'Z'+1           ;SUBTRACT 'Z'+1
886 04EC 9405            JP      $FAIL           ;NOT VARIABLE IF POSITIVE
887 04EE 03              SCL
888 04EF FCE6            CAI     'A'-'Z'-1       ;SUBTRACT 'A'
889 04F1 9412            JP      $MAYBE          ;IF POS, MAY BE VARIABLE
890 04F3 C5FF    $FAIL:  LD      @-1(P1)         ;BACKSPACE CURSOR
891 04F5 C2FB            LD      PCLOW(P2)       ;GET TEST-FAIL ADDRESS
892 04F7 33              XPAL    P3              ; FROM I.L. TABLE, PUT IT
893 04F8 C2FA            LD      PCHIGH(P2)      ; INTO I.L. PROGRAM COUNTER
894 04FA 37              XPAH    P3
895 04FB C300            LD      (P3)
896 04FD CAFA            ST      PCHIGH(P2)
897 04FF C301            LD      1(P3)
898 0501 CAFB            ST      PCLOW(P2)
899 0503 90D5            JMP     X10
900 0505 01      $MAYBE: XAE                     ;SAVE VALUE (0-25)
901 0506 C100            LD      (P1)            ;CHECK FOLLOWING CHAR
 9  0508 03              SCL                     ;MUST NOT BE A LETTER
903 0509 FC5B            CAI     'Z'+1           ; OTHERWISE WE'D BE LOOKING
904 050B 9405            JP      $OK             ; AT A KEYWORD, NOT VARIABLE
905 050D 03              SCL
906 050E FCE6            CAI     'A'-'Z'-1
907 0510 94E1            JP      $FAIL
908 0512 C410    $OK:    LDI     H(AESTK)        ;SET P3 TO CURRENT
909 0514 37              XPAH    P3              ; STACK LOCATION
910 0515 AAFD            ILD     LSTK(P2)        ;INCR STACK POINTER
911 0517 33              XPAL    P3
912 0518 02              CCL                     ;DOUBLE VARIABLE INDEX
913 0519 40              LDE
914 051A 70              ADE
915 051B CBFF            ST      -1(P3)          ;PUT INDEX ON STACK
916 051D C402            LDI     2               ;INCREMENT I.L. PC, SKIPPING
917 051F 02              CCL                     ; OVER TEST-FAIL ADDRESS
918 0520 F2FB            ADD     PCLOW(P2)
```

```
                        ST      PCLOW(P2)
0524 C400               LDI     0
0526 F2FA               ADD     PCHIGH(P2)
0528 CAFA               ST      PCHIGH(P2)
052A 90AE               JMP     X10

                ;*****************************************
                ;*     IND -- EVALUATE A VARIABLE        *
                ;*****************************************

052C C410       IND:    LDI     H(AESTK)        ;SET P3 TO STACK
052E 37                 XPAH    P3
052F AAFD               ILD     LSTK(P2)
0531 33                 XPAL    P3
0532 C3FE               LD      -2(P3)          ;GET INDEX OFF TOP
0534 01                 XAE                     ;PUT INDEX IN E REG
0535 C280               LD      EREG(P2)        ;GET LOWER 8 BITS
0537 CBFE               ST      -2(P3)          ;SAVE ON STACK
0539 02                 CCL
053A 40                 LDE                     ;INCREMENT E REG
053B F401               ADI     1
053D 01                 XAE
053E C280               LD      EREG(P2)        ;GET UPPER 8 BITS
0540 CBFF               ST      -1(P3)          ;SAVE ON STACK
0542 9096       X11:    JMP     X10

                ;*****************************************
                ;*     RELATIONAL OPERATORS              *
                ;*****************************************

0544 C401       EQ:     LDI     1               ;EACH RELATIONAL OPERATOR
0546 9012               JMP     CMP             ; LOADS A NUMBER USED LATER
0548 C402       NEQ:    LDI     2               ; AS A CASE SELECTOR, AFTER
054A 900E               JMP     CMP             ; THE TWO OPERANDS ARE COM-
054C C403       LSS:    LDI     3               ; PARED.  BASED ON THE COM-
054E 900A               JMP     CMP             ; PARISON, FLAGS ARE SET THA
0550 C404       LEQ:    LDI     4               ; ARE EQUIVALENT TO THOSE SE
0552 9006               JMP     CMP             ; BY THE 'CMP' INSTRUCTION I
0554 C405       GTR:    LDI     5               ; THE PDP-11.  THESE PSEUDO-
0556 9002               JMP     CMP             ; FLAGS ARE USED TO DETERMIN
0558 C406       GEQ:    LDI     6               ; WHETHER THE PARTICULAR
                                                ; RELATION IS SATISFIED OR N

055A CAEB       CMP:    ST      NUM(P2)
055C C410               LDI     H(AESTK)        ;SET P3 -> ARITH STACK
055E 37                 XPAH    P3
055F BAFD               DLD     LSTK(P2)
0561 BAFD               DLD     LSTK(P2)
0563 33                 XPAL    P3
0564 03                 SCL
0565 C3FE               LD      -2(P3)          ;SUBTRACT THE TWO OPERANDS,
0567 FB00               CAD     (P3)            ; STORING RESULT IN LO & HI
0569 CAEF               ST      LO(P2)
```

```
SC/MP ASSEMBLER REV-A
NIBL  12/17/76

 973 056B C3FF          LD      -1(P3)
 974 056D FB01          CAD     1(P3)
 975 056F CAEE          ST      HI(P2)
 976 0571 E3FF          XOR     -1(P3)          ;OVERFLOW OCCURS IF SIGNS OF
 977 0573 01            XAE                     ; RESULT AND 1ST OPERAND
 978 0574 C3FF          LD      -1(P3)          ; DIFFER, AND SIGNS OF THE
 979 0576 E301          XOR     1(P3)           ; TWO OPERANDS DIFFER
 980 0578 50            ANE                     ;BIT 7 EQUIVALENT TO V FLAG
 981 0579 E2EE          XOR     HI(P2)          ;BIT 7 EQUIVALENT TO N XOR V
 982 057B CAEA          ST      TEMP(P2)        ;STORE IN TEMP
 983 057D C2EE          LD      HI(P2)          ;DETERMINE IF RESULT WAS ZER
 984 057F DAEF          OR      LO(P2)
 985 0581 9802          JZ      SETZ            ;IF RESULT=0, SET Z FLAG
 986 0583 C480          LDI     080             ; ELSE CLEAR Z FLAG
 987 0585 E480  SETZ:   XRI     080
 988 0587 01            XAE                     ;BIT 7 OF EX = Z FLAG
 989
 990 0588 BAEB          DLD     NUM(P2)         ;TEST FOR =
 991 058A 9C05          JNZ     NEQ1
 992 058C 40            LDE                     ; EQUAL IF Z = 1
 993 058D 902B          JMP     CMP1
 994 058F 90B1  X12:    JMP     X11
 995 0591 BAEB  NEQ1:   DLD     NUM(P2)         ;TEST FOR <>
 996 0593 9C05          JNZ     LSS1
 997 0595 40            LDE                     ; NOT EQUAL IF Z = 0
 998 0596 E480          XRI     080
 999 0598 9020          JMP     CMP1
1000 059A BAEB  LSS1:   DLD     NUM(P2)         ;TEST FOR <
1001 059C 9C04          JNZ     LEQ1
1002 059E C2EA          LD      TEMP(P2)        ; LESS THAN IF (N XOR V)=1
1003 05A0 9018          JMP     CMP1
1004 05A2 BAEB  LEQ1:   DLD     NUM(P2)         ;TEST FOR <=
1005 05A4 9C05          JNZ     GTR1
1006 05A6 40            LDE                     ; LESS THAN OR EQUAL
1007 05A7 DAEA          OR      TEMP(P2)        ;   IF (Z OR (N XOR V))=1
1008 05A9 900F          JMP     CMP1
1009 05AB BAEB  GTR1:   DLD     NUM(P2)         ;TEST FOR >
1010 05AD 9C07          JNZ     GEQ1
1011 05AF 40            LDE                     ; GREATER THAN
1012 05B0 DAEA          OR      TEMP(P2)        ;   IF (Z OR (N XOR V))=0
1013 05B2 E480          XRI     080
1014 05B4 9004          JMP     CMP1
1015 05B6 C2EA  GEQ1:   LD      TEMP(P2)        ;GREATER THAN OR EQUAL
1016 05B8 E480          XRI     080             ; IF (N XOR V)=0
1017 05BA 9404  CMP1:   JP      FALSE           ;IS RELATION SATISFIED?
1018 05BC C401          LDI     1               ;YES - PUSH 1 ON STACK
1019 05BE 9002          JMP     CMP2
1020 05C0 C400  FALSE:  LDI     0               ;NO - PUSH 0 ON STACK
1021 05C2 CBFE  CMP2:   ST      -2(P3)
1022 05C4 C400          LDI     0
1023 05C6 CBFF          ST      -1(P3)
1024 05C8 C400          JS      P3,RTN          ;DO AN I.L. RETURN
1025 05CF 90BE          JMP     X12
1026
```

```
1027                ;*****************************************
1028                ;*    IF STATEMENT TEST FOR ZERO         *
1029                ;*****************************************
1030
1031
1032 05D1 C2EF     CMPR:   LD      LO(P2)          ;GET LOW & HI BYTES OF EXPR.
1033 05D3 DAEE             OR      HI(P2)          ;TEST IF EXPRESSION IS ZERO
1034 05D5 9802             JZ      FAIL            ;YES - IT IS
1035 05D7 90B6             JMP     X12             ;NO - IT ISN'T SO CONTINUE
1036 05D9 C501     FAIL:   LD      @1(P1)          ;SKIP TO NEXT LINE IN PROGRAM
1037 05DB E40D             XRI     0D              ; (I.E. TIL NEXT CR)
1038 05DD 9CFA             JNZ     FAIL
1039 05DF C402             JS      P3,NXT          ;CALL NXT AND RETURN
1040 05E6 90A7     X12A:   JMP     X12
1041
1042
1043                ;*****************************************
1044                ;*         AND, OR, & NOT                *
1045                ;*****************************************
1046
1047                        .LOCAL
1048 05E8 C401     ANDOP:  LDI     1               ;EACH OPERATION HAS ITS
1049 05EA 9006             JMP     $1              ; OWN CASE SELECTOR.
1050 05EC C402     OROP:   LDI     2
1051 05EE 9002             JMP     $1
1052 05F0 C403     NOTOP:  LDI     3
1053 05F2 CAEB     $1:     ST      NUM(P2)
1054 05F4 C410             LDI     H(AESTK)        ;SET P3 -> ARITH. STACK
1055 05F6 37               XPAH    P3
1056 05F7 BAFD             DLD     LSTK(P2)
1057 05F9 BAFD             DLD     LSTK(P2)
1058 05FB 33               XPAL    P3
1059 05FC BAEB             DLD     NUM(P2)         ;TEST FOR 'AND'
1060 05FE 9C0E             JNZ     $OR
1061 0600 C301             LD      1(P3)           ;REPLACE TWO TOP ITEMS ON
1062 0602 D3FF             AND     -1(P3)          ; STACK BY THEIR 'AND'
1063 0604 CBFF             ST      -1(P3)
1064 0606 C300             LD      0(P3)
1065 0608 D3FE             AND     -2(P3)
1066 060A CBFE             ST      -2(P3)
1067 060C 90D8             JMP     X12A
1068 060E BAEB     $OR:    DLD     NUM(P2)         ;TEST FOR 'OR'
1069 0610 9C0E             JNZ     $NOT
1070 0612 C301             LD      1(P3)           ;REPLACE TWO TOP ITEMS ON
1071 0614 DBFF             OR      -1(P3)          ; STACK BY THEIR 'OR'
1072 0616 CBFF             ST      -1(P3)
1073 0618 C300             LD      0(P3)
1074 061A DBFE             OR      -2(P3)
1075 061C CBFE             ST      -2(P3)
1076 061E 90C6             JMP     X12A
1077 0620 C701     $NOT:   LD      @1(P3)          ;'NOT' OPERATION
1078 0622 E4FF             XRI     0FF
1079 0624 CBFF             ST      -1(P3)          ;REPLACE TOP ITEM ON STACK
1080 0626 C701             LD      @1(P3)          ; BY ITS ONE'S COMPLEMENT
```

```
1081 0628 E4FF            XRI     OFF
1082 062A CBFF            ST      -1(P3)
1083 062C 33              XPAL    P3
1084 062D CAFD            ST      LSTK(P2)          ;STACK POINTER FIXUP
1085 062F 90B5    X12B:   JMP     X12A
1086
1087
1088                      ;*****************************************
1089                      ;*      EXCHANGE CURSOR WITH RAM         *
1090                      ;*****************************************
1091
1092 0631 C2F1    XCHGP1: LD      P1LOW(P2)         ;THIS ROUTINE IS HANDY WHEN
1093 0633 31              XPAL    P1                ; EXECUTING AN 'INPUT' STMT
1094 0634 CAF1            ST      P1LOW(P2)         ; IT EXCHANGES THE CURRENT
1095 0636 C2F0            LD      P1HIGH(P2)        ; TEXT CURSOR WITH ONE SAVED
1096 0638 35              XPAH    P1                ; IN RAM
1097 0639 CAF0            ST      P1HIGH(P2)
1098 063B 3F              XPPC    P3
1099
1100
1101                      ;*****************************************
1102                      ;*           CHECK RUN MODE              *
1103                      ;*****************************************
1104
1105 063C C2F4    CKMODE: LD      RUNMOD(P2)        ;THIS ROUTINE CAUSES AN ERR
1106 063E 9801            JZ      CK1               ; IF CURRENTLY IN EDIT MODE
1107 0640 3F              XPPC    P3
1108 0641 C403    CK1:    LDI     3
1109 0643 CAEB    E8:     ST      NUM(P2)           ;ERROR IF RUN MODE = 0
1110 0645 C402            JS      P3,ERR2           ;MINOR KLUGE
1111
1112
1113                      ;*****************************************
1114                      ;*       GET HEXADECIMAL NUMBER          *
1115                      ;*****************************************
1116
1117                      .LOCAL
1118 064C AAFD    HEX:    ILD     LSTK(P2)          ;POINT P3 AT ARITH STACK
1119 064E AAFD            ILD     LSTK(P2)
1120 0650 33              XPAL    P3
1121 0651 C410            LDI     H(AESTK)
1122 0653 37              XPAH    P3
1123 0654 C400            LDI     0                 ;NUMBER INITIALLY ZERO
1124 0656 CBFF            ST      -1(P3)            ;PUT IT ON STACK
1125 0658 CBFE            ST      -2(P3)
1126 065A CAEB            ST      NUM(P2)           ;ZERO NUMBER OF DIGITS
1127 065C C501    $SKIP:  LD      @1(P1)            ;SKIP ANY SPACES
1128 065E E420            XRI     ' '
1129 0660 98FA            JZ      $SKIP
1130 0662 C5FF            LD      @-1(P1)
1131 0664 C100    $LOOP:  LD      (P1)              ;GET A CHARACTER
1132 0666 03              SCL
1133 0667 FC3A            CAI     '9'+1             ;CHECK FOR A NUMERIC CHAR
1134 0669 9409            JP      $LETR
```

```
1135 066B 03              SCL
1136 066C FCF6            CAI     '0'-'9'-1       ;IF NUMERIC, SHIFT NUMBER
1137 066E 9413            JP      $ENTER          ; AND ADD NEW HEX DIGIT
1138 0670 9032            JMP     $END
1139 0672 90BB  X12C:     JMP     X12B
1140 0674 03    $LETR:    SCL                     ;CHECK FOR HEX LETTER
1141 0675 FC0D            CAI     'G'-'9'-1
1142 0677 942B            JP      $END
1143 0679 03              SCL
1144 067A FCFA            CAI     'A'-'G'
1145 067C 9402            JP      $OK
1146 067E 9024            JMP     $END
1147 0680 02    $OK:      CCL                     ;ADD 10 TO GET TRUE VALUE
 148 0681 F40A            ADI     10              ; OF LETTER
1149 0683 01    $ENTER:   XAE                     ;NEW DIGIT IN EX REG
1150 0684 C404            LDI     4               ;SET SHIFT COUNTER
1151 0686 CAEA            ST      TEMP(P2)
1152 0688 CAEB            ST      NUM(P2)         ;DIGIT COUNT IS NON-ZERO
1153 068A C3FE  $SHIFT:   LD      -2(P3)          ;SHIFT NUMBER LEFT BY 4
1154 068C 02              CCL
1155 068D F3FE            ADD     -2(P3)
1156 068F CBFE            ST      -2(P3)
1157 0691 C3FF            LD      -1(P3)
1158 0693 F3FF            ADD     -1(P3)
1159 0695 CBFF            ST      -1(P3)
1160 0697 BAEA            DLD     TEMP(P2)
1161 0699 9CEF            JNZ     $SHIFT
1162 069B C3FE            LD      -2(P3)          ;ADD NEW DIGIT
1163 069D 58              ORE                     ; INTO NUMBER
1164 069E CBFE            ST      -2(P3)
1165 06A0 C501            LD      @1(P1)          ;ADVANCE THE CURSOR
1166 06A2 90C0            JMP     $LOOP           ;GET NEXT CHAR
 167 06A4 C2EB  $END:     LD      NUM(P2)         ;CHECK IF THERE WERE
1168 06A6 9C87            JNZ     X12B            ; MORE THAN 0 CHARACTERS
1169 06A8 C405            LDI     5               ;ERROR IF THERE WERE NONE
1170 06AA 9097  E8B:      JMP     E8
1171
1172
1173                      ;****************************************
1174                      ;*      TEST FOR NUMBER IN TEXT         *
1175                      ;****************************************
1176
1177                      ; THIS ROUTINE TESTS FOR A NUMBER IN THE TEXT.  IF NO
1178                      ; NUMBER IS FOUND, I.L. CONTROL PASSES TO THE ADDRESS
1179                      ; INDICATED IN THE 'TSTN' INSTRUCTION.  OTHERWISE, THE
1180                      ; NUMBER IS SCANNED AND PUT ON THE ARITHMETIC STACK,
1181                      ; WITH I.L. CONTROL PASSING TO THE NEXT INSTRUCTION.
1182
1183                      .LOCAL
1184 06AC C501  TSTNUM:   LD      @1(P1)
1185 06AE E420            XRI     ' '             ;SKIP OVER ANY SPACES
1186 06B0 98FA            JZ      TSTNUM
1187 06B2 C5FF            LD      @-1(P1)         ;GET FIRST CHAR
1188 06B4 03              SCL                     ;TEST FOR DIGIT
```

```
1189 06B5 FC3A          CAI     '9'+1
1190 06B7 9405          JP      $ABORT
1191 06B9 03            SCL
1192 06BA FCF6          CAI     '0'-'9'-1
1193 06BC 9421          JP      $1
1194 06BE C2FB  $ABORT: LD      PCLOW(P2)       ;GET TEST-FAIL ADDRESS
1195 06C0 33            XPAL    P3              ; FROM I.L. TABLE
1196 06C1 C2FA          LD      PCHIGH(P2)
1197 06C3 37            XPAH    P3
1198 06C4 C300          LD      (P3)            ;PUT TEST-FAIL ADDRESS
1199 06C6 CAFA          ST      PCHIGH(P2)      ; INTO I.L. PC
1200 06C8 C301          LD      1(P3)
1201 06CA CAFB          ST      PCLOW(P2)
1202 06CC 90A4          JMP     X12C
1203 06CE C402  $RET:   LDI     2               ;SKIP OVER ONE IL INSTRUCTION
1204 06D0 02            CCL                     ; IF NUMBER IS DONE
1205 06D1 F2FB          ADD     PCLOW(P2)
1206 06D3 CAFB          ST      PCLOW(P2)
1207 06D5 C400          LDI     0
1208 06D7 F2FA          ADD     PCHIGH(P2)
1209 06D9 CAFA          ST      PCHIGH(P2)
1210 06DB 9095  X13:    JMP     X12C
1211 06DD 90CB  E8A:    JMP     E8B
1212 06DF 01    $1:     XAE                     ;SAVE DIGIT IN EX REG
1213 06E0 C410          LDI     H(AESTK)        ;POINT P3 AT AE STACK
1214 06E2 37            XPAH    P3
1215 06E3 AAFD          ILD     LSTK(P2)
1216 06E5 AAFD          ILD     LSTK(P2)
1217 06E7 33            XPAL    P3
1218 06E8 C400          LDI     0
1219 06EA CBFF          ST      -1(P3)
1220 06EC 40            LDE
1221 06ED CBFE          ST      -2(P3)
1222 06EF C501  $LOOP:  LD      @1(P1)          ;GET NEXT CHAR
1223 06F1 C100          LD      (P1)
1224 06F3 03            SCL                     ;TEST IF IT IS DIGIT
1225 06F4 FC3A          CAI     '9'+1
1226 06F6 94D6          JP      $RET            ;RETURN IF IT ISN'T
1227 06F8 03            SCL
1228 06F9 FCF6          CAI     '0'-'9'-1
1229 06FB 9402          JP      $2
1230 06FD 90CF          JMP     $RET
1231 06FF 01    $2:     XAE                     ;SAVE DIGIT
1232 0700 C3FF          LD      -1(P3)          ;PUT RESULT IN SCRATCH SPACE
1233 0702 CB01          ST      1(P3)
1234 0704 C3FE          LD      -2(P3)
1235 0706 CB00          ST      (P3)
1236 0708 C402          LDI     2
1237 070A CAEA          ST      TEMP(P2)        ;MULTIPLY RESULT BY 10
1238 070C 02    $SHIFT: CCL                     ;FIRST MULTIPLY BY 4
1239 070D C3FE          LD      -2(P3)
1240 070F F3FE          ADD     -2(P3)
1241 0711 CBFE          ST      -2(P3)
1242 0713 C3FF          LD      -1(P3)
```

```
1243 0715 F3FF            ADD     -1(P3)
1244 0717 CBFF            ST      -1(P3)
1245 0719 D480            ANI     080             ';MAKE SURE NO OVERFLOW
1246 071B 9C34            JNZ     $ERR            ; OCCURRED
1247 071D BAEA            DLD     TEMP(P2)
1248 071F 9CEB            JNZ     $SHIFT
1249 0721 02              CCL                     ;THEN ADD OLD RESULT,
1250 0722 C3FE            LD      -2(P3)          ; SO WE HAVE RESULT * 5
1251 0724 F300            ADD     (P3)
1252 0726 CBFE            ST      -2(P3)
1253 0728 C3FF            LD      -1(P3)
1254 072A F301            ADD     1(P3)
1255 072C CBFF            ST      -1(P3)
1256 072E D480            ANI     080             ;MAKE SURE NO OVERFLOW
1257 0730 9C1F            JNZ     $ERR            ; OCCURRED
1258 0732 02              CCL                     ;THEN MULTIPLY BY TWO
1259 0733 C3FE            LD      -2(P3)
1260 0735 F3FE            ADD     -2(P3)
1261 0737 CBFE            ST      -2(P3)
1262 0739 C3FF            LD      -1(P3)
1263 073B F3FF            ADD     -1(P3)
1264 073D CBFF            ST      -1(P3)
1265 073F D480            ANI     080             ;MAKE SURE NO OVERFLOW
1266 0741 9C0E            JNZ     $ERR            ; OCCURRED
1267 0743 02              CCL                     ;THEN ADD IN NEW DIGIT
1268 0744 40              LDE
1269 0745 F3FE            ADD     -2(P3)
1270 0747 CBFE            ST      -2(P3)
1271 0749 C400            LDI     0
1272 074B F3FF            ADD     -1(P3)
1273 074D CBFF            ST      -1(P3)
1274 074F 949E            JP      $LOOP           ;REPEAT IF NO OVERFLOW
1275 0751 C406  $ERR:     LDI     6               ;ELSE REPORT ERROR
1276 0753 9088  E9:       JMP     E8A
1277 0755 9084  X14:      JMP     X13
1278
1279
1280                      ;****************************************
1281                      ;*     GET LINE FROM TELETYPE           *
1282                      ;****************************************
1283
1284                      .LOCAL
1285 0757        GETL:    LDPI    P1,LBUF         ;SET P1 TO LBUF
1286 0759 C400            LDI     0               ;CLEAR NO. OF CHAR
1287 075B CAE7            ST      CHRNUM(P2)
1288 075D                 LDPI    P3,PUTC-1       ;POINT P3 AT PUTC ROUTINE
1289 075F C2F4            LD      RUNMOD(P2)      ;PRINT '? ' IF RUNNING
1290 0761 9808            JZ      $0              ; (I.E. DURING 'INPUT')
1291 0763 C43F            LDI     '?'
1292 0765 3F              XPPC    P3
1293 0766 C420            LDI     ' '
1294 0768 3F              XPPC    P3
1295 0769 9003            JMP     $1
1296 076B C43E  $0:       LDI     '>'             ;OTHERWISE PRINT '>'
```

```
 1297 0775 3F              XPPC    P3
1298 0776 C40F   $1:       JS      P3,GECO          ;GET CHARACTER
 1299 077D C4C1            LDI     L(PUTC)-1        ;POINT P3 AT PUTC AGAIN
 1300 077F 33              XPAL    P3
1301 0780 40               LDE                      ;GET TYPED CHAR
 1302 0781 98F3            JZ      $1               ;IGNORE NULLS
 1303 0783 E40A            XRI     0A               ;IGNORE LINE FEED
 1304 0785 98EF            JZ      $1
 1305 0787 40              LDE
 1306 0788 E40D            XRI     0D               ;CHECK FOR CR
 1307 078A 9850            JZ      $CR
 1308 078C 40              LDE
  309 078D E45F            XRI     'O'+010          ;CHECK FOR SHIFT/O
 1310 078F 9841            JZ      $RUB
 1311 0791 40              LDE                      ;CHECK FOR CTRL/H
 1312 0792 E408            XRI     8
 1313 0794 9836            JZ      $XH
 1314 0796 40              LDE
 1315 0797 E415            XRI     015              ;CHECK FOR CTRL/U
 1316 0799 980F            JZ      $XU
 1317 079B 40              LDE
 1318 079C E403            XRI     3                ;CHECK FOR CTRL/C
 1319 079E 9C1A            JNZ     $ENTER
 1320 07A0 C45E            LDI     '^'              ;ECHO CONTROL/C AS ^C
1321 07A2 3F               XPPC    P3
 1322 07A3 C443            LDI     'C'
 1323 07A5 3F              XPPC    P3
1324 07A6 C40E             LDI     14               ;CAUSE A BREAK
 1325 07A8 90A9            JMP     E9
 1326 07AA C45E   $XU:     LDI     '^'              ;ECHO CONTROL/U AS ^U
 1327 07AC 3F              XPPC    P3
 1328 07AD C455            LDI     'U'
 1329 07AF 3F              XPPC    P3
 1330 07B0 C40D            LDI     0D               ;PRINT CR/LF
 1331 07B2 3F              XPPC    P3
 1332 07B3 C40A            LDI     0A
 1333 07B5 3F              XPPC    P3
 1334 07B6 909F   $2:      JMP     GETL             ;GO GET ANOTHER LINE
 1335 07B8 909B   X15:     JMP     X14
 1336 07BA 40     $ENTER:  LDE
 1337 07BB CD01            ST      @1(P1)           ;PUT CHAR IN LBUF
 1338 07BD AAE7            ILD     CHRNUM(P2)       ;INCREMENT CHRNUM
 1339 07BF E448            XRI     72               ;IF=72, LINE FULL
 1340 07C1 9CB3            JNZ     $1
 1341 07C3 C40D            LDI     0D
 1342 07C5 01              XAE                      ;SAVE CARRIAGE RET
 1343 07C6 40              LDE
 1344 07C7 3F              XPPC    P3               ;PRINT IT
1345 07C8 9012             JMP     $CR              ;STORE IT IN LBUF
 1346 07CA 9087   E10:     JMP     E9
 1347 07CC C420   $XH:     LDI     ' '              ;BLANK OUT THE CHARACTER
1348 07CE 3F               XPPC    P3
 1349 07CF C408            LDI     8                ;PRINT ANOTHER BACKSPACE
 1350 07D1 3F              XPPC    P3
```

```
1350 07D2 C2E7   $RUB:   LD      CHRNUM(P2)
1352 07D4 98A0           JZ      $1
1353 07D6 BAE7           DLD     CHRNUM(P2)      ;ONE LESS CHAR
1354 07D8 C5FF           LD      @-1(P1)         ;BACKSPACE CURSOR
1355 07DA 909A           JMP     $1
1356 07DC 40     $CR:    LDE
1357 07DD CD01           ST      @1(P1)          ;STORE CR IN LBUF
1358 07DF C40A           LDI     0A              ;PRINT LINE FEED
1359 07E1 3F             XPPC    P3
1360 07E2 C410           LDI     H(LBUF)         ;SET P1 TO BEGIN-
1361 07E4 35             XPAH    P1              ; NING OF LBUF
1362 07E5 C4D6           LDI     L(LBUF)
 363 07E7 31             XPAL    P1
1364 07E8 90CE   X16:    JMP     X15
1365
1366
1367                     ;*************************************
1368                     ;*      EVAL -- GET MEMORY CONTENTS    *
1369                     ;*************************************
1370
1371                     ; THIS ROUTINE IMPLEMENTS THE '@' OPERATOR IN EXPRESSIONS
1372
1373 07EA C410   EVAL:   LDI     H(AESTK)
1374 07EC 37             XPAH    P3
1375 07ED C2FD           LD      LSTK(P2)
1376 07EF 33             XPAL    P3              ;P3 -> ARITH STACK
1377 07F0 C3FF           LD      -1(P3)          ;GET ADDR OFF STACK,
1378 07F2 35             XPAH    P1              ; AND INTO P1,
1379 07F3 01             XAE                     ; SAVING OLD P1 IN EX & LO
1380 07F4 C3FE           LD      -2(P3)
1381 07F6 31             XPAL    P1
1382 07F7 CAEF           ST      LO(P2)
1383 07F9 C100           LD      0(P1)           ;GET MEMORY CONTENTS,
1384 07FB CBFE           ST      -2(P3)          ; SHOVE ONTO STACK
1385 07FD C400           LDI     0
1386 07FF CBFF           ST      -1(P3)          ;HIGH ORDER 8 BITS ZEROED
1387 0801 C2EF           LD      LO(P2)
1388 0803 31             XPAL    P1              ;RESTORE ORIGINAL P1
1389 0804 40             LDE
1390 0805 35             XPAH    P1
1391 0806 90B0           JMP     X15
1392
1393
1394                     ;*************************************
1395                     ;*     MOVE -- STORE INTO MEMORY     *
1396                     ;*************************************
1397
1398                     ; THIS ROUTINE IMPLEMENTS THE STATEMENT:
1399                     ;     '@' FACTOR '=' REL-EXP
1400
1401 0808 C410   MOVE:   LDI     H(AESTK)
1402 080A 37             XPAH    P3
1403 080B C2FD           LD      LSTK(P2)
1404 080D 33             XPAL    P3              ;P3 -> ARITH STACK
```

```
1405 080E C7FE            LD      @-2(P3)         ;GET BYTE TO BE MOVED
1406 0810 01              XAE
1407 0811 C7FF            LD      @-1(P3)         ;NOW GET ADDRESS INTO P3
1408 0813 CAEA            ST      TEMP(P2)
1409 0815 C7FF            LD      @-1(P3)
1410 0817 33              XPAL    P3
1411 0818 CAFD            ST      LSTK(P2)        ;STACK PTR UPDATED NOW
1412 081A C2EA            LD      TEMP(P2)
1413 081C 37              XPAH    P3
1414 081D 40              LDE
1415 081E CB00            ST      0(P3)           ;MOVE THE BYTE INTO MEMORY
1416 0820 90C6    X17:    JMP     X16
1417 0822 90A6    E11:    JMP     E10
1418
1419                      ;*************************************
1420                      ;*              TEXT EDITOR          *
1421                      ;*************************************
1422
1423                      ;INPUTS TO THIS ROUTINE: POINTER TO LINE BUFFER IN P1LOW &
1424                      ; P1HIGH.  P1 POINTS TO THE INSERTION POINT IN THE TEXT.
1425                      ; THE A.E. STACK HAS THE LINE NUMBER ON IT (STACK POINTER
1426                      ; IS ALREADY POPPED).
1427
1428                      ;EACH LINE IN THE NIBL TEXT IS STORED IN THE FOLLOWING
1429                      ; FORMAT: TWO BYTES CONTAINING THE LINE NUMBER (IN BINARY,
1430                      ; HIGH ORDER BYTE FIRST), THEN ONE BYTE CONTAINING THE
1431                      ; LENGTH OF THE LINE, AND FINALLY THE LINE ITSELF FOLLOWED
1432                      ; BY A CARRIAGE RETURN.  THE LAST LINE IN THE TEXT IS
1433                      ; FOLLOWED BY TWO CONSECUTIVE BYTES OF X'FF.
1434
1435                      .LOCAL
1436 0824 C410    INSRT:  LDI     H(AESTK)        ;POINT P3 AT AE STACK,
1437 0826 37              XPAH    P3              ; WHICH HAS THE LINE #
1438 0827 C2FD            LD      LSTK(P2)        ; ON IT
1439 0829 33              XPAL    P3
1440 082A C301            LD      1(P3)           ;SAVE NEW LINE'S NUMBER
1441 082C CAF7            ST      HILINE(P2)
1442 082E C300            LD      0(P3)
1443 0830 CAF8            ST      LOLINE(P2)
1444 0832 C2F1            LD      P1LOW(P2)       ;PUT POINTER TO LBUF INTO P3
1445 0834 33              XPAL    P3
1446 0835 C2F0            LD      P1HIGH(P2)
1447 0837 37              XPAH    P3
1448 0838 C404            LDI     4               ;INITIALLY LENGTH OF NEW LIN
1449 083A CAE7            ST      CHRNUM(P2)      ; = 4.  ADD 1 TO LENGTH FOR
1450 083C C701    $1:     LD      @1(P3)          ; EACH CHAR IN LINE UP TO,
1451 083E E40D            XRI     0D              ; BUT NOT INCLUDING,
1452 0840 9804            JZ      $2              ; CARRIAGE RETURN
1453 0842 AAE7            ILD     CHRNUM(P2)
1454 0844 90F6            JMP     $1
1455 0846 C2E7    $2:     LD      CHRNUM(P2)      ;IF LENGTH STILL 4,
1456 0848 E404            XRI     4               ; WE'LL DELETE A LINE,
1457 084A 9C02            JNZ     $3              ; SO SET LENGTH = 0
1458 084C CAE7            ST      CHRNUM(P2)
```

```
 1459 084E C2E7   $3:     LD      CHRNUM(P2)      ;PUT NEW LINE LENGTH IN EX
 1460 0850 01             XAE
 1461 0851 C2F2           LD      LABLHI(P2)      ;IS NEW LINE REPLACING OLD
 1462 0853 9406           JP      $4              ;YES - DO REPLACE
 1463 0855 D47F           ANI     07F             ;NO - WE'LL INSERT LINE HE
 1464 0857 CAF2           ST      LABLHI(P2)      ; WHERE ENDLBL GOT US
 1465 0859 9018           JMP     $MOVE           ;BUT FIRST MAKE ROOM
 1466 085B C503   $4:     LD      @3(P1)          ;SKIP LINE # AND LENGTH
 1467 085D 40             LDE                     ;EX, NOW HOLDING NEW LINE
 1468 085E 02             CCL                     ; LENGTH, WILL SOON HOLD
 1469 085F F4FC           ADI     -4              ; DISPLACEMENT OF LINES
 1470 0861 01             XAE                     ; TO BE MOVED
 1471 0862 C501   $5:     LD      @1(P1)          ;SUBTRACT 1 FROM DISPLACEM
 1472 0864 E40D           XRI     0D              ; FOR EACH CHAR IN LINE BE
 1473 0866 980B           JZ      $MOVE           ; REPLACED
 1474 0868 40             LDE
 1475 0869 02             CCL
 1476 086A F4FF           ADI     -1
 1477 086C 01             XAE
 1478 086D 90F3           JMP     $5
 1479 086F 90AF   X19:    JMP     X17
 1480 0871 90AF   E12:    JMP     E11
 1481 0873 40     $MOVE:  LDE                     ;IF DISPLACEMENT AND LENGT
 1482 0874 DAE7           OR      CHRNUM(P2)      ; OF NEW LINE ARE 0, RETUR
 1483 0876 98F7           JZ      X19
 1484 0878 C47A           LDI     L(DOSTAK)       ;CLEAR SOME STACKS
 1485 087A CAFF           ST      DOPTR(P2)
 1486 087C C46A           LDI     L(SBRSTK)
 1487 087E CAFC           ST      SBRPTR(P2)
 1488 0880 C48A           LDI     L(FORSTK)
 1489 0882 CAFE           ST      FORPTR(P2)
 1490 0884 40             LDE
 1491 0885 9860           JZ      $ADD            ;DON'T NEED TO MOVE LINES
 1492 0887 9410           JP      $UP             ;SKIP IF DISP. POSITIVE
 1493 0889 C100   $DOWN:  LD      0(P1)           ;NEGATIVE DISPLACEMENT:
 1494 088B C980           ST      EREG(P1)        ;DO;
 1495 088D C501           LD      @1(P1)          ;    M(P1+DISP) = M(P1);
 1496 088F 94F8           JP      $DOWN           ;    P1 = P1+1;
 1497 0891 C100           LD      0(P1)           ;UNTIL M(P1)<0 & M(P1-1)<0
 1498 0893 94F4           JP      $DOWN
 1499 0895 C980           ST      EREG(P1)        ;M(P1+DISP) = M(P1);
 1500 0897 904E           JMP     $ADD
 1501 0899 C1FE   $UP:    LD      -2(P1)          ;POSITIVE DISPLACEMENT:
 1502 089B CACA           ST      TEMP(P2)        ;FLAG BEGINNING OF MOVE WI
 1503 089D C4FF           LDI     -1              ; A -1 FOLLOWED BY 80, WHI
 1504 089F C9FE           ST      -2(P1)          ; CAN NEVER APPEAR IN A
 1505 08A1 C450           LDI     80              ; NIBL TEXT
 1506 08A3 C9FF           ST      -1(P1)
 1507 08A5 C501   $UP1:   LD      @1(P1)          ;ADVANCE P1 TO END OF TEXT
 1508 08A7 94FC           JP      $UP1
 1509 08A9 C100           LD      0(P1)
 1510 08AB 94F8           JP      $UP1
 1511 08AD 35             XPAH    P1              ;SAVE P1 IN LO, HI
 1512 08AE CAEE           ST      HI(P2)
```

```
1513 08B0 35              XPAH    P1
1514 08B1 31              XPAL    P1
1515 08B2 CAEF            ST      LO(P2)
1516 08B4 31              XPAL    P1
1517 08B5 C2EF            LD      LO(P2)          ;ADD DISPLACEMENT TO
1518 08B7 02              CCL                     ; VALUE OF P1, TO CHECK
1519 08B8 70              ADE                     ; WHETHER WE'RE OUT OF
1520 08B9 C400            LDI     0               ; RAM FOR USER'S PROGRAM
1521 08BB F2EE            ADD     HI(P2)
1522 08BD E2EE            XOR     HI(P2)
1523 08BF D4F0            ANI     0F0
1524 08C1 9803            JZ      $UP2
1525 08C3 C400            LDI     0               ;IF OUT OF RAM, CHANGE
1526 08C5 01              XAE                     ; DISPLACEMENT TO ZERO
1527 08C6 C4FF    $UP2:   LDI     -1
1528 08C8 C980    $UP3:   ST      EREG(P1)        ;MOVE TEXT UP UNTIL WE REA
1529 08CA C5FF            LD      @-1(P1)         ; THE FLAGS SET ABOVE
1530 08CC 94FA            JP      $UP3
1531 08CE C101            LD      1(P1)
1532 08D0 E450            XRI     80
1533 08D2 9804            JZ      $UP4
1534 08D4 C100            LD      0(P1)
1535 08D6 90F0            JMP     $UP3
1536 08D8 C2EA    $UP4:   LD      TEMP(P2)        ;RESTORE THE FLAGGED LOCAT
1537 08DA C900            ST      0(P1)           ; TO THEIR ORIGINAL VALUES
1538 08DC C40D            LDI     0D
1539 08DE C901            ST      1(P1)
1540 08E0 40              LDE                     ;IF DISPLACEMENT = 0, WE'RE
1541 08E1 9C04            JNZ     $ADD            ; OUT OF RAM, SO REPORT ER
1542 08E3 C402            LDI     2
1543 08E5 908A    E12A:   JMP     E12
1544 08E7 C2E7    $ADD:   LD      CHRNUM(P2)      ;INSERT NEW LINE
1545 08E9 9884    X19A:   JZ      X19             ; UNLESS LENGTH IS ZERO
1546 08EB C2F1            LD      P1LOW(P2)       ;POINT P1 AT LINE BUFFER
1547 08ED 31              XPAL    P1
1548 08EE C2F0            LD      P1HIGH(P2)
1549 08F0 35              XPAH    P1
1550 08F1 C2F3            LD      LABLLO(P2)      ;POINT P3 AT INSERTION PLA
1551 08F3 33              XPAL    P3
1552 08F4 C2F2            LD      LABLHI(P2)
1553 08F6 37              XPAH    P3
1554 08F7 C2F7            LD      HILINE(P2)      ;PUT LINE NUMBER INTO TEXT
1555 08F9 CF01            ST      @1(P3)
1556 08FB C2F8            LD      LOLINE(P2)
1557 08FD CF01            ST      @1(P3)
1558 08FF C2E7            LD      CHRNUM(P2)      ;STORE LINE LENGTH IN TEXT
1559 0901 CF01            ST      @1(P3)
1560 0903 C501    $ADD1:  LD      @1(P1)          ;PUT REST OF CHARS
1561 0905 CF01            ST      @1(P3)          ; (INCLUDING CR) INTO TEXT
1562 0907 E40D            XRI     0D
1563 0909 9CF8            JNZ     $ADD1
1564 090B 90DC            JMP     X19A            ;RETURN
1565 090D C400    X20:    JS      P3,EXECIL
1566 0914 90CF    E13:    JMP     E12A
```

```
1567
1568
1569                ;****************************************
1570                ;*        POP ARITHMETIC STACK          *
1571                ;****************************************
1572
1573 0916 BAFD   POPAE:  DLD     LSTK(P2)        ;THIS ROUTINE POP THE A.E.
1574 0918 BAFD           DLD     LSTK(P2)        ; STACK, AND PUTS THE RESU
1575 091A 33             XPAL    P3              ; INTO LO(P2) AND HI(P2)
1576 091B C410           LDI     H(AESTK)
1577 091D 37             XPAH    P3
1578 091E C300           LD      (P3)
1579 0920 CAEF           ST      LO(P2)
1580 0922 C301           LD      1(P3)
1581 0924 CAEE           ST      HI(P2)
1582 0926 90E5           JMP     X20
1583
1584
1585                ;****************************************
1586                ;*                UNTIL                 *
1587                ;****************************************
1588
1589                      .LOCAL
1590 0928 C2FF   UNTIL:  LD      DOPTR(P2)       ;CHECK FOR DO-STACK UNDERF
1591 092A 01             XAE
1592 092B 40             LDE
1593 092C E47A           XRI     L(DOSTAK)
1594 092E 9C04           JNZ     $1
1595 0930 C40F           LDI     15
1596 0932 90E0           JMP     E13
1597 0934 C2EF   $1:     LD      LO(P2)          ;CHECK FOR EXPRESSION = 0
1598 0936 DAEE           OR      HI(P2)
1599 0938 9806           JZ      $REDO           ;IF ZERO, REPEAT DO-LOOP
1600 093A BAFF           DLD     DOPTR(P2)       ;ELSE POP SAVE STACK
1601 093C BAFF           DLD     DOPTR(P2)
1602 093E 90CD           JMP     X20             ;CONTINUE TO NEXT STMT
1603 0940 40     $REDO:  LDE                     ;POINT P3 AT DO-STACK
1604 0941 33             XPAL    P3
1605 0942 C410           LDI     H(DOSTAK)
1606 0944 37             XPAH    P3
1607 0945 C3FF           LD      -1(P3)          ;LOAD P1 FROM DO STACK
1608 0947 35             XPAH    P1
1609 0948 C3FE           LD      -2(P3)
1610 094A 31             XPAL    P1              ;CURSOR NOW POINTS TO FIR
1611 094B 90C0           JMP     X20             ; STATEMENT OF DO-LOOP
1612
1613
1614                ;****************************************
1615                ;*     STORE INTO STATUS REGISTER       *
1616                ;****************************************
1617
1618                ; THIS ROUTINE IMPLEMENTS THE STATEMENT:
1619                ;      'STAT' '=' REL-EXP
1620
```

```
1621 094D C2EF   MOVESR: LD      LO(P2)          ;LOW BYTE GOES TO STATUS
1622 094F D4F7           ANI     0F7             ;; BUT WITH IEN BIT CLEARED
1623 0951 07             CAS
1624 0952 90B9   X21:    JMP     X20
1625 0954 90BE   E14:    JMP     E13
1626
1627
1628                     ;****************************************
1629                     ;*          STAT FUNCTION               *
1630                     ;****************************************
1631
1632 0956 C410   STATUS: LDI     H(AESTK)
1633 0958 37             XPAH    P3              ;POINT P3 AT AE STACK
1634 0959 AAFD           ILD     LSTK(P2)
1635 095B AAFD           ILD     LSTK(P2)
1636 095D 33             XPAL    P3
1637 095E 06             CSA
1638 095F CBFE           ST      -2(P3)          ;STATUS REG IS LOW BYTE
1639 0961 C400           LDI     0
1640 0963 CBFF           ST      -1(P3)          ;ZERO IS HIGH BYTE
1641 0965 90EB           JMP     X21
1642
1643
1644                     ;****************************************
1645                     ;*    MACHINE LANGUAGE SUBROUTINE       *
1646                     ;****************************************
1647
1648                     ; THIS ROUTINE IMPLEMENTS THE 'LINK' STATEMENT
1649
1650 0967 C2EE   CALLML: LD      HI(P2)          ;GET HIGH BYTE OF ADDRESS
1651 0969 37             XPAH    P3
1652 096A C2EF           LD      LO(P2)          ;GET LOW BYTE
1653 096C 33             XPAL    P3              ;P3 -> USER'S ROUTINE
1654 096D C7FF           LD      @-1(P3)         ;CORRECT P3
1655 096F 3F             XPPC    P3              ;CALL ROUTINE (PRAY IT WORKS
1656 0970                LDPI    P2,VARS         ;RESTORE RAM POINTER
1657 0976 90DA           JMP     X21             ;RETURN
1658
1659
1660                     ;****************************************
1661                     ;*        SAVE DO LOOP ADDRESS          *
1662                     ;****************************************
1663
1664                     ; THIS ROUTINE IMPLEMENTS THE 'DO' STATEMENT.
1665
1666                      .LOCAL
1667 0978 C2FF   SAVEDO: LD      DOPTR(P2)       ;CHECK FOR STACK OVERFLOW
1668 097A E48A           XRI     L(FORSTK)
1669 097C 9C04           JNZ     $1
1670 097E C40A           LDI     10
1671 0980 90D2   E15:    JMP     E14
1672 0982 AAFF   $1:     ILD     DOPTR(P2)
1673 0984 AAFF           ILD     DOPTR(P2)
1674 0986 33             XPAL    P3
```

```
1675 0987 C410            LDI     H(DOSTAK)
1676 0989 37              XPAH    P3              ;P3 -> TOP OF DO STACK
1677 098A 35              XPAH    P1              ;SAVE CURSOR ON THE STACK
1678 098B CBFF            ST      -1(P3)
1679 098D 35              XPAH    P1
1680 098E 31              XPAL    P1
1681 098F CBFE            ST      -2(P3)
1682 0991 31              XPAL    P1
1683 0992 90BE    X22:    JMP     X21
1684
1685
1686                      ;****************************************
1687                      ;*         TOP OF RAM FUNCTION          *
1688                      ;****************************************
1689
1690                      .LOCAL
1691 0994 C2E9    TOP:    LD      TEMP2(P2)       ;SET P3 TO POINT TO
1692 0996 37              XPAH    P3              ; START OF NIBL TEXT
1693 0997 C2E8            LD      TEMP3(P2)
1694 0999 33              XPAL    P3
1695 099A C300    $0:     LD      (P3)            ;HAVE WE HIT END OF TEXT?
1696 099C 9402            JP      $1              ;NO - SKIP TO NEXT LINE
1697 099E 9007            JMP     $2              ;YES - PUT CURSOR ON STACK
1698 09A0 C302    $1:     LD      2(P3)           ;GET LENGTH OF LINE
1699 09A2 01              XAE
1700 09A3 C780            LD      @EREG(P3)       ;SKIP TO NEXT LINE
1701 09A5 90F3            JMP     $0              ;GO CHECK FOR EOF
1702 09A7 C702    $2:     LD      @2(P3)          ;P3 := P3 + 2
1703 09A9 AAFD            ILD     LSTK(P2)        ;SET P3 TO STACK, SAVING
1704 09AB AAFD            ILD     LSTK(P2)        ; OLD P3 (WHICH CONTAINS TOF
1705 09AD 33              XPAL    P3              ; ON IT SOMEHOW
1706 09AE 01              XAE
1707 09AF C410            LDI     H(AESTK)
1708 09B1 37              XPAH    P3
1709 09B2 CBFF            ST      -1(P3)
1710 09B4 40              LDE
1711 09B5 CBFE            ST      -2(P3)
1712 09B7 90D9            JMP     X22
1713
1714
1715                      ;****************************************
1716                      ;*       SKIP TO NEXT NIBL LINE         *
1717                      ;****************************************
1718
1719 09B9 C501    IGNORE: LD      @1(P1)          ;SCAN TIL WE'RE PAST
1720 09BB E40D            XRI     0D              ; CARRIAGE RETURN
1721 09BD 9CFA            JNZ     IGNORE
1722 09BF 3F              XPPC    P3
1723
1724
1725                      ;****************************************
1726                      ;*          MODULO FUNCTION             *
1727                      ;****************************************
1728
```

```
1729 09C0 C2FD   MODULO: LD      LSTK(P2)        ;THIS ROUTINE MUST BE
1730 09C2 33             XPAL    P3              ;  IMMEDIATELY AFTER A
1731 09C3 C410           LDI     H(AESTK)        ;  DIVIDE TO WORK CORRECTLY
1732 09C5 37             XPAH    P3
1733 09C6 C303           LD      3(P3)           ;GET LOW BYTE OF REMAINDER
1734 09C8 CBFE           ST      -2(P3)          ;PUT ON STACK
1735 09CA C302           LD      2(P3)           ;GET HIGH BYTE OF REMAINDER
1736 09CC CBFF           ST      -1(P3)          ;PUT ON STACK
1737 09CE 90C2   X23:    JMP     X22
1738 09D0 90AE   E16:    JMP     E15
1739
1740
1741             ;****************************************
1742             ;*           RANDOM FUNCTION            *
1743             ;****************************************
1744
1745                      .LOCAL
1746 09D2 C408   RANDOM: LDI     8               ;LOOP COUNTER FOR MULTIPLY
1747 09D4 CAEB           ST      NUM(P2)
1748 09D6 C2E5           LD      RNDX(P2)
1749 09D8 01             XAE
1750 09D9 C2E4           LD      RNDY(P2)
1751 09DB CAE9           ST      TEMP2(P2)
1752 09DD C2E5   $LOOP:  LD      RNDX(P2)        ;MULTIPLY THE SEEDS BY 9
1753 09DF 02             CCL
1754 09E0 70             ADE
1755 09E1 01             XAE
1756 09E2 C2E4           LD      RNDY(P2)
1757 09E4 02             CCL
1758 09E5 F2E9           ADD     TEMP2(P2)
1759 09E7 CAE4           ST      RNDY(P2)
1760 09E9 BAEB           DLD     NUM(P2)
1761 09EB 9CF0           JNZ     $LOOP
1762 09ED 40             LDE                     ;ADD 7 TO SEEDS
1763 09EE 02             CCL
1764 09EF F407           ADI     7
1765 09F1 01             XAE
1766 09F2 C2E4           LD      RNDY(P2)
1767 09F4 02             CCL
1768 09F5 F407           ADI     7
1769 09F7 1E             RR
1770 09F8 CAE4           ST      RNDY(P2)
1771 09FA AAE6           ILD     RNDF(P2)        ;HAVE WE GONE THROUGH
1772 09FC 9803           JZ      $1              ;  256 GENERATIONS?
1773 09FE 40             LDE                     ;IF SO, SKIP GENERATING
1774 09FF CAE5           ST      RNDX(P2)        ;  THE NEW RNDX
1775 0A01 C2FD   $1:     LD      LSTK(P2)        ;START MESSING WITH THE STA
1776 0A03 33             XPAL    P3
1777 0A04 C410           LDI     H(AESTK)
1778 0A06 37             XPAH    P3
1779 0A07 C401           LDI     1               ;FIRST PUT 1 ON STACK
1780 0A09 CB00           ST      (P3)
1781 0A0B C400           LDI     0
1782 0A0D CB01           ST      1(P3)
```

```
1783 0A0F C3FE            LD      -2(P3)          ;PUT EXPR2 ON STACK
1784 0A11 CB02            ST      2(P3)
1785 0A13 C3FF            LD      -1(P3)
1786 0A15 CB03            ST      3(P3)
1787 0A17 C3FC            LD      -4(P3)          ;PUT EXPR1 ON STACK
1788 0A19 CB04            ST      4(P3)
1789 0A1B C3FD            LD      -3(P3)
1790 0A1D CB05            ST      5(P3)
1791 0A1F C2E4            LD      RNDY(P2)        ;PUT RANDOM # ON STACK
1792 0A21 CBFE            ST      -2(P3)
1793 0A23 C2E5            LD      RNDX(P2)
1794 0A25 E4FF            XRI     0FF
1795 0A27 D47F            ANI     07F
1796 0A29 CBFF            ST      -1(P3)
1797 0A2B C706            LD      @6(P3)          ;ADD 6 TO STACK POINTER
1798 0A2D 33              XPAL    P3
1799 0A2E CAFD            ST      LSTK(P2)
1800 0A30 909C    X24:    JMP     X23
1801 0A32 909C    E16A:   JMP     E16
1802
1803
1804                      ;****************************************
1805                      ;*      PUSH 1 ON ARITHMETIC STACK      *
1806                      ;****************************************
1807
1808 0A34 AAFD    LIT1:   ILD     LSTK(P2)
1809 0A36 AAFD            ILD     LSTK(P2)
1810 0A38 33              XPAL    P3
1811 0A39 C410            LDI     H(AESTK)
1812 0A3B 37              XPAH    P3
1813 0A3C C400            LDI     0
1814 0A3E CBFF            ST      -1(P3)
1815 0A40 C401            LDI     1
1816 0A42 CBFE            ST      -2(P3)
1817 0A44 90EA            JMP     X24
1818
1819
1820                      ;****************************************
1821                      ;*      FOR-LOOP INITIALIZATION         *
1822                      ;****************************************
1823
1824                      .LOCAL
1825 0A46 C2FE    SAVFOR: LD      FORPTR(P2)      ;CHECK FOR FOR STACK
1826 0A48 E4A6            XRI     L(PCSTAK)       ; OVERFLOW
1827 0A4A 9C04            JNZ     $1
1828 0A4C C40A            LDI     10
1829 0A4E 90E2    E17:    JMP     E16A
1830 0A50 E4A6    $1:     XRI     L(PCSTAK)
1831 0A52 31              XPAL    P1              ;POINT P1 AT FOR STACK
1832 0A53 CAF1            ST      P1LOW(P2)       ; SAVING OLD P1
1833 0A55 C410            LDI     H(FORSTK)
1834 0A57 35              XPAH    P1
1835 0A58 CAF0            ST      P1HIGH(P2)
1836 0A5A C2FD            LD      LSTK(P2)        ;POINT P3 AT AE STACK
```

```
1837 0A5C 33                XPAL    P3
1838 0A5D C410              LDI     H(AESTK)
1839 0A5F 37                XPAH    P3
1840 0A60 C3F9              LD      -7(P3)          ;GET VARIABLE INDEX
1841 0A62 CD01              ST      @1(P1)          ;SAVE ON FOR-STACK
1842 0A64 C3FC              LD      -4(P3)          ;GET L(LIMIT)
1843 0A66 CD01              ST      @1(P1)          ;SAVE
1844 0A68 C3FD              LD      -3(P3)          ;GET H(LIMIT)
1845 0A6A CD01              ST      @1(P1)          ;SAVE
1846 0A6C C3FE              LD      -2(P3)          ;GET L(STEP)
1847 0A6E CD01              ST      @1(P1)          ;SAVE
1848 0A70 C3FF              LD      -1(P3)          ;GET H(STEP)
1849 0A72 CD01              ST      @1(P1)          ;SAVE
1850 0A74 C2F1              LD      P1LOW(P2)       ;GET L(P1)
1851 0A76 CD01              ST      @1(P1)          ;SAVE
1852 0A78 C2F0              LD      P1HIGH(P2)      ;GET H(P1)
1853 0A7A CD01              ST      @1(P1)          ;SAVE
1854 0A7C 35                XPAH    P1              ;RESTORE OLD P1
1855 0A7D C2F1              LD      P1LOW(P2)
1856 0A7F 31                XPAL    P1
1857 0A80 CAFE              ST      FORPTR(P2)      ;UPDATE FOR STACK PTR
1858 0A82 C7FC              LD      @-4(P3)
1859 0A84 33                XPAL    P3
1860 0A85 CAFD              ST      LSTK(P2)        ;UPDATE AE STACK PTR
1861 0A87 90A7      X25:    JMP     X24
1862
1863
1864                ;*****************************************
1865                ;*      FIRST PART OF 'NEXT VAR'         *
1866                ;*****************************************
1867
1868                        .LOCAL
1869 0A89 C2FE      NEXTV:  LD      FORPTR(P2)      ;POINT P1 AT FOR STACK,
1870 0A8B E48A              XRI     L(FORSTK)       ; CHECKING FOR UNDERFLOW
1871 0A8D 9C04              JNZ     $1
1872 0A8F C40B              LDI     11              ;REPORT ERROR
1873 0A91 90BB              JMP     E17
1874 0A93 E48A      $1:     XRI     L(FORSTK)
1875 0A95 31                XPAL    P1
1876 0A96 CAF1              ST      P1LOW(P2)       ;SAVE OLD P1
1877 0A98 C410              LDI     H(FORSTK)
1878 0A9A 35                XPAH    P1
1879 0A9B CAF0              ST      P1HIGH(P2)
1880 0A9D C2F9              LD      LSTK(P2)        ;POINT P3 AT AE STACK
1881 0A9F 33                XPAL    P3
1882 0AA0 C410              LDI     H(AESTK)
1883 0AA2 37                XPAH    P3
1884 0AA3 C7FF              LD      @-1(P3)         ;GET VARIABLE INDEX
1885 0AA5 E1F9              XOR     -7(P1)          ;COMPARE WITH INDEX
1886 0AA7 9804              JZ      $10             ; ON FOR STACK: ERROR
1887 0AA9 C40C              LDI     12              ; IF NOT EQUAL
1888 0AAB 90A1      E18:    JMP     E17
1889 0AAD E1F9      $10:    XOR     -7(P1)          ;RESTORE INDEX
1890 0AAF 01                XAE                     ;SAVE IN EREG
```

```
1891 0AB0 C280          LD      EREG(P2)        ;GET L(VARIABLE)
1892 0AB2 02            CCL
1893 0AB3 F1FC          ADD     -4(P1)          ;ADD L(STEP)
1894 0AB5 CA80          ST      EREG(P2)        ;STORE IN VARIABLE
1895 0AB7 CB00          ST      (P3)            ;  AND ON STACK
1896 0AB9 C601          LD      @1(P2)          ;INCREMENT RAM PTR
1897 0ABB C280          LD      EREG(P2)        ;GET H(VARIABLE)
1898 0ABD F1FD          ADD     -3(P1)          ;ADD H(STEP)
1899 0ABF CA80          ST      EREG(P2)        ;STORE IN VARIABLE
1900 0AC1 CB01          ST      1(P3)           ;  AND ON STACK
1901 0AC3 C6FF          LD      @-1(P2)         ;RESTORE RAM POINTER
1902 0AC5 C1FA          LD      -6(P1)          ;GET L(LIMIT)
1903 0AC7 CB02          ST      2(P3)           ;PUT ON STACK
1904 0AC9 C1FB          LD      -5(P1)          ;GET H(LIMIT)
1905 0ACB CB03          ST      3(P3)           ;PUT ON STACK
1906 0ACD C1FD          LD      -3(P1)          ;GET H(STEP)
1907 0ACF 9410          JP      $2              ;IF NEGATIVE, INVERT
1908 0AD1 C404          LDI     4               ;  ITEMS ON A.E. STACK
1909 0AD3 CAEB          ST      NUM(P2)         ;NUM = LOOP COUNTER
1910 0AD5 C701  $LOOP:  LD      @1(P3)          ;GET BYTE FROM STACK
1911 0AD7 E4FF          XRI     OFF             ;INVERT IT
1912 0AD9 CBFF          ST      -1(P3)          ;PUT BACK ON STACK
1913 0ADB BAEB          DLD     NUM(P2)         ;DO UNTIL NUM = 0
1914 0ADD 9CF6          JNZ     $LOOP
1915 0ADF 9002          JMP     $3
1916 0AE1 C704  $2:     LD      @4(P3)          ;UPDATE AE STACK POINTER
1917 0AE3 33    $3:     XPAL    P3
1918 0AE4 CAFD          ST      LSTK(P2)
1919 0AE6 C2F1          LD      P1LOW(P2)       ;RESTORE OLD P1
1920 0AE8 31            XPAL    P1
1921 0AE9 C2F0          LD      P1HIGH(P2)
1922 0AEB 35            XPAH    P1
1923 0AEC 9099  X26:    JMP     X25
1924
1925
1926            ;****************************************
1927            ;*      SECOND PART OF 'NEXT VAR'       *
1928            ;****************************************
1929
1930 0AEE C2EF  NEXTV1: LD      LO(P2)          ;IS FOR-LOOP OVER WITH?
1931 0AF0 9808          JZ      $REDO           ;NO - REPEAT LOOP
1932 0AF2 C2FE          LD      FORPTR(P2)      ;YES - POP FOR-STACK
1933 0AF4 02            CCL
1934 0AF5 F4F9          ADI     -7
1935 0AF7 CAFE          ST      FORPTR(P2)
1936 0AF9 3F            XPPC    P3              ;RETURN TO I.L. INTERPRETER
1937 0AFA C2FE  $REDO:  LD      FORPTR(P2)      ;POINT P3 AT FOR STACK
1938 0AFC 33            XPAL    P3
1939 0AFD C410          LDI     H(FORSTK)
1940 0AFF 37            XPAH    P3
1941 0B00 C3FF          LD      -1(P3)          ;GET OLD P1 OFF STACK
1942 0B02 35            XPAH    P1
1943 0B03 C3FE          LD      -2(P3)
1944 0B05 31            XPAL    P1
```

```
 1945 0B06 90E4          JMP     X26
 1946 0B08 90A1   E19:   JMP     E18
 1947
 1948
 1949             ;*************************************
 1950             ;*      PRINT MEMORY AS STRING       *
 1951             ;*************************************
 1952                    !
 1953             ; THIS ROUTINE IMPLEMENTS THE STATEMENT:
 1954             ;      'PRINT' '$' FACTOR
 1955
 1956                    .LOCAL
 1957 0B0A C2EE   PSTRNG: LD     HI(P2)          ;POINT P1 AT STRING TO PRINT
 1958 0B0C 35             XPAH   P1
 1959 0B0D C2EF           LD     LO(P2)
 1960 0B0F 31             XPAL   P1
 1961 0B10                LDPI   P3,PUTC-1       ;POINT P3 AT PUTC ROUTINE
 1962 0B16 C501   $1:     LD     @1(P1)          ;GET A CHARACTER
 1963 0B18 E40D           XRI    0D              ;IS IT A CARRIAGE RETURN?
 1964 0B1A 98D0           JZ     X26             ;YES - WE'RE DONE
 1965 0B1C E40D           XRI    0D              ;NO - PRINT THE CHARACTER
 1966 0B1E 3F             XPPC   P3
 1967 0B1F 06             CSA                    ;MAKE SURE NO ONE IS
 1968 0B20 D420           ANI    020             ; TYPING ON THE TTY
 1969 0B22 9CF2           JNZ    $1              ; BEFORE REPEATING LOOP
 1970 0B24 90C6           JMP    X26
 1971
 1972
 1973             ;*************************************
 1974             ;*         INPUT A STRING            *
 1975             ;*************************************
 1976
 1977             ; THIS ROUTINE IMPLEMENTS THE STATEMENT:
 1978             ;      'INPUT' '$' FACTOR
 1979
 1980 0B26 C2EE   ISTRNG: LD     HI(P2)          ;GET ADDRESS TO STORE THE
 1981 0B28 37             XPAH   P3              ; STRING, PUT IT INTO P3
 1982 0B29 C2FF           LD     LO(P2)
 1983 0B2B 33             XPAL   P3
 1984 0B2C C501   $2:     LD     @1(P1)          ;GET A BYTE FROM LINE BUFFER
 1985 0B2E CF01           ST     @1(P3)          ;PUT IT IN SPECIFIED LOCATI
 1986 0B30 E40D           XRI    0D              ;DO UNTIL CHAR = CARR. RETU
 1987 0B32 9CF8           JNZ    $2
 1988 0B34 90B6   X27:    JMP    X26
 1989
 1990
 1991             ;*************************************
 1992             ;*   STRING CONSTANT ASSIGNMENT      *
 1993             ;*************************************
 1994
 1995             ; THIS ROUTINE IMPLEMENTS THE STATEMENT:
 1996             ;      '$' FACTOR '=' STRING
 1997
 1998                    .LOCAL
```

```
1999 0B36 C2EF   PUTSTR: LD      LO(P2)          ;GET ADDRESS TO STORE STRING
2000 0B38 33             XPAL    P3              ; PUT IT INTO P3
2001 0B39 C2EE           LD      HI(P2)
2002 0B3B 37             XPAH    P3
2003 0B3C C501   $LOOP:  LD      @1(P1)          ;GET A BYTE FROM STRING
2004 0B3E E422           XRI     '"'             ;CHECK FOR END OF STRING
2005 0B40 980E           JZ      $END
2006 0B42 E42F           XRI     '"' ! 0D        ;MAKE SURE THERE'S NO CR
2007 0B44 9C04           JNZ     $1
2008 0B46 C407           LDI     7
2009 0B48 90BE           JMP     E19             ;ERROR IF CARRIAGE RETURN
2010 0B4A E40D   $1:     XRI     0D              ;RESTORE CHARACTER
2011 0B4C CF01           ST      @1(P3)          ;PUT IN SPECIFIED LOCATION
2012 0B4E 90EC           JMP     $LOOP           ;GET NEXT CHARACTER
2013 0B50 C40D   $END:   LDI     0D              ;APPEND CARRIAGE RETURN
2014 0B52 CB00           ST      (P3)            ; TO STRING
2015 0B54 90DE           JMP     X27
2016
2017
2018             ;*************************************
2019             ;*            MOVE STRING            *
2020             ;*************************************
2021
2022             ; THIS ROUTINE IMPLEMENTS THE STATEMENT:
2023             ;     '$' FACTOR '=' '$' FACTOR
2024
2025                     .LOCAL
2026 0B56 C2FD   MOVSTR: LD      LSTK(P2)        ;POINT P3 AT A.E. STACK
2027 0B58 33             XPAL    P3
2028 0B59 C410           LDI     H(AESTK)
2029 0B5B 37             XPAH    P3
2030 0B5C C7FF           LD      @-1(P3)         ;GET ADDRESS OF SOURCE STRI
2031 0B5E 35             XPAH    P1              ; INTO P1
2032 0B5F C7FF           LD      @-1(P3)
2033 0B61 31             XPAL    P1
2034 0B62 C7FF           LD      @-1(P3)         ;GET ADDRESS OF DESTINATION
2035 0B64 01             XAE                     ; STRING INTO P3
2036 0B65 C7FF           LD      @-1(P3)
2037 0B67 33             XPAL    P3
2038 0B68 CAFD           ST      LSTK(P2)        ;UPDATE STACK POINTER
2039 0B6A 40             LDE
2040 0B6B 37             XPAH    P3
2041 0B6C C501   $LOOP:  LD      @1(P1)          ;GET A SOURCE CHARACTER
2042 0B6E CF01           ST      @1(P3)          ;SEND IT TO DESTINATION
2043 0B70 E40D           XRI     0D              ;REPEAT UNTIL CARRIAGE RET.
2044 0B72 98C0           JZ      X27
2045 0B74 06             CSA                     ; OR KEYBOARD INTERRUPT
2046 0B75 D420           ANI     020
2047 0B77 9CF3           JNZ     $LOOP
2048 0B79 90B9           JMP     X27
2049
2050
2051             ;*************************************
2052             ;*     PUT PAGE NUMBER ON STACK      *
```

```
2053                ;*************************************
2054
2055 0B7B AAFD   PUTPGE: ILD     LSTK(P2)
2056 0B7D AAFD           ILD     LSTK(P2)
2057 0B7F 33             XPAL    P3
2058 0B80 C410           LDI     H(AESTK)
2059 0B82 37             XPAH    P3
2060 0B83 C2F6           LD      PAGE(P2)
2061 0B85 CBFE           ST      -2(P3)
2062 0B87 C400           LDI     0
2063 0B89 CBFF           ST      -1(P3)
2064 0B8B 90A7           JMP     X27
2065
2066
2067                ;*************************************
2068                ;*         ASSIGN NEW PAGE           *
2069                ;*************************************
2070
2071                .LOCAL
2072 0B8D C2EF   NUPAGE: LD      LO(P2)          ;GET PAGE # FROM STACK,
2073 0B8F D407           ANI     7               ; GET THE LOW 3 BITS
2074 0B91 9C02           JNZ     $0              ;PAGE 0 BECOMES PAGE 1
2075 0B93 C401           LDI     1
2076 0B95 CAF6   $0:     ST      PAGE(P2)
2077 0B97 3F             XPPC    P3              ;RETURN
2078
2079
2080                ;*************************************
2081                ;*          FIND START OF PAGE       *
2082                ;*************************************
2083
2084                ; THIS ROUTINE COMPUTES THE START OF THE CURRENT TEXT PAGE,
2085                ; STORING THE ADDRESS IN TEMP2(P2) [THE HIGH BYTE], AND
2086                ; TEMP3(P2) [THE LOW BYTE].
2087
2088 0B98 C2F6   FNDPGE: LD      PAGE(P2)
2089 0B9A E401           XRI     1               ;SPECIAL CASE IS PAGE 1, BU
2090 0B9C 9C09           JNZ     $1              ; OTHERS ARE CONVENTIONAL
2091 0B9E C411           LDI     H(PGM)          ;PAGE 1 STARTS AT 'PGM'
2092 0BA0 CAE9           ST      TEMP2(P2)
2093 0BA2 C420           LDI     L(PGM)
2094 0BA4 CAE8           ST      TEMP3(P2)
2095 0BA6 3F             XPPC    P3              ;RETURN
2096 0BA7 E401   $1:     XRI     1               ;RESTORE PAGE #
2097 0BA9 01             XAE                     ;SAVE IT
2098 0BAA C404           LDI     4               ;LOOP COUNTER = 4
2099 0BAC CAEB           ST      NUM(P2)
2100 0BAE 40     $LOOP:  LDE                     ;MULTIPLY PAGE# BY 16
2101 0BAF 02             CCL
2102 0BB0 70             ADE
2103 0BB1 01             XAE
2104 0BB2 BAEB           DLD     NUM(P2)
2105 0BB4 9CF8           JNZ     $LOOP
2106 0BB6 40             LDE
```

```
/MP ASSEMBLER REV-A
BL  12/17/76


2107 0BB7 CAE9              ST      TEMP2(P2)       ;TEMP2 HAS HIGH BYTE
2108 0BB9 C402              LDI     2               ; OF ADDRESS NOW
2109 0BBB CAE8              ST      TEMP3(P2)       ;LOW BYTE IS ALWAYS 2
2110 0BBD 3F                XPPC    P3
2111
2112                    ;*************************************
2113                    ;*      MOVE CURSOR TO NEW PAGE      *
2114                    ;*************************************
2115
2116
2117 0BBE C2E9  CHPAGE: LD      TEMP2(P2)       ;PUT START OF PAGE
  18 0BC0 35            XPAH    P1              ; INTO P1.   THIS ROUTINE
2119 0BC1 C2E8          LD      TEMP3(P2)       ; MUST BE CALLED RIGHT
2120 0BC3 31            XPAL    P1              ; AFTER 'FNDPGE'
2121 0BC4 3F            XPPC    P3              ;RETURN
2122
212                     ;*************************************
212                     ;*      DETERMINE CURRENT PAGE       *
2125                    ;*************************************
2126
2127
2128 0BC5 35    DETPGE: XPAH    P1              ;CURRENT PAGE IS HIGH
2129 0BC6 01            XAE                     ; PART OF CURSOR DIVIDED
2130 0BC7 40            LDE                     ; BY 16
2131 0BC8 35            XPAH    P1
2132 0BC9 40            LDE
2133 0BCA 1C            SR
2134 0BCB 1C            SR
2135 0BCC 1C            SR
2136 0BCD 1C            SR
2137 0BCE CAF6          ST      PAGE(P2)
2138 0BD0 3F            XPPC    P3              ;RETURN
2139
2140
2141                    ;*************************************
2142                    ;*      CLEAR CURRENT PAGE           *
2143                    ;*************************************
2144
2145 0BD1 C2E9  NEWPGM: LD      TEMP2(P2)       ;POINT P1 AT CURRENT PAGE
2146 0BD3 35            XPAH    P1
2147 0BD4 C2E8          LD      TEMP3(P2)
2148 0BD6 31            XPAL    P1
2149 0BD7 C40D          LDI     0D              ;PUT DUMMY END-OF-LINE
2150 0BD9 C9FF          ST      -1(P1)          ; JUST BEFORE TEXT
2151 0BDB C4FF          LDI     -1              ;PUT -1 AT START OF TEXT
2152 0BDD C900          ST      (P1)
2153 0BDF C901          ST      1(P1)
2154 0BE1 3F            XPPC    P3              ;RETURN
2155
2156                    ;*************************************
2157                    ;*      FIND LINE NUMBER IN TEXT     *
2158                    ;*************************************
2159
2160
```

```
                    ;INPUTS:  THE START OF THE CURRENT PAGE IN TEMP2 AND TEMP3,
                    ;         THE LINE NUMBER TO LOOK FOR IN LO AND HI.
                    ;OUTPUTS: THE ADDRESS OF THE FIRST LINE IN THE NIBL TEXT
                    ;         WHOSE LINE NUMBER IS GREATER THAN OR EQUAL TO THE
                    ;         NUMBER IN HI AND LO, RETURNED IN P1 AND ALSO IN
                    ;         IN THE RAM VARIABLES LABLLO AND LABLHI.  THE SIGN
                    ;         BIT OF LABLHI IS SET IF EXACT LINE IS NOT FOUND.

                              .LOCAL
0BE2 C2E9   FNDLBL:   LD      TEMP2(P2)       ;POINT P1 AT START OF TEXT
0BE4 35               XPAH    P1
0BE5 C2E8             LD      TEMP3(P2)
0BE7 31               XPAL    P1
0BE8 C100   $1:       LD      (P1)            ;HAVE WE HIT END OF TEXT?
0BEA E4FF             XRI     0FF
0BEC 9412             JP      $2              ;YES - STOP LOOKING
0BEE 03               SCL                     ;NO - COMPARE LINE NUMBERS
0BEF C101             LD      1(P1)           ; BY SUBTRACTING
0BF1 FAEF             CAD     LO(P2)
0BF3 C100             LD      0(P1)
0BF5 FAEE             CAD     HI(P2)          ;IS TEXT LINE # >= LINE #?
0BF7 9407             JP      $2              ;YES - STOP LOOKING
0BF9 C102             LD      2(P1)           ;NO - TRY NEXT LINE IN TEXT
0BFB 01               XAE
0BFC C580             LD      @EREG(P1)       ; SKIP LENGTH OF LINE
0BFE 90E8             JMP     $1
0C00 31     $2:       XPAL    P1              ;SAVE ADDRESS OF FOUND LINE
0C01 CAF3             ST      LABLLO(P2)      ; IN LABLHI AND LABLLO
0C03 31               XPAL    P1
0C04 35               XPAH    P1
0C05 CAF2             ST      LABLHI(P2)
0C07 35               XPAH    P1
0C08 C2EF             LD      LO(P2)          ;WAS THERE AN EXACT MATCH?
0C0A E101             XOR     1(P1)
0C0C 9C07             JNZ     $3
0C0E C2EE             LD      HI(P2)
0C10 E100             XOR     0(P1)
0C12 9C01             JNZ     $3              ;NO - FLAG THE ADDRESS
0C14 3F               XPPC    P3              ;YES - RETURN NORMALLY
0C15 C2F2   $3:       LD      LABLHI(P2)      ;SET SIGN BIT OF HIGH PART
0C17 DC80             ORI     080             ; OF ADDRESS TO INDICATE
0C19 CAF2             ST      LABLHI(P2)      ; INEXACT MATCH OF LINE #'S
0C1B 3F               XPPC    P3
```

```
                        .PAGE   ' I. L. MACROS'
2205
2206                    ;*************************************
2207                    ;*          I. L. MACROS             *
2208                    ;*************************************
2209
2210                            .LOCAL
2211
2212
2213    2000    $TSTBIT =       TSTBIT*256
2214    8000    $CALBIT =       CALBIT*256
2215    4000    $JMPBIT =       JMPBIT*256
.6
2217                            .MACRO  TST,FAIL,A,B
2218                            .DBYTE  FAIL & OFFF ! $TSTBIT
2219                            .IF     #=2
2220                            .BYTE   'A'!080
2221                            .ELSE
2222                            .ASCII  'A'
2223                            .BYTE   'B'!080
2224                            .ENDIF
2225                            .ENDM
2226
2227                            .MACRO  TSTCR,FAIL
2228                            .DBYTE  FAIL & OFFF ! $TSTBIT
2229                            .BYTE   0D!080
2230                            .ENDM
2231
2232                            .MACRO  TSTV,FAIL
2233                            .ADDR   TSTVAR & OFFF
2234                            .DBYTE  FAIL
2235                            .ENDM
2236
2237                            .MACRO  TSTN,FAIL
2238                            .ADDR   TSTNUM & OFFF
2239                            .DBYTE  FAIL
2240                            .ENDM
2241
2242                            .MACRO  JUMP,ADR
2243                            .DBYTE  ADR & OFFF ! $JMPBIT
2244                            .ENDM
2245
2246                            .MACRO  CALL,ADR
2247                            .DBYTE  ADR & OFFF ! $CALBIT
2248                            .ENDM
2249
2250                            .MACRO  DO
2251                            .MLOC   I
2252                            .SET    I,1
2253                            .DO     #
2254                            .ADDR   #I & OFFF
2255                            .SET    I,I+1
2256                            .ENDDO
2257                            .ENDM
```

```
2258                        .PAGE   ' I. L. TABLE'
2259
2260                        ;****************************************
2261                        ;*              I. L. TABLE             *
2262                        ;****************************************
2263
2264 0C1C          START:   DO      NLINE
2265 0C1E          PROMPT:  DO      GETL
2266 0C20                   TSTCR   PRMPT1
2267 0C23                   JUMP    PROMPT
2268 0C25          PRMPT1:  TSTN    LIST
2269 0C29                   DO      FNDPGE,XCHGP1,POPAE,FNDLBL,INSRT
2270 0C33                   JUMP    PROMPT
2271
2272 0C35          LIST:    TST     RUN,'LIS','T'
2273 0C3B                   DO      FNDPGE
2274 0C3D                   TSTN    LIST1
2275 0C41                   DO      POPAE,FNDLBL
2276 0C45                   JUMP    LIST2
2277 0C47          LIST1:   DO      CHPAGE
2278 0C49          LIST2:   DO      LST
2279 0C4B          LIST3:   CALL    PRNUM
2280 0C4D                   DO      LST3
2281 0C4F                   JUMP    START
2282
2283 0C51          RUN:     TST     CLR,'RU','N'
2284 0C56                   DO      DONE
2285 0C58          BEGIN:   DO      FNDPGE,CHPAGE,STRT,NXT
2286
2287 0C60          CLR:     TST     NEW,'CLEA','R'
2288 0C67                   DO      DONE,CLEAR,NXT
2289
2290 0C6D          NEW:     TST     STMT,'NE','W'
2291 0C72                   TSTN    DFAULT
2292 0C76                   JUMP    NEW1
2293 0C78          DFAULT:  DO      LIT1
2294 0C7A          NEW1:    DO      DONE,POPAE,NUPAGE,FNDPGE,NEWPGM,NXT
2295
2296 0C86          STMT:    TST     LET,'LE','T'
2297 0C8B          LET:     TSTV    AT
2298 0C8F                   TST     SYNTAX,'='
2299 0C92                   CALL    RELEXP
2300 0C94                   DO      STORE,DONE,NXT
2301 0C9A          AT:      TST     IF,'@'
2302 0C9D                   CALL    FACTOR
2303 0C9F                   TST     SYNTAX,'='
2304 0CA2                   CALL    RELEXP
2305 0CA4                   DO      MOVE,DONE,NXT
2306
2307 0CAA          IF:      TST     UNT,'I','F'
2308 0CAE                   CALL    RELEXP
2309 0CB0                   TST     IF1,'THE','N'
2310 0CB6          IF1:     DO      POPAE,CMPR
2311 0CBA                   JUMP    STMT
```

```
2312
2313 0CBC        UNT:    TST     DO,'UNTI','L'
2314 0CC3                DO      CKMODE
2315 0CC5                CALL    RELEXP
2316 0CC7                DO      DONE,POPAE,UNTIL,DETPGE,NXT
2317
2318 0CD1        DO:     TST     GOTO,'D','O'
2319 0CD5                DO      CKMODE,DONE,SAVEDO,NXT
2320
2321 0CDD        GOTO:   TST     RETURN,'G','O'
2322 0CE1                TST     GOSUB,'T','O'
2323 0CE5                CALL    RELEXP
2324 0CE7                DO      DONE
2325 0CE9                JUMP    G01
2326 0CEB        GOSUB:  TST     SYNTAX,'SU','B'
2327 0CF0                CALL    RELEXP
2328 0CF2                DO      DONE,SAV
2329 0CF6        G01:    DO      FNDPGE,POPAE,FNDLBL,XFER,NXT
2330
2331 0D00        RETURN: TST     NEXT,'RETUR','N'
2332 0D08                DO      DONE,RSTR,DETPGE,NXT
2333
2334 0D10        NEXT:   TST     FOR,'NEX','T'
2335 0D16                DO      CKMODE
2336 0D18                TSTV    SYNTAX
2337 0D1C                DO      DONE,NEXTV
2338 0D20                CALL    GTROP
2339 0D22                DO      POPAE,NEXTV1,DETPGE,NXT
2340
2341 0D2A        FOR:    TST     STAT,'FO','R'
2342 0D2F                DO      CKMODE
2343 0D31                TSTV    SYNTAX
2344 0D35                TST     SYNTAX,'='
2345 0D38                CALL    RELEXP
2346 0D3A                TST     SYNTAX,'T','O'
2347 0D3E                CALL    RELEXP
2348 0D40                TST     FOR1,'STE','P'
2349 0D46                CALL    RELEXP
2350 0D48                JUMP    FOR2
2351 0D4A        FOR1:   DO      LIT1
2352 0D4C        FOR2:   DO      DONE,SAVFOR,STORE,NXT
2353
2354 0D54        STAT:   TST     PGE,'STA','T'
2355 0D5A                TST     SYNTAX,'='
2356 0D5D                CALL    RELEXP
2357 0D5F                DO      POPAE,MOVESR
2358 0D63                DO      DONE,NXT
2359
2360 0D67        PGE:    TST     DOLLAR,'PAG','E'
2361 0D6D                TST     SYNTAX,'='
2362 0D70                CALL    RELEXP
2363 0D72                DO      DONE,POPAE,NUPAGE,FNDPGE,CHPAGE,NXT
2364
2365 0D7E        DOLLAR: TST     PRINT,'$'
```

```
2366 0D81                    CALL    FACTOR
2367 0D83                    TST     SYNTAX,'='
2368 0D86                    TST     DOLR1,'"'
2369 0D89                    DO      POPAE,PUTSTR
2370 0D8D                    JUMP    DOLR2
2371 0D8F          DOLR1:    TST     SYNTAX,'$'
2372 0D92                    CALL    FACTOR
2373 0D94                    DO      XCHGP1,MOVSTR,XCHGP1
2374 0D9A          DOLR2:    DO      DONE,NXT
2375
2376 0D9E          PRINT:    TST     INPUT,'P','R'
2377 0DA2                    TST     PR1,'IN','T'
2378 0DA7          PR1:      TST     PR2,'"'
2379 0DAA                    DO      PRS
2380 0DAC                    JUMP    COMMA
2381 0DAE          PR2:      TST     PR3,'$'
2382 0DB1                    CALL    FACTOR
2383 0DB3                    DO      XCHGP1,POPAE,PSTRNG,XCHGP1
2384 0DBB                    JUMP    COMMA
2385 0DBD          PR3:      CALL    RELEXP
2386 0DBF                    CALL    PRNUM
2387 0DC1          COMMA:    TST     PR4,','
2388 0DC4                    JUMP    PR1
2389 0DC6          PR4:      TST     PR5,';'
2390 0DC9                    JUMP    PR6
2391 0DCB          PR5:      DO      NLINE
2392 0DCD          PR6:      DO      DONE,NXT
2393
2394 0DD1          INPUT:    TST     END,'INPU','T'
2395 0DD8                    DO      CKMODE
2396 0DDA                    TSTV    IN2
2397 0DDE                    DO      XCHGP1,GETL
2398 0DE2          IN1:      CALL    RELEXP
2399 0DE4                    DO      STORE,XCHGP1
2400 0DE8                    TST     IN3,','
2401 0DEB                    TSTV    SYNTAX
2402 0DEF                    DO      XCHGP1
2403 0DF1                    TST     SYNTAX,','
2404 0DF4                    JUMP    IN1
2405 0DF6          IN2:      TST     SYNTAX,'$'
2406 0DF9                    CALL    FACTOR
2407 0DFB                    DO      XCHGP1,GETL,POPAE,ISTRNG,XCHGP1
2408 0E05          IN3:      DO      DONE,NXT
2409
2410 0E09          END:      TST     ML,'EN','D'
2411 0E0E                    DO      DONE,BREAK
2412
2413 0E12          ML:       TST     REM,'LIN','K'
2414 0E18                    CALL    RELEXP
2415 0E1A                    DO      DONE,XCHGP1,POPAE,CALLML,XCHGP1,NXT
2416
2417 0E26          REM:      TST     SYNTAX,'RE','M'
2418 0E2B                    DO      IGNORE,NXT
2419
```

```
2420 0E2F          SYNTAX: DO      ERR
2421 0E31          ERRNUM: CALL    PRNUM
2422 0E33                  DO      FIN
2423
2424               ; NOTE: EACH RELATIONAL OPERATOR (EQ, LEQ, ETC.) DOES AN
2425               ; AUTOMATIC 'RTN' (THIS SAVES VALUABLE BYTES AND TIME)
2426
2427 0E35          RELEXP: CALL    EXPR
2428 0E37                  TST     REL1,'='
2429 0E3A                  CALL    EXPR
2430 0E3C                  DO      EQ
2431 0E3E          REL1:   TST     REL4,'<'
2432 0E41                  TST     REL2,'='
2433 0E44                  CALL    EXPR
2434 0E46                  DO      LEQ
2435 0E48          REL2:   TST     REL3,'>'
2436 0E4B                  CALL    EXPR
2437 0E4D                  DO      NEQ
2438 0E4F          REL3:   CALL    EXPR
2439 0E51                  DO      LSS
2440 0E53          REL4:   TST     RETEXP,'>'
2441 0E56                  TST     REL5,'='
2442 0E59                  CALL    EXPR
2443 0E5B                  DO      GEQ
2444 0E5D          REL5:   CALL    EXPR
2445 0E5F          GTROP:  DO      GTR
2446
2447 0E61          EXPR:   TST     EX1,'-'
2448 0E64                  CALL    TERM
2449 0E66                  DO      NEG
2450 0E68                  JUMP    EX3
2451 0E6A          EX1:    TST     EX2,'+'
2452 0E6D          EX2:    CALL    TERM
2453 0E6F          EX3:    TST     EX4,'+'
2454 0E72                  CALL    TERM
2455 0E74                  DO      ADD
2456 0E76                  JUMP    EX3
2457 0E78          EX4:    TST     EX5,'-'
2458 0E7B                  CALL    TERM
2459 0E7D                  DO      SUB
2460 0E7F                  JUMP    EX3
2461 0E81          EX5:    TST     RETEXP,'O','R'
2462 0E85                  CALL    TERM
2463 0E87                  DO      OROP
2464 0E89                  JUMP    EX3
2465 0E8B          RETEXP: DO      RTN
2466
2467 0E8D          TERM:   CALL    FACTOR
2468 0E8F          T1:     TST     T2,'*'
2469 0E92                  CALL    FACTOR
2470 0E94                  DO      MUL
2471 0E96                  JUMP    T1
2472 0E98          T2:     TST     T3,'/'
2473 0E9B                  CALL    FACTOR
```

```
2474 0E9D                  DO      DIV
2475 0E9F                  JUMP    T1
2476 0EA1          .T3:    TST     RETEXP,'AN','D'
2477 0EA6                  CALL    FACTOR
2478 0EA8                  DO      ANDOP
2479 0EAA                  JUMP    T1
2480
2481 0EAC          FACTOR: TSTV    F1
2482 0EB0                  DO      IND,RTN
2483 0EB4          F1:     TSTN    F2
2484 0EB8                  DO      RTN
2485 0EBA          F2:     TST     F3,'#'
2486 0EBD                  DO      HEX,RTN
2487 0EC1          F3:     TST     F4,'('
2488 0EC4                  CALL    RELEXP
2489 0EC6                  TST     SYNTAX,')'
2490 0EC9                  DO      RTN
2491 0ECB          F4:     TST     F5,'@'
2492 0ECE                  CALL    FACTOR
2493 0ED0                  DO      EVAL,RTN
2494 0ED4          F5:     TST     F6,'NO','T'
2495 0ED9                  CALL    FACTOR
2496 0EDB                  DO      NOTOP,RTN
2497 0EDF          F6:     TST     F7,'STA','T'
2498 0EE5                  DO      STATUS,RTN
2499 0EE9          F7:     TST     F8,'TO','P'
2500 0EEE                  DO      ENDPGE,TOP,RTN
2501 0EF4          F8:     TST     F9,'MO','D'
2502 0EF9                  CALL    DOUBLE
2503 0EFB                  DO      DIV,MODULO,RTN
2504 0F01          F9:     TST     F10,'RN','D'
2505 0F06                  CALL    DOUBLE
2506 0F08                  DO      RANDOM,SUB,ADD,DIV,MODULO,ADD,RTN
2507 0F16          F10:    TST     SYNTAX,'PAG','E'
2508 0F1C                  DO      PUTPGE,RTN
2509
2510 0F20          DOUBLE: TST     SYNTAX,'('
2511 0F23                  CALL    RELEXP
2512 0F25                  TST     SYNTAX,','
2513 0F28                  CALL    RELEXP
2514 0F2A                  TST     SYNTAX,')'
2515 0F2D                  DO      RTN
2516
2517 0F31          PRNUM:  DO      XCHGP1,PRN
2518 0F33          PRNUM1: DO      DIV,PRN1,XCHGP1,RTN
```

```
2519                        .PAGE   'ERROR MESSAGES'
2520
2521                    ;****************************************
2522                    ;*            ERROR MESSAGES            *
2523                    ;****************************************
2524
2525                            .MACRO  MESSAGE,A,B
2526                            .ASCII  'A'
2527                            .BYTE   'B'!080
2528                            .ENDM
2529
2530 0F3B           MESGS:   MESSAGE ' ERRO','R'      ; 1
2531 0F41                   MESSAGE 'ARE','A'       ; 2
2532 0F45                   MESSAGE 'STM','T'       ; 3
2533 0F49                   MESSAGE 'CHA','R'       ; 4
2534 0F4D                   MESSAGE 'SNT','X'       ; 5
2535 0F51                   MESSAGE 'VAL','U'       ; 6
2536 0F55                   MESSAGE 'END','"'       ; 7
2537 0F59                   MESSAGE 'NOG','O'       ; 8
2538 0F5D                   MESSAGE 'RTR','N'       ; 9
2539 0F61                   MESSAGE 'NES','T'       ; 10
2540 0F65                   MESSAGE 'NEX','T'       ; 11
2541 0F69                   MESSAGE 'FO','R'        ; 12
2542 0F6C                   MESSAGE 'DIV','O'       ; 13
2543 0F70                   MESSAGE 'BR','K'        ; 14
2544 0F73                   MESSAGE 'UNT','L'       ; 15
```

```
SC/MP ASSEMBLER REV-A
NIBL  12/17/76
 TELETYPE ROUTINES

                          .PAGE   ' TELETYPE ROUTINES'
2545
2546
2547                      ;****************************************
2548                      ;*      GET CHARACTER AND ECHO IT       *
2549                      ;****************************************
2550                      ;
2551                      .LOCAL
2552 OF77 C408   GECO:   LDI     8                ;SET COUNT = 8
2553 OF79 CAEB           ST      NUM(P2)
2554 OF7B 06             CSA                      ;SET READER RELAY
2555 OF7C DC02           ORI     2
2556 OF7E 07             CAS
2557 OF7F 06      $1:    CSA                      ;WAIT FOR START BIT
2558 OF80 D420           ANI     020
2559 OF82 9CFB           JNZ     $1               ;NOT FOUND
2560 OF84 C4C3           LDI     0C3              ;DELAY 1/2 BIT TIME
2561 OF86 8F08           DLY     8
2562 OF88 06             CSA                      ;IS START BIT STILL THERE?
2563 OF89 D420           ANI     020
2564 OF8B 9CF2           JNZ     $1               ;NO
2565 OF8D 06             CSA                      ;SEND START BIT
2566 OF8E D4FD           ANI     %2               ;RESET READER RELAY
2567 OF90 DC01           ORI     1
2568 OF92 07             CAS
2569 OF93 C445    $2:    LDI 10  045              ;DELAY 1 BIT TIME
2570 OF95 8F11           DLY     011
2571 OF97 06             CSA 5                    ;GET BIT (SENSEB)
2572 OF98 D420           ANI 10  020
2573 OF9A 9804           JZ  9   $3
2574 OF9C C401           LDI 10  1
2575 OF9E 9004           JMP 9   $4
2576 OFA0 C400    $3:    LDI 10  0
2577 OFA2 9C00           JNZ 9   $4
2578 OFA4 CAEA    $4:     ST 18   TEMP(P2)         ;SAVE BIT VALUE (0 OR 1)
2579 OFA6 1F             RRL 5                    ;ROTATE INTO LINK
2580 OFA7 01             XAE 7
2581 OFA8 1D             SRL 5                    ;SHIFT INTO CHARACTER
2582 OFA9 01             XAE 7                    ;RETURN CHAR TO E
2583 OFAA 06             CSA 5                    ;ECHO BIT TO OUTPUT
2584 OFAB DC01           ORI 10  1
2585 OFAD E2EA           XOR 18  TEMP(P2)
2586 OFAF 07             CAS 6
2587 OFB0 BAEB           DLD 22  NUM(P2)          ;DECREMENT BIT COUNT
2588 OFB2 9CDF           JNZ 9   $2       T=184   ;LOOP UNTIL 0
2589 OFB4 06             CSA                      ;SET STOP BIT
2590 OFB5 D4FE           ANI     0FE
2591 OFB7 07             CAS
2592 OFB8 8F11           DLY     011              ;DELAY APPROX. 1 BIT TIME
2593 OFBA 40             LDE                      ;AC HAS INPUT CHARACTER
2594 OFBB D47F           ANI     07F
2595 OFBD 01             XAE
2596 OFBE 40             LDE
2597 OFBF 3F             XPPC    P3               ;RETURN
2598 OFC0 90B5           JMP     GECO
```

```
SC/MP ASSEMBLER REV-A
NIBL  12/17/76
 TELETYPE ROUTINES


2599
2600                ;****************************************
2601                ;*      PRINT CHARACTER AT TTY          *
2602                ;****************************************
2603
2604 0FC2 01       PUTC:  XAE
2605 0FC3 C4BB  86         LDI     0BB             ;DELAY ALMOST
2606 0FC5 8F2F  101        DLY     02F             ; 3 BIT TIMES
2607 0FC7 06               CSA                     ;SET OUTPUT BIT TO LOGIC 0
2608 0FC8 DC01             ORI     1               ; FOR START BIT
2609 0FCA 07               CAS
2610 0FCB C409             LDI     9               ;INITIALIZE BIT COUNT
2611 0FCD CAE8  191        ST      TEMP3(P2)
2612 0FCF C454  PUTC1:     LDI  10 054             ;DELAY 1 BIT TIME
2613 0FD1 8F11  100        DLY     011
2614 0FD3 BAE8             DLD  22 TEMP3(P2)       ;DECREMENT BIT COUNT.
2615 0FD5 9810             JZ   9  PUTC2
2616 0FD7 40               LDE  6                  ;PREPARE NEXT BIT
2617 0FD8 D401             ANI  10 1
2618 0FDA CAE9             ST   16 TEMP2(P2)
2619 0FDC 01               XAE  7                  ;SHIFT DATA RIGHT 1 BIT
2620 0FDD 1C               SR   5
2621 0FDE 01               XAE  7
2622 0FDF 06               CSA  5                  ;SET UP OUTPUT BIT
2623 0FE0 DC01             ORI  10 1
2624 0FE2 E2E9             XOR  16 TEMP2(P2)
2625 0FE4 07               CAS  6                  ;PUT BIT TO TTY
2626 0FE5 90E8             JMP  9  PUTC1   T=142
2627 0FE7 06       PUTC2:  CSA                     ;SET STOP BIT
2628 0FE8 D4FE             ANI     0FE
2629 0FEA 07               CAS
2630 0FEB 3F               XPPC    P3              ;RETURN
2631 0FEC 90D4             JMP     PUTC
2632      0000             .END    0
```

A[illegible]	032D	AESTK	1050	ANDOP	05E8	AT	0C9A
BEGIN	0C58	BREAK	0280	CALBIT	0080	CALLML	0967
CHEAT	007C	CHEAT1	00C0	CHPAGE	0BBE	CHRNUM	FFE7
CK1	0641	CKMODE	063C	CLEAR	0051	CLEAR1	0056
CLR	0C60	CMP	055A	CMP1	05BA	CMP2	05C2
CMPR	05D1	COMMA	0DC1	DETPGE	0BC5	DFAULT	0C78
DIV	0408	DO	0CD1	DOLLAR	0D7E	DOLR1	0D8F
DOLR2	0D9A	DONE	0130	DONE1	013E	DONE2	013F
DOPTR	FFFF	DOSTAK	107A	DOUBLE	0F20	E0	014B
E0A	0108	E1	018D	E10	07CA	E11	0822
E12	0871	E12A	08E5	E13	0914	E14	0954
E15	0980	E16	09D0	E16A	0A32	E17	0A4E
E18	0AAB	E19	0B08	E2	01C4	E3A	0282
E4	02D7	E5	0304	E6	0370	E6A	03CA
E8	0643	E8A	06DD	E8B	06AA	E9	0753
END	0E09	EQ	0544	EREG	FF80	ERR	021B
ERR1	021D	ERR2	021F	ERRNUM	0E31	EVAL	07EA
EX1	0E6A	EX2	0E6D	EX3	0E6F	EX4	0E78
E[illegible]	0E81	EXECIL	0076	EXPR	0E61	F1	0EB4
F[illegible]	0F16	F2	0EBA	F3	0EC1	F4	0ECB
F5	0ED4	F6	0EDF	F7	0EE9	F8	0EF4
F9	0F01	FACTOR	0EAC	FAIL	05D9	FAILHI	FFEC
FAILLO	FFED	FALSE	05C0	FIN	02AA	FNDLBL	0BE2
FNDPGE	0B98	FOR	0D2A	FOR1	0D4A	FOR2	0D4C
FORPTR	FFFE	FORSTK	108A	GECO	0F77	GEQ	0558
GEQ1	05B6	GETL	0757	GO1	0CF6	GOSUB	0CEB
GOTO	0CDD	GTR	0554	GTR1	05AB	GTROP	0E5F
HEX	064C	HI	FFEE	HILINE	FFF7	IF	0CAA
IF1	0CB6	IGNORE	09B9	ILC1	00A8	ILCALL	009E
IN1	0DE2	IN2	0DF6	IN3	0E05	IND	052C
INPUT	0DD1	INSRT	0824	ISTRNG	0B26	JMPBIT	0040
LABLHI	FFF2	LABLLO	FFF3	LBUF	10D6	LEQ	0550
LEQ1	05A2	LET	0C8B	LIST	0C35	LIST1	0C47
LIST2	0C49	LIST3	0C4B	LISTNG	FFF5	LIT1	0A34
LO	FFEF	LOLINE	FFF8	LSS	054C	LSS1	059A
LST	02D9	LST2	02F7	LST3	0306	LST4	030C
LST5	031C	LSTK	FFFD	MESGS	0F3B	ML	0E12
M[illegible]ULO	09C0	MOVE	0808	MOVESR	094D	MOVSTR	0B56
MUL	0372	NEG	035B	NEQ	0548	NEQ1	0591
NEW	0C6D	NEW1	0C7A	NEWPGM	0BD1	NEXT	0D10
NEXTV	0A89	NEXTV1	0AEE	NLINE	020D	NOJUMP	009B *
NOTOP	05F0	NUM	FFEB	NUPAGE	0B8D	NXT	0284
NXT1	02A1	OROP	05EC	P1	0001	P1HIGH	FFF0
P1LOW	FFF1	P2	0002	P3	0003	PAGE	FFF6
PCHIGH	FFFA	PCLOW	FFFB	PCSTAK	10A6	PCSTK	FFF9
PGE	0D67	PGM	1120	POPAE	0916	PR1	0DA7
PR2	0DAE	PR3	0DBD	PR4	0DC6	PR5	0DCB
PR6	0DCD	PRINT	0D9E	PRMPT1	0C25	PRN	018F
PRN1	01C6	PRNUM	0F2F	PRNUM1	0F33	PROMPT	0C1E
PRS	0176	PRS1	018B	PSTRNG	0B0A	PUTC	0FC2
PUTC1	0FCF	PUTC2	0FE7	PUTPGE	0B7B	PUTSTR	0B36
RANDOM	09D2	REL1	0E3E	REL2	0E48	REL3	0E4F
REL4	0E53	REL5	0E5D	RELEXP	0E35	REM	0E26
RETEXP	0E8B	RETURN	0D00	RNDF	FFE6	RNDX	FFE5

R[illegible]Y	FFE4	RSTR	0143	RSTR1	014D	RSTR2	015F
RTN	00F6	RUN	0C51	RUNMOD	FFF4	SAV	010A
SAV1	0126	SAV2	012C	SAVEDO	0978	SAVFOR	0A46
SBRPTR	FFFC	SBRSTK	106A	SETZ	0585	START	0C1C
STAT	0D54	STATUS	0956	STMT	0C86	STORE	04C1
STRT	02C0	SUB	0344	SYNTAX	0E2F	T1	0E8F
T2	0E98	T3	0EA1	TEMP	FFEA	TEMP2	FFE9
TEMP3	FFE8	TERM	0E8D	TOP	0994	TST	00C2
TSTBIT	0020	TSTNUM	06AC	TSTVAR	04E1	UNT	0CBC
UNTIL	0928	VARS	101C	X0	00E7	X1	015D
X10	04DA	X11	0542	X12	058F	X12A	05E6
X12B	062F	X12C	0672	X13	06DB	X14	0755
X15	07B8	X16	07E8	X17	0820	X19	086F
19A	08E9	X20	090D	X21	0952	X22	0992
X23	09CE	X24	0A30	X25	0A87	X26	0AEC
X27	0B34	X4	01C2	X5	0219	X5A	027E
X6	02D5	X6A	0302	X7	0342	X8	036E
X9	03E7	X9A	0431	X9B	049C	XCHGP1	0631
XF[illegible]R	0169	XFER1	0171	ZZ0001	101C	ZZ0002	1120
ZZ0003	0FC1	ZZ0004	0FC1	ZZ0005	0FC1	ZZ0006	0FC1
ZZ0007	0F3B	ZZ0008	0F3B	ZZ0009	0FC1	ZZ000A	10D6
ZZ000B	0FC1	ZZ000C	101C	ZZ000D	0FC1	ZZ000E	0002
ZZ000F	0002	ZZ0010	0006	ZZ0011	0002	ZZ0012	0003
ZZ0013	0002	ZZ0014	0002	ZZ0015	0002	ZZ0016	0002
ZZ0017	0005	ZZ0018	0004	ZZ0019	0002	ZZ001A	0007
ZZ001B	0004	ZZ001C	0004	ZZ001D	0003	ZZ001E	0002
ZZ001F	0006	ZZ0020	0005	ZZ0021	0002	ZZ0022	0003
ZZ0023	0006	ZZ0024	0005	ZZ0025	0002	ZZ0026	0003
ZZ0027	0005	ZZ0028	0002	ZZ0029	0002	ZZ002A	0005
ZZ002B	0003	ZZ002C	0003	ZZ002D	0007	ZZ002E	0003
ZZ002F	0004	ZZ0030	0003	ZZ0031	0002	ZZ0032	0005
ZZ0033	0002	ZZ0034	0003	ZZ0035	0002	ZZ0036	0003
ZZ0037	0003	ZZ0038	0002	ZZ0039	0006	ZZ003A	0003
ZZ003B	0003	ZZ003C	0007	ZZ003D	0003	ZZ003E	0002
ZZ003F	0002	ZZ0040	0002	ZZ0041	0002	ZZ0042	0002
ZZ0043	0002	ZZ0044	0002	ZZ0045	0002	ZZ0046	0002
ZZ0047	0002	ZZ0048	0002	ZZ0049	0002	ZZ004A	0002
ZZ004B	0002	ZZ004C	0002	ZZ004D	0002	ZZ004E	0003
ZZ004F	0002	ZZ0050	0003	ZZ0051	0002	ZZ0052	0003
ZZ0053	0003	ZZ0054	0003	ZZ0055	0004	ZZ0056	0004
ZZ0057	0008	ZZ0058	0003	ZZ0059	0002	ZZ005A	0003
ZZ005B	0005	$0	002A	$0	0418	$0	0773
$0	099A	$0	0B95	$1	0043	$1	01BE
$1	0235	$1	038F	$1	043B	$1	05F2
$1	06DF	$1	0776	$1	083C	$1	0934
$1	0982	$1	09A0	$1	0A01	$1	0A50
$1	0A93	$1	0B16	$1	0B4A	$1	0BA7
$1	0BE8	$1	0F7F	$10	0AAD	$2	01F7
$2	0252	$2	03A0	$2	044C	$2	06FF
$2	07B6 *	$2	0846	$2	09A7	$2	0AE1
$2	0B2C	$2	0C00	$2	0F93	$3	0259
$3	03CC	$3	04A4	$3	084E	$3	0AE3
$3	0C15	$3	0FA0	$4	03E9	$4	085B
$4	0FA4	$5	0862	$ABOR	06BE	$ADD	08E7

```
SC/MP ASSEMBLER REV-A
NIBL  12/17/76



$A  1   0903    $CALB   8000    $CR     07DC    $DOWN   0889
$END    04BB    $END    06A4    $END    0B50    $ENT1   049E
$ENTE   0683    $ENTE   07BA    $ERR    0751    $EXIT   03FA
$FAIL   04F3    $JMPB   4000    $LETR   0674    $LOOP   002C
$LOOP   00D6    $LOOF   01FB    $LOOP   0239    $LOOP   03AE
$LOOP   0458    $LOOP   0664    $LOOP   06EF    $LOOP   09DD
$LOOP   0AD5    $LOOP   0B3C    $LOOP   0B6C    $LOOP   0BAE
$MAYB   0505    $MOVE   0873    $MSG    023F    $NEQ    00E9
$NOT    0620    $OK     0512    $OK     0680    $OR     060E
$POS    0433    $PRNT   01E7    $REDO   0940    $REDO   0AFA
$RET    06CE    $RUB    07D2    $SCAN   00C4    $SHIF   068A
$SHIF   070C    $SKIP   065C    $TSTB   2000    $UP     0899
$UP1    08A5    $UP2    08C6    $UP3    08C8    $UP4    08D8
.XH     07CC    $XU     07AA

NO ERROR LINES
SOURCE CHECKSUM = B7E4
INPUT FILE     1:NIBL1217.SRC
```