```
*************************************************************
*                                                           *
*                      B 2 2 0 S I M                        *
*                                                           *
*                 Burroughs 220 Simulator                   *
*                                                           *
*   Written by Michael J. Mahon   -   March 21, 2016       *
*                                                           *
* The B220 is a BCD word-oriented computer with 5000        *
* 11-digit words in the following format:                   *
*                                                           *
*      _____________________________________________        *
*     | S | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |       *
*     |___|___|___|___|___|___|___|___|___|___|___|       *
*                                                           *
* If the sign digit (S) is even, the number is positive,    *
* if odd, negative.  If S is 2, the word is interpreted     *
* as five alphanumeric characters.                          *
*                                                           *
* "Partial fields" may be specified within a word by a      *
* 2-digit partial field specification, sL, where s is       *
* the rightmost digit of the field and L is the length,     *
* extending to the left no further than the Sign digit.     *
*                                                           *
* Decimal floating-point data is stored in this format:     *
*                                                           *
*      _____________________________________________        *
*     | S | E | E | M | M | M | M | M | M | M | M |       *
*     |___|___|___|___|___|___|___|___|___|___|___|       *
*                                                           *
* S is the sign of the mantissa, as for fixed-point data.   *
*                                                           *
* EE is the excess-50 power of ten.                         *
*                                                           *
* MMMMMMMM is the fractional, normalized mantissa.          *
*                                                           *
* Instructions have the following format:                   *
*                                                           *
*      _____________________________________________        *
*     | S | V | V | V | V | O | P | A | D | D | R |       *
*     |___|___|___|___|___|___|___|___|___|___|___|       *
*                                                           *
* If S is odd, ADDR is modified by the B register before    *
* use.                                                      *
*                                                           *
* The Variant field (VVVV) has an op-specific format.       *
*                                                           *
* The OP field is the opcode.                               *
*                                                           *
* The ADDR field is the address part of the instruction     *
* which is augmented by B if the Sign digit is odd.         *
*                                                           *
*************************************************************
```

```
54  ************************************************************
55  *                                                          *
56  *                         History                          *
57  *                                                          *
58  * 03/29/16 - Ran first B220 op--HLT!  BCD address to MEM *
59  *            address is OK.                                *
60  *                                                          *
61  * 03/31/16 - Began implementing B220 front panel display *
62  *            in 40-column text mode.                       *
63  *                                                          *
64  * 04/02/16 - Front panel complete, adding keyboard cntl. *
65  *                                                          *
66  * 04/05/16 - Keyboard control complete, adding opcodes.  *
67  *                                                          *
68  * 04/11/16 - Refined error handling. Added B220CODE file *
69  *            loading. Implemented partial field STA/R/B.  *
70  *                                                          *
71  * 04/12/16 - Added conditional branches, STx, LDR, LDB,  *
72  *            LSA, CLx, CLL, SRx, IBB, DBB.                 *
73  *            Revised manual (keyboard) control.            *
74  *                                                          *
75  * 04/13/16 - Added non-BCD digit checking for addresses. *
76  *            Improved macros for B220 code assembly.       *
77  *            Split source into small 'put' files.          *
78  *                                                          *
79  * 04/15/16 - Added SLx and tested all shifts.              *
80  *                                                          *
81  * 04/18/16 - Added ADD and SUB and variants.               *
82  *                                                          *
83  * 04/19/16 - Added ADL, tested ADD, ADA, SUB, SUA, ADL.  *
84  *                                                          *
85  * 04/22/16 - Added simple MUL and a faster, byte-shifting*
86  *            version (currently FMU).                      *
87  *                                                          *
88  * 04/26/16 - Added EXT and RND. Added special cases for  *
89  *            SRT 10 and  SLT 10.                           *
90  *                                                          *
91  * 04/27/16 - Added simple version of DIV.                  *
92  *                                                          *
93  * 04/29/16 - Added CFA, CFR.                               *
94  *                                                          *
95  * 05/02/16 - Added BFA, BFR. Made 'compare' subroutine.  *
96  *                                                          *
97  * 05/04/16 - Added RTF, DFL, and DLB.  Split B220EXEC.   *
98  *                                                          *
99  * 05/09/16 - Added help redisplay. Paginated EXEC1 & 2.  *
100 *                                                          *
101 * 05/12/16 - Moved HLT execution to 'fetch'. Looks good! *
102 *                                                          *
103 * 05/15/16 - Fixed bug in 'compare'.  Added simple SPO.  *
104 *                                                          *
105 * 05/16/16 - Added Z reset command, revised help.          *
106 *                                                          *
107 * 05/18/16 - Added PWR command; first disk command.        *
108 *                                                          *
109 * 06/02/16 - Added PRD, PRB commands, removed B220CODE   *
110 *            pre-load hack.                                *
111 *                                                          *
112 * 06/07/16 - Moved FP ops to B220EXEC2. Changed Quit to  *
113 *            go to full text window and reconnect ProDOS.*
114 *                                                          *
115 * 06/19/16 - Fixed STR/STB partial field bug.              *
116 *                                                          *
117 * 06/24/16 - Changed PWR to truncate preexisting file.   *
118 *                                                          *
119 * 07/01/16 - Added FAD, FSU.                               *
120 *                                                          *
```

```
121  * 07/21/16 - Added FMU.                                        *
122  *                                                              *
123  * 07/25/16 - Many small JMP --> Bxx space optimizations.       *
124  *            RTF now moves upward! Generalized 'clear'.        *
125  *                                                              *
126  * 07/28/16 - Added FDV. Organized shift subroutines.           *
127  *                                                              *
128  * 08/22/16 - Modified 'b220asc' table for ) and %.             *
129  *                                                              *
130  * 08/27/16 - Fixed LBC bug--hi byte was high by one.           *
131  *            Fixed SPO: +, form feed, and 'ignore'.            *
132  *                                                              *
133  * 09/01/16 - Implemented B220 "tab" in SPO.                    *
134  *                                                              *
135  * 09/02/16 - Fixed RTF: rB now incremented when NN = 00.       *
136  *                                                              *
137  * 09/03/16 - Fixed BCH. Was branching on equal.                *
138  *                                                              *
139  * 09/05/16 - Fixed IFL, DFL, DLB: if s odd, zeroed s+1.        *
140  *                                                              *
141  * 09/09/16 - Added SOR/SOH op and subset of Mag Tape ops.      *
142  *                                                              *
143  * 09/10/16 - Split PTUNITn into PTRDRn and PTPCHn.             *
144  *                                                              *
145  * 09/11/16 - Combined paper tape and mag tape I/O.             *
146  *                                                              *
147  * 09/16/16 - Added MRD B-modification.                         *
148  *                                                              *
149  * 09/20/16 - Added MPE as NOP.                                 *
150  *                                                              *
151  * 09/21/16 - Added MLS for SNAP 1E.                            *
152  *                                                              *
153  * 09/23/16 - Added IOM (Interrogate Overflow Mode).            *
154  *                                                              *
155  * 09/24/16 - Fixed IFL bug: No Ov if hi field posn even.       *
156  *                                                              *
157  * 11/12/16 - Several small cleanups. ** RELEASED v1.0 **       *
158  *                                                              *
159  * 01/16/17 - Moved MEM to top in prep for IOCFG addition.      *
160  *                                                              *
161  * 01/17/17 - Added I/O configuration editor.                   *
162  *            Restricted PTRDR and PTPCH units to 0 and 1.      *
163  *                                                              *
164  * 01/25/17 - Integrated I/O Config Editor into B220SIM.        *
165  *            Fixed MPB bug.                                    *
166  *                                                              *
167  * 02/01/17 - Added "v1.1" and I/O Config help line.            *
168  *            ** RELEASED v1.1 **                               *
169  *                                                              *
170  * 04/27/17 - Added 'skipincP' to skip P reg increment if       *
171  *            PRB sign 6/7 instruction executed.                *
172  *                                                              *
173  * 05/01/17 - Char code matched to CCONV: 04 = ), 10 = (,       *
174  *            27 = $, 32 = ?, 34 = '                            *
175  *                                                              *
176  ****************************************************************
```

```
 180
 181            use   B220DEFS
>1     * 6502 equates
>2
>3     BCSop    equ   $B0        ; BCS opcode
>4     BNEop    equ   $D0        ; BNE opcode
>5     CLCop    equ   $18        ; CLC opcode
>6     SECop    equ   $38        ; SEC opcode
>7     NOPop    equ   $EA        ; NOP opcode
>8     ADCZop   equ   $65        ; ADC zp opcode
>9     BITZop   equ   $24        ; BIT zp opcode
>10    CMPIop   equ   $C9        ; CMP # opcode
>11    SBCZop   equ   $E5        ; SBC zp opcode
>12    ADCYop   equ   $79        ; ADC aaaa,y opcode
>13    SBCYop   equ   $F9        ; SBC aaaa,y opcode
>14
>15    * Apple equates
>16
>17    WNDTOP   equ   $22        ; Top line of text window
>18    CH       equ   $24        ; COUT horizontal cursor
>19    BASL     equ   $28        ; Screen base address
>20    IN       equ   $200       ; Keyboard input buffer
>21    KBD      equ   $C000      ; Keyboard port
>22    ALTCHAR  equ   $C00F      ; Store to enable alt charset
>23    KBSTROBE equ   $C010      ; Keyboard strobe reset
>24    SPKR     equ   $C030      ; Toggle speaker
>25
>26    * Apple entry points
>27
>28    DOSCON   equ   $3D0       ; ProDOS reconnect vector
>29    DOSCMD   equ   $BE03      ; BASIC.SYSTEM PDOS command
>30    PRINTERR equ   $BE0C      ; Print ProDOS error msg
>31    BSSTATE  equ   $BE42      ; BASIC.SYSTEM state var
>32    PRBL2    equ   $F94A      ; Print (X) blanks
>33    TABV     equ   $FB5B      ; Vertical tab to (A)
>34    BASCALC  equ   $FBC1      ; Set BASL to line (A)
>35    BEEP     equ   $FBDD      ; Beep
>36    HOME     equ   $FC58      ; Clear screen
>37    CROUT    equ   $FD8E      ; Output a CR
>38    COUT     equ   $FDED      ; Output char in A
>39
>40    * Simulation parameters
>41
>42    memb     equ   5000*6     ; 5000 6-byte B220 words
>43    MEM      equ   $9600-memb ; Simulated B220 memory
>44    dispcnt  equ   100        ; Update panel every 100 instrs
```

```
                    >46   ************************************************************
                    >47   *                                                          *
                    >48   *                  Page zero variables                     *
                    >49   *                                                          *
                    >50   ************************************************************
                    >51
                    >52
                    >53            dum   $90        ; Start of Page Zero variables
                    >54
                    >55   * B220 memory fields
                    >56
                    >57   S         equ   0          ; Sign digit
                    >58   sL        equ   1          ; rC sL specifier
                    >59   VV        equ   2          ; rC Variant
                    >60   OP        equ   3          ; rC Op code
                    >61   ADDR      equ   4          ; rC BCD address
                    >62   EXP       equ   1          ; FP exponent
                    >63   MANT      equ   2          ; FP mantissa
                    >64
                    >65   * Simulated B220 State Variables
                    >66
                    >67   B220strt  equ   *          ; Start of simulated B220 state
0090: 00 00 00      >68   rBx       ds    4          ; 4 const zero byte prefix to rB
0094: 00 00         >69   rB        dw    0          ; BCD B register
0096: 00 00         >70   rP        dw    0          ; BCD P register
0098: 00 00 00      >71   rC        ds    6          ; BCD Control (instruction) reg
009E: 00 00 00      >72   rA        ds    6          ; BCD A register
00A4: 00 00 00      >73   rR        ds    6          ; BCD R register
00AA: 00 00 00      >74   rD        ds    6          ; BCD D register
00B0: 00 00 00      >75   rD10      ds    6          ; BCD D10 reg (rD * 10)
00B6: 00 00 00      >76   CSW       ds    10         ; Control switches (0=off)
00C0: 00            >77   RUN       db    0          ; RUN mode/indicator (0=off)
00C1: 00            >78   ERR       db    0          ; ERR indicator (0=off)
00C2: 00            >79   COMP      db    0          ; Compare lo,eql,hi (<0,0,>0)
00C3: 00            >80   Ov        db    0          ; Overflow indicator (0=off)
00C4: 00            >81   Rp        db    0          ; Repeat indicator (0=off)
00C5: 00            >82   newp      db    0          ; "P changed manually" indicator
00C6: 00            >83   skipincP  db    0          ; Skip incP if PRB sign 6/7.
                    >84   B220end   equ   *          ; End of B220 simulated state
                    >85
                    >86   * Simulator page zero variables
                    >87
00C7: FF            >88   OvHlt     db    $FF        ; Oflow Halt (0=off)
00C8: 00 00         >89   instptr   dw    0          ; Pointer corresponding to rP
00CA: 00 00         >90   memptr    dw    0          ; Pointer to instruction data
00CC: 00 00         >91   ptr       dw    0          ; Utility pointer
00CE: 00 00         >92   inptr     dw    0          ; 'keyin' register label ptr
00D0: 00            >93   t1        db    0          ; Temp byte
00D1: 00            >94   NN        db    0          ; 2-digit BCD count
00D2: 64            >95   dispctr   db    dispcnt    ; Display refresh counter
00D3: 00 00         >96   linev     dw    0          ; Line base for decimal value
00D5: 00 00         >97   line1     dw    0          ; Line base for 1-bits
00D7: 00 00         >98   line2     dw    0          ; Line base for 2-bits
00D9: 00 00         >99   line4     dw    0          ; Line base for 4-bits
00DB: 00 00         >100  line8     dw    0          ; Line base for 8-bits
                    >101           dend
```

```
>103  ************************************************************
>104  *                                                          *
>105  *                  Macro Definitions                       *
>106  *                                                          *
>107  ************************************************************
>108
>109  seti     mac              ; Set indicator
>110           lda   #$FF
>111           sta   ]1         ; Set non-zero.
>112           eom
>113
>114  resi     mac              ; Reset indicator
>115           lda   #0
>116           sta   ]1         ; Zero indicator.
>117           eom
>118
>119           org   $800
>120           dsk   /ap/merlin/work/b220/b220sim
```

```
                >122  ************************************************************
                >123  *                                                          *
                >124  *                    Entry Point                           *
                >125  *                                                          *
                >126  ************************************************************
                >127
0800: 4C 57 0C  >128  B220SIM  jmp   init       ; Start simulator.
0803: 4C 91 0C  >129  RESTART  jmp   restart    ; Restart warm.
```

```
                182           put   B220KEYB
                >1    ************************************************************
                >2    *                                                          *
                >3    *                 Keyboard Input Routines                  *
                >4    *                                                          *
                >5    ************************************************************
                >6
0806: 8D 10 C0  >7    keyin   sta   KBSTROBE   ; Clear strobe.
0809: C9 A0     >8            cmp   #"         " ; Space bar?
080B: D0 3F     >9            bne   :bleep     ; -No, beep & continue.
                >10   ]stop   resi  RUN        ; -Yes, reset RUN mode
080D: A9 00     >10           lda   #0
080F: 85 C0     >10           sta   RUN        ; Zero indicator.
                >10           eom
0811: 20 28 0F  >11   :edit   jsr   display    ; Update B220 panel
                >12           resi  ERR        ; Reset ERR indicator
0814: A9 00     >12           lda   #0
0816: 85 C1     >12           sta   ERR        ; Zero indicator.
                >12           eom
0818: AD 00 C0  >13   :waitkey lda  KBD        ; Get a key.
081B: 10 FB     >14           bpl   :waitkey
081D: 8D 10 C0  >15           sta   KBSTROBE   ; Clear strobe
0820: C9 A0     >16           cmp   #"         " ; Space bar?
0822: F0 0E     >17           beq   :step      ; -Yes, step.
0824: C9 BF     >18           cmp   #"?"       ; Show help?
0826: F0 5F     >19           beq   :disphlp   ; -Yes, do it.
0828: 29 DF     >20           and   #$DF       ; Force upper case.
082A: C9 C7     >21           cmp   #"G"       ; G = Go?
082C: D0 24     >22           bne   :nx1       ; -No, analyze keypress.
                >23           seti  RUN        ; -Yes, set RUN mode
082E: A9 FF     >23           lda   #$FF
0830: 85 C0     >23           sta   RUN        ; Set non-zero.
                >23           eom
0832: A9 F2     >24   :step   lda   #"r"       ; Reset ERRlab on screen
0834: 8D 67 05  >25           sta   ERRlab
0837: A5 C5     >26           lda   newp       ; rP changed manually?
0839: D0 0A     >27           bne   :new       ; -Yes, re-fetch.
083B: A5 9B     >28           lda   rC+OP      ; -No, is OP a HLT?
083D: D0 10     >29           bne   :xeq       ; -No, execute current OP
083F: 20 14 0C  >30           jsr   incP       ; -Yes, skip HLT
0842: 4C 72 0B  >31           jmp   fetch      ;   and fetch next.
                >32
                >33   :new    resi  newp       ; Reset new P indicator
0845: A9 00     >33           lda   #0
0847: 85 C5     >33           sta   newp       ; Zero indicator.
                >33           eom
0849: 4C 54 0B  >34           jmp   newP       ;  and re-fetch.
                >35
084C: 20 DD FB  >36   :bleep  jsr   BEEP       ; Beep
084F: 4C C1 0B  >37   :xeq    jmp   ]contin    ; Execute current OP.
                >38
0852: C9 D1     >39   :nx1    cmp   #"Q"       ; Quit?
0854: D0 0B     >40           bne   :nx2       ; -No, continue.
0856: D8        >41           cld              ; -Yes, clear decimal
0857: 18        >42           clc              ;   and Carry.
0858: A9 00     >43           lda   #0         ; Set full-screen
085A: 85 22     >44           sta   WNDTOP     ;  text window,
085C: 68        >45           pla              ;  pop return
085D: 68        >46           pla              ;   address, and
085E: 4C D0 03  >47           jmp   DOSCON     ;    reconnect ProDOS.
                >48
0861: C9 D3     >49   :nx2    cmp   #"S"       ; Toggle switch?
0863: F0 28     >50           beq   :flipsw    ; -Yes.
0865: C9 C1     >51           cmp   #"A"       ; A register?
0867: F0 64     >52           beq   :inputA    ; -Yes, get input.
0869: C9 D2     >53           cmp   #"R"       ; R register?
086B: F0 64     >54           beq   :inputR    ; -Yes, get input.
```

```
086D: C9 C2    >55            cmp   #"B"       ; B register?
086F: F0 64    >56            beq   :inputB    ; -Yes, get input.
0871: C9 D0    >57            cmp   #"P"       ; P register?
0873: F0 68    >58            beq   :inputP    ; -Yes, get input.
0875: C9 C3    >59            cmp   #"C"       ; C register?
0877: F0 60    >60            beq   :inputC    ; -Yes, get input.
0879: C9 DA    >61            cmp   #"Z"       ; Reset?
087B: F0 39    >62            beq   :reset     ; -Yes, clear state.
087D: C9 C9    >63            cmp   #"I"       ; I/O configuration?
087F: F0 3F    >64            beq   :edio      ; -Yes, edit I/O config.
0881: 20 DD FB >65   :beep    jsr   BEEP       ; Unrecognized key, beep
0884: 4C 18 08 >66            jmp   :waitkey   ;  and get another key.
               >67
0887: 20 17 0F >68   :disphlp jsr   disphelp   ; Display help lines
088A: 4C 18 08 >69            jmp   :waitkey   ;  and get another key.
               >70
088D: A9 13    >71   :flipsw  lda   #$13       ; Set "Sw" label to inverse.
088F: 8D 53 05 >72            sta   SWlab
0892: A9 77    >73            lda   #$77
0894: 8D 54 05 >74            sta   SWlab+1
0897: 20 57 09 >75            jsr   getdig     ; Get digit or CR
089A: B0 0D    >76            bcs   :swdone    ; Done if CR.
089C: AA       >77            tax              ; -No, handle digit.
089D: B5 B6    >78            lda   CSW,x      ; Pick up switch,
089F: F0 04    >79            beq   :seti      ; -If reset, set it.
08A1: A9 00    >80            lda   #0         ; -If set, reset it.
08A3: F0 02    >81            beq   :store     ; (always)
               >82
08A5: A9 FF    >83   :seti    lda   #$FF
08A7: 95 B6    >84   :store   sta   CSW,x      ;   put it back.
08A9: A9 D3    >85   :swdone  lda   #"S"       ; Set "Sw" label to normal.
08AB: 8D 53 05 >86            sta   SWlab
08AE: A9 F7    >87            lda   #"w"
08B0: 8D 54 05 >88            sta   SWlab+1
08B3: 4C 11 08 >89   :ed      jmp   :edit
               >90
08B6: 20 7C 0C >91   :reset   jsr   reset      ; Reset B220 state
               >92            seti  newp       ; Force refetch.
08B9: A9 FF    >92            lda   #$FF
08BB: 85 C5    >92            sta   newp       ; Set non-zero.
               >92            eom
08BD: 4C B3 08 >93            jmp   :ed
               >94
08C0: 4C 15 0A >95   :edio    jmp   ediocfg    ; Relay jump
               >96
08C3: A0 00    >97   :indone  ldy   #0         ; Flip reg label to normal.
08C5: B1 CE    >98            lda   (inptr),y
08C7: 09 80    >99            ora   #$80
08C9: 91 CE    >100           sta   (inptr),y
08CB: D0 E6    >101           bne   :ed        ; (always)
               >102
08CD: A2 00    >103  :inputA  ldx   #Ain-intabl
08CF: B0 12    >104           bcs   :inreg     ; (always)
               >105
08D1: A2 10    >106  :inputR  ldx   #Rin-intabl
08D3: B0 0E    >107           bcs   :inreg     ; (always)
               >108
08D5: A2 04    >109  :inputB  ldx   #Bin-intabl
08D7: B0 0A    >110           bcs   :inreg     ; (always)
               >111
08D9: A2 08    >112  :inputC  ldx   #Cin-intabl
08DB: B0 06    >113           bcs   :inreg     ; (always)
               >114
08DD: A2 0C    >115  :inputP  ldx   #Pin-intabl
               >116           seti  newp       ; Signal manual rP change.
08DF: A9 FF    >116           lda   #$FF
08E1: 85 C5    >116           sta   newp       ; Set non-zero.
```

```
                >116            eom
                >117  *                       ;  and fall into :inreg.
                >118
                >119  * Input register value from keyboard
                >120  * Y = Hi (left) byte of register, X = # of bytes - 1
                >121
08E3: BD 1C 09  >122  :inreg    lda   intabl,x   ; Set inptr to reg label
08E6: 85 CE     >123            sta   inptr
08E8: BD 1D 09  >124            lda   intabl+1,x
08EB: 85 CF     >125            sta   inptr+1
08ED: BC 1E 09  >126            ldy   intabl+2,x ; Y = hi byte of reg
08F0: 8C 10 09  >127            sty   :ordig+1   ; Save register address
08F3: 8C 12 09  >128            sty   :stdig+1
08F6: BD 1F 09  >129            lda   intabl+3,x
08F9: AA        >130            tax              ; X = reg length - 1
08FA: A0 00     >131            ldy   #0
08FC: B1 CE     >132            lda   (inptr),y  ; Flip reg label to inverse.
08FE: 29 7F     >133            and   #$7F
0900: 91 CE     >134            sta   (inptr),y
0902: 20 57 09  >135  :getdig   jsr   getdig     ; Get digit or CR
0905: B0 BC     >136            bcs   :indone    ; CR ==> done.
0907: 48        >137            pha              ; Save digit
0908: AC 10 09  >138            ldy   :ordig+1   ; Restore Y
090B: 20 30 09  >139            jsr   shleft1    ; Shift register left 1 digit
090E: 68        >140            pla              ; Recover the digit
090F: 15 00     >141  :ordig    ora   0*0,x      ; OR in the low digit
0911: 95 00     >142  :stdig    sta   0*0,x      ;  and store it back.
0913: 8A        >143            txa              ; Save X
0914: 48        >144            pha
0915: 20 28 0F  >145            jsr   display    ; Update display
0918: 68        >146            pla              ; Restore X
0919: AA        >147            tax
091A: D0 E6     >148            bne   :getdig    ; (always)
                >149
                >150  intabl    equ   *          ; Table of reg input params
091C: 83 05     >151  Ain       dw    Alab       ; Address of "A" label
091E: 9E 05     >152            db    rA,6-1     ; Addr of hi digit, length-1
0920: AB 05     >153  Bin       dw    Blab
0922: 94 01     >154            db    rB,2-1
0924: BB 05     >155  Cin       dw    Clab
0926: 98 05     >156            db    rC,6-1
0928: B3 05     >157  Pin       dw    Plab
092A: 96 01     >158            db    rP,2-1
092C: 95 05     >159  Rin       dw    Rlab
092E: A4 05     >160            db    rR,6-1
```

```
                >162  ************************************************************
                >163  *                                                          *
                >164  *          Shift Register left 1 digit (4 bits)            *
                >165  *                                                          *
                >166  * Y = addr of Hi (left) byte of reg, X = byte length - 1   *
                >167  * X and Y are unchanged on exit.                           *
                >168  * High digit of sign byte of rA, rR, and rC is cleared.    *
                >169  *                                                          *
                >170  ************************************************************
                >171
0930: 8C 38 09  >172  shleft1  sty   :shift+1   ; Save register address
0933: 8A        >173           txa              ;  and byte length - 1.
0934: A0 04     >174           ldy   #4         ; Digit = 4 bits.
0936: 18        >175  :nxshift clc              ; Shift in zeroes.
0937: 36 00     >176  :shift   rol   0*0,x      ; Shift 1 bit
0939: CA        >177           dex              ;  for all bytes.
093A: 10 FB     >178           bpl   :shift
093C: AA        >179           tax              ; Restore X
093D: 88        >180           dey
093E: D0 F6     >181           bne   :nxshift   ; Shift 4 times.
0940: AC 38 09  >182           ldy   :shift+1   ; Restore Y = reg address.
0943: C0 96     >183           cpy   #rP        ; rP has no sign byte,
0945: F0 0C     >184           beq   :rts       ;  so skip it.
0947: C0 94     >185           cpy   #rB        ; rB has no sign byte,
0949: F0 08     >186           beq   :rts       ;  so skip it.
094B: B9 00 00  >187           lda   0,y        ; Clear high digit
094E: 29 0F     >188           and   #$0F       ;  of sign byte.
0950: 99 00 00  >189           sta   0,y
0953: 60        >190  :rts     rts
                >191
                >192  ************************************************************
                >193  *                                                          *
                >194  *                   Get Digit or CR                        *
                >195  *                                                          *
                >196  * On exit: If C = 0, A = digit value                       *
                >197  *          If C = 1, CR received                           *
                >198  *          X and Y unchanged.                              *
                >199  *                                                          *
                >200  ************************************************************
                >201
0954: 20 DD FB  >202  beepget  jsr   BEEP        ; Signal error key
0957: AD 00 C0  >203  getdig   lda   KBD         ; Get digit or <Enter>
095A: 10 FB     >204           bpl   getdig
095C: 8D 10 C0  >205           sta   KBSTROBE   ; Clear strobe
095F: C9 8D     >206           cmp   #$8D       ; <Enter>?
0961: F0 0B     >207           beq   :done      ; Yes, exit.
0963: C9 B0     >208           cmp   #"0"       ; -No, less than "0"?
0965: 90 ED     >209           bcc   beepget    ; -Yes, get another.
0967: C9 BA     >210           cmp   #"9"+1     ; -No, greater than "9"?
0969: B0 E9     >211           bcs   beepget    ; -Yes, get another.
096B: 29 0F     >212           and   #$0F       ; -No, isolate digit
096D: 18        >213           clc              ; C = 0 for digit
096E: 60        >214  :done    rts              ; C = 1 for CR.
```

```
                        >216  ************************************************************
                        >217  *                                                          *
                        >218  *            Edit B220SIM I/O Configuration                *
                        >219  *                                                          *
                        >220  ************************************************************
                        >221
                        >222  cursor   equ   $57        ; Mousetext checkerboard
                        >223  uparrow  equ   $8B        ; Up arrow
                        >224  dnarrow  equ   $8A        ; Down arrow
                        >225  ltarrow  equ   $88        ; Left arrow
                        >226  escape   equ   $9B        ; ESCAPE key
                        >227  delete   equ   $FF        ; DELETE key
                        >228  iocfgtt  equ   11         ; HTAB for screen title
                        >229  rtmargin equ   4          ; Right margin
                        >230  fnamecol equ   rtmargin+8 ; File name column
                        >231
                        >232  fnx      equ   linev      ; File name index (0..7)
                        >233  selected equ   linev+1    ; Selected index (0..7)
                        >234  selsave  equ   line1      ; Temp Y storage
                        >235  savex    equ   line1+1    ; Temp X storage
                        >236  selch    equ   line2      ; Selected fname cursor
                        >237  line     equ   line2+1    ; Line number (0..23)
                        >238  changed  equ   line4      ; File name changed flg
                        >239  selBASL  equ   line8      ; Selected line base (DA.DB)
                        >240
                        >241  iocfgstr equ   *          ; I/O Config Screen string
096F: C9 AF CF          >242           asc   "I/O Configuration",0D
0981: 0D                >243           db    $0D
0982: A0 D5 EE          >244           asc   " Unit    File pathname",0D
0999: AD AD AD          >245           asc   "------  ------------------------",0D
09BA: D0 D4 D2          >246           asc   "PTRDR0",01
09C1: D0 D4 D2          >247           asc   "PTRDR1",01
09C8: D0 D4 D0          >248           asc   "PTPCH0",01
09CF: D0 D4 D0          >249           asc   "PTPCH1",01
09D6: 0D                >250           db    $0D
09D7: CD D4 D5          >251           asc   "MTU0L0",01
09DE: CD D4 D5          >252           asc   "MTU0L1",01
09E5: CD D4 D5          >253           asc   "MTU1L0",01
09EC: CD D4 D5          >254           asc   "MTU1L1",01
09F3: 0D 0D 0D          >255           db    $0D,$0D,$0D,$0D,$0D
09F8: A0 A0 A0          >256           asc   "    ESC to return to B220SIM"
0A14: 00                >257           db    00         ; End of screen
                        >258
0A15: A2 00             >259  ediocfg  ldx   #0         ; Edit I/O Configuration
0A17: 86 22             >260           stx   WNDTOP     ; Set full screen.
0A19: 86 D4             >261           stx   selected   ; Select first file name.
0A1B: 20 58 FC          >262           jsr   HOME       ; Clear screen
0A1E: A2 00             >263  disiocfg ldx   #0         ; iocfgstr index = 0
0A20: 86 D3             >264           stx   fnx        ; fname index = 0
0A22: 86 D8             >265           stx   line       ; Line = 0
0A24: 8A                >266           txa
0A25: 20 C1 FB          >267           jsr   BASCALC    ; Set BASL for line 0
0A28: A0 0B             >268           ldy   #iocfgtt   ; HTAB to title
0A2A: BD 6F 09          >269  :nxch    lda   iocfgstr,x ; Next disp string char
0A2D: 10 06             >270           bpl   :command   ; -Command char if +
0A2F: 91 28             >271           sta   (BASL),y   ; -Display if not cmd.
0A31: C8                >272           iny              ; Advance CH
0A32: E8                >273  :advance inx              ; Advance str index
0A33: D0 F5             >274           bne   :nxch      ; (always)
                        >275
0A35: F0 48             >276  :command beq   :editfn    ; Screen complete, edit.
0A37: C9 0D             >277           cmp   #$0D       ; CR?
0A39: D0 0B             >278           bne   :fname     ; -No, insert file name.
0A3B: E6 D8             >279  :nxtline inc   line       ; -Yes, next line.
0A3D: A5 D8             >280           lda   line       ; Compute new line's
0A3F: 20 C1 FB          >281           jsr   BASCALC    ;  base addr (BASL)
0A42: A0 04             >282           ldy   #rtmargin  ; Set right margin.
```

```
0A44: 10 EC    >283           bpl   :advance   ; (always)
               >284
0A46: 86 D6    >285  :fname   stx   savex      ; Insert file name.
0A48: A9 BA    >286           lda   #":"       ; Insert punctuation.
0A4A: 91 28    >287           sta   (BASL),y
0A4C: A4 D3    >288           ldy   fnx
0A4E: C4 D4    >289           cpy   selected   ; This fname selected?
0A50: F0 01    >290           beq   :selectd   ; -Yes, C = selected.
0A52: 18       >291           clc              ; -No, /C = not selected.
0A53: BE F9 1D >292  :selectd ldx   fnxfn,y    ; Index into fnames
0A56: A0 0C    >293           ldy   #fnamecol  ; Y = 1st char of filename.
0A58: BD 00 03 >294  :nxtchar lda   fnames,x   ; Next file name char.
0A5B: F0 0C    >295           beq   :fndone    ; End of file name.
0A5D: 90 04    >296           bcc   :store     ; /C ==> keep normal.
0A5F: 20 3B 0B >297           jsr   inverse    ; C ==> make inverse
0A62: 38       >298           sec              ;  and stay selected.
0A63: 91 28    >299  :store   sta   (BASL),y   ; Display character
0A65: E8       >300           inx              ; Advance fnames index
0A66: C8       >301           iny              ; Advance CH
0A67: D0 EF    >302           bne   :nxtchar   ; (always)
               >303
0A69: E6 D3    >304  :fndone  inc   fnx        ; Advance fnames index
0A6B: A6 D6    >305           ldx   savex      ; Restore string index
0A6D: 90 CC    >306           bcc   :nxtline   ; Not selected ==> done.
0A6F: A9 57    >307           lda   #cursor    ; Selected ==> add cursor.
0A71: 91 28    >308           sta   (BASL),y
0A73: 84 D7    >309           sty   selch      ; Save cursor column.
0A75: A5 28    >310           lda   BASL       ; Save selected line base
0A77: 85 DB    >311           sta   selBASL
0A79: A5 29    >312           lda   BASL+1
0A7B: 85 DC    >313           sta   selBASL+1
0A7D: D0 BC    >314           bne   :nxtline   ; (always)
               >315
0A7F: A4 D7    >316  :editfn  ldy   selch      ; Cursor col of selected.
0A81: A9 00    >317           lda   #0         ; Mark unchanged.
0A83: 85 D9    >318           sta   changed
0A85: AD 00 C0 >319  :kbdloop lda   KBD        ; Read key and
0A88: 10 FB    >320           bpl   :kbdloop   ;  wait for keypress.
0A8A: 8D 10 C0 >321           sta   KBSTROBE   ; Clear keyboard strobe.
0A8D: A6 D4    >322           ldx   selected   ; Save index of currently
0A8F: 86 D5    >323           stx   selsave    ;  selected file name.
0A91: C9 8B    >324           cmp   #uparrow
0A93: D0 58    >325           bne   :notup
0A95: C6 D4    >326           dec   selected   ; Move cursor up
0A97: A5 D4    >327           lda   selected   ;  and wrap around.
0A99: 29 07    >328           and   #7
0A9B: 85 D4    >329           sta   selected
0A9D: A9 A0    >330  :edited  lda   #"         " ; Blank out cursor
0A9F: A4 D7    >331           ldy   selch
0AA1: 91 DB    >332           sta   (selBASL),y
0AA3: A5 D9    >333           lda   changed    ; Fname changed?
0AA5: F0 2F    >334           beq   :chkexit   ; -No, exit or resdiplay.
0AA7: A4 D5    >335           ldy   selsave    ; -Yes, get selected index
0AA9: BE F9 1D >336           ldx   fnxfn,y    ; -Yes, commit new
0AAC: A0 0C    >337           ldy   #fnamecol  ;   file name.
0AAE: C4 D7    >338  :copy    cpy   selch      ; End of file name?
0AB0: F0 11    >339           beq   :fnend     ; -Yes.
0AB2: B1 DB    >340           lda   (selBASL),y
0AB4: 09 80    >341           ora   #$80       ; -No. Make normal.
0AB6: C9 A0    >342           cmp   #$A0       ; Upper case?
0AB8: B0 02    >343           bcs   :norm      ; -No, already normal.
0ABA: 09 40    >344           ora   #$40       ; -Yes, make normal.
0ABC: 9D 00 03 >345  :norm    sta   fnames,x
0ABF: E8       >346           inx
0AC0: C8       >347           iny
0AC1: D0 EB    >348           bne   :copy      ; (always)
               >349
```

```
0AC3: A9 00    >350  :fnend   lda   #0         ; Null at end
0AC5: 9D 00 03 >351           sta   fnames,x   ;  of fname.
0AC8: A4 D5    >352           ldy   selsave    ; Zero file offset
0ACA: BE F1 1D >353           ldx   fnxoff,y   ;  for new file.
0ACD: 9D DD 1D >354           sta   rdroff,x
0AD0: 9D DE 1D >355           sta   rdroff+1,x
0AD3: 9D DF 1D >356           sta   rdroff+2,x
0AD6: AD 00 C0 >357  :chkexit lda   KBD        ; Check last key.
0AD9: C9 1B    >358           cmp   #escape&$7F ; Was it ESCAPE?
0ADB: F0 03    >359           beq   :restart   ; -Yes, back to sim.
0ADD: 4C 1E 0A >360  :disiocr jmp   disiocfg   ; Redisplay & continue.
               >361
0AE0: 4C 91 0C >362  :restart jmp   restart    ; Restart B220SIM.
               >363
0AE3: 84 D5    >364  :beep    sty   selsave    ; Scratch to save Y.
0AE5: 20 DD FB >365           jsr   BEEP       ; Signal invalid key
0AE8: A4 D5    >366           ldy   selsave    ; Restore Y
0AEA: 4C 85 0A >367  :kbdlpr  jmp   :kbdloop   ;  and continue.
               >368
0AED: C9 8A    >369  :notup   cmp   #dnarrow
0AEF: F0 04    >370           beq   :down
0AF1: C9 8D    >371           cmp   #$8D
0AF3: D0 0A    >372           bne   :notdown   ; Not down arrow or return.
0AF5: E6 D4    >373  :down    inc   selected   ; Move cursor down
0AF7: A5 D4    >374           lda   selected   ;  and wrap around.
0AF9: 29 07    >375           and   #7
0AFB: 85 D4    >376           sta   selected
0AFD: 10 9E    >377           bpl   :edited    ; (always)
               >378
0AFF: C9 9B    >379  :notdown cmp   #escape    ; ESC?
0B01: F0 9A    >380           beq   :edited    ; -Yes, commit fname.
0B03: C9 88    >381           cmp   #ltarrow   ; Left arrow?
0B05: F0 04    >382           beq   :backsp    ; -Yes, backspace.
0B07: C9 FF    >383           cmp   #delete    ; DELETE?
0B09: D0 13    >384           bne   :addchar   ; -No, add character.
0B0B: C0 0C    >385  :backsp  cpy   #fnamecol  ; At start?
0B0D: F0 D4    >386           beq   :beep      ; -Yes, complain.
0B0F: A9 A0    >387           lda   #"         " ; -No, blank cursor
0B11: 91 DB    >388           sta   (selBASL),y
0B13: 88       >389           dey              ; Back up.
0B14: A9 57    >390  :changed lda   #cursor    ; Place cursor.
0B16: 91 DB    >391           sta   (selBASL),y
0B18: 84 D7    >392           sty   selch      ; Save cursor column.
0B1A: 85 D9    >393           sta   changed    ; Mark changed & cont.
0B1C: D0 CC    >394           bne   :kbdlpr    ; (always)
               >395
0B1E: A6 D9    >396  :addchar ldx   changed    ; Any prior change?
0B20: D0 0D    >397           bne   :notfrst   ; -Yes, just add char.
0B22: AA       >398           tax              ; Save character.
0B23: A9 A0    >399           lda   #"         " ; Blank out file name.
0B25: C0 0C    >400  :cloop   cpy   #fnamecol
0B27: F0 05    >401           beq   :addit
0B29: 91 DB    >402           sta   (selBASL),y
0B2B: 88       >403           dey
0B2C: D0 F7    >404           bne   :cloop     ; (always)
               >405
0B2E: 8A       >406  :addit   txa              ; Restore character.
0B2F: C0 24    >407  :notfrst cpy   #fnamecol+24 ; At end?
0B31: B0 B0    >408           bcs   :beep      ; -Yes, complain.
0B33: 20 3B 0B >409           jsr   inverse    ; -No, make inverse.
0B36: 91 DB    >410           sta   (selBASL),y ;  and add to fname.
0B38: C8       >411           iny              ; Advance CH
0B39: D0 D9    >412           bne   :changed   ; (always)
               >413
0B3B: 29 7F    >414  inverse  and   #$7F       ; Make inverse
0B3D: C9 40    >415           cmp   #$40       ; Upper case?
0B3F: 90 06    >416           bcc   :rts       ; -No, special char.
```

```
0B41: C9 60    >417           cmp   #$60       ; Upper case?
0B43: B0 02    >418           bcs   :rts       ; -No, lower case.
0B45: 29 1F    >419           and   #$1F       ; -Yes, make inverse
0B47: 60       >420  :rts     rts
```

```
                183           put   B220FETCH
               >1    ************************************************************
               >2    *                                                          *
               >3    *            Simulate next B220 Instruction                *
               >4    *                                                          *
               >5    ************************************************************
               >6
0B48: 4C 40 0C >7    ADDRerrR jmp   ADDRerr    ; Relay branch
0B4B: 4C 4A 0C >8    UNDIGerR jmp   UNDIGerr   ; Relay branch
0B4E: 4C 06 08 >9    keyinR   jmp   keyin      ; Relay branch
0B51: 4C 0D 08 >10   stopR    jmp   ]stop      ; Relay branch
               >11
               >12   * Convert rP to instruction address
               >13
0B54: A6 97    >14   newP     ldx   rP+1        ; Low 2 BCD digits of rP
0B56: E0 9A    >15           cpx   #$99+1      ; Undigits?
0B58: B0 F1    >16           bcs   UNDIGerR    ; -Yes, error.
0B5A: A4 96    >17           ldy   rP          ; High 2 BCD digits of rP
0B5C: C0 4A    >18           cpy   #$49+1      ; ADDR error?
0B5E: B0 E8    >19           bcs   ADDRerrR    ; -Yes, stop.
0B60: BD 8B 1E >20           lda   BCDLadrl,x  ; -No, compute 'instptr'
0B63: 79 BF 1F >21           adc   BCDHadrl,y
0B66: 85 C8    >22           sta   instptr     ; Low byte of instr address
0B68: BD 25 1F >23           lda   BCDLadrh,x
0B6B: 79 09 20 >24           adc   BCDHadrh,y
0B6E: B0 DB    >25           bcs   UNDIGerR    ; Carry out ==> undigit(s)
0B70: 85 C9    >26           sta   instptr+1   ; High byte of instr address
0B72: A0 00    >27   fetch    ldy   #0          ; Fetch next instruction.
0B74: 84 C6    >28           sty   skipincP    ; Don't skip incP
0B76: B1 C8    >29           lda   (instptr),y
0B78: 85 98    >30           sta   rC+S        ; Sign
0B7A: C8       >31           iny
0B7B: B1 C8    >32           lda   (instptr),y
0B7D: 85 99    >33           sta   rC+sL       ; (field) start, Length
0B7F: C8       >34           iny
0B80: B1 C8    >35           lda   (instptr),y
0B82: 85 9A    >36           sta   rC+VV       ; Variants
0B84: C8       >37           iny
0B85: B1 C8    >38           lda   (instptr),y
0B87: 85 9B    >39           sta   rC+OP       ; OPcode
0B89: C8       >40           iny
0B8A: B1 C8    >41           lda   (instptr),y
0B8C: 85 9C    >42           sta   rC+ADDR     ; High 2 digits of ADDR
0B8E: C8       >43           iny
0B8F: B1 C8    >44           lda   (instptr),y
0B91: 85 9D    >45           sta   rC+ADDR+1   ; Low 2 digits of ADDR
0B93: A5 98    >46   execute  lda   rC+S        ; Is Sign negative?
0B95: 29 01    >47           and   #1
0B97: F0 0F    >48           beq   :noBmod     ; -No, skip rB modification
0B99: F8       >49           sed               ; / Decimal mode
0B9A: 18       >50           clc
0B9B: A5 9D    >51           lda   rC+ADDR+1   ; Add rB to rC+ADDR
0B9D: 65 95    >52           adc   rB+1
0B9F: 85 9D    >53           sta   rC+ADDR+1
0BA1: A5 9C    >54           lda   rC+ADDR
0BA3: 65 94    >55           adc   rB
0BA5: 85 9C    >56           sta   rC+ADDR
0BA7: D8       >57           cld               ; \ Back to binary mode
0BA8: AD 00 C0 >58   :noBmod  lda   KBD         ; User interaction?
0BAB: 30 A1    >59           bmi   keyinR      ; -Yes, handle it.
0BAD: A5 C0    >60           lda   RUN         ; RUN mode off
0BAF: 25 9B    >61           and   rC+OP       ;  or HLT instruction?
0BB1: F0 9E    >62           beq   stopR       ; -Yes, stop.
0BB3: 8D 30 C0 >63           sta   SPKR        ; -No, toggle speaker.
0BB6: C6 D2    >64           dec   dispctr     ; Update display every
0BB8: 10 07    >65           bpl   ]contin     ;  'dispcnt' instructions.
0BBA: A9 64    >66           lda   #dispcnt    ; Reset counter
```

```
0BBC: 85 D2    >67            sta   dispctr
0BBE: 20 28 0F >68            jsr   display
0BC1: A4 9B    >69   ]contin  ldy   rC+OP       ; Op code
0BC3: C0 60    >70            cpy   #$60        ; OP out of range?
0BC5: B0 6D    >71            bcs   OPerr       ; -Yes, stop.
0BC7: A5 C3    >72            lda   Ov          ; -No, is Overflow set
0BC9: 25 C7    >73            and   OvHlt       ;   and Ovflo Halt mode?
0BCB: F0 04    >74            beq   :ok         ; -No, continue.
0BCD: C0 31    >75            cpy   #$31        ; -Yes, is OP BOF?
0BCF: D0 67    >76            bne   OFLerr      ; -No, Overflow error.
0BD1: A5 C6    >77   :ok      lda   skipincP    ; -Yes, skip increment P?
0BD3: D0 03    >78            bne   :skip       ; -Yes, PRB hit sign 6/7.
0BD5: 20 14 0C >79            jsr   incP        ; -No, inc rP and instptr.
0BD8: B9 59 10 >80   :skip    lda   optabl,y    ; Get execute address.
0BDB: 8D 0B 0C >81            sta   :go+1
0BDE: B9 B3 10 >82            lda   optabh,y    ; High bit set?
0BE1: 30 2A    >83            bmi   :noADDR     ; -Yes, ignore ADDR
0BE3: 8D 0C 0C >84            sta   :go+2       ; -No, save execute address
0BE6: A6 9D    >85            ldx   rC+ADDR+1   ; Low 2 BCD ADDR digits
0BE8: E0 9A    >86            cpx   #$99+1      ; Undigits?
0BEA: B0 5E    >87            bcs   UNDIGerr    ; -Yes, error.
0BEC: A4 9C    >88            ldy   rC+ADDR     ; High 2 BCD ADDR digits
0BEE: C0 4A    >89            cpy   #$49+1      ; ADDR error?
0BF0: B0 4E    >90            bcs   ADDRerr     ; -Yes, stop.
0BF2: BD 8B 1E >91            lda   BCDLadrl,x  ; -No, compute 'memptr'
0BF5: 79 BF 1F >92            adc   BCDHadrl,y
0BF8: 85 CA    >93            sta   memptr      ; Low byte of memory address
0BFA: BD 25 1F >94            lda   BCDLadrh,x
0BFD: 79 09 20 >95            adc   BCDHadrh,y
0C00: B0 48    >96            bcs   UNDIGerr    ; Carry out ==> undigit(s).
0C02: 85 CB    >97            sta   memptr+1    ; High byte of memory address
0C04: A0 00    >98            ldy   #0          ; Enter execute with Y=0
0C06: B1 CA    >99            lda   (memptr),y  ;  & operand sign in A & rD+S.
0C08: 85 AA    >100           sta   rD+S
0C0A: 4C 00 00 >101  :go      jmp   0*0         ; Go to execute routine.
               >102
0C0D: 29 7F    >103  :noADDR  and   #$7F        ; Turn off "noADDR" bit
0C0F: 8D 0C 0C >104           sta   :go+2       ;  and save execute address.
0C12: D0 F6    >105           bne   :go         ; (always)
               >106
               >107  * Increment rP and instptr
               >108
0C14: F8       >109  incP     sed               ; / BCD mode arithmetic
0C15: 18       >110           clc
0C16: A5 97    >111           lda   rP+1        ; Increment rP by 1
0C18: 69 01    >112           adc   #1
0C1A: 85 97    >113           sta   rP+1
0C1C: 90 0A    >114           bcc   :nocar      ; Hi digits don't change.
0C1E: A5 96    >115           lda   rP          ; Propagate carry.
0C20: 69 00    >116           adc   #0
0C22: 85 96    >117           sta   rP
0C24: C9 4A    >118           cmp   #$49+1      ; Did we pass 4999?
0C26: B0 18    >119           bcs   ADDRerr     ; -Yes, ADDR error.
0C28: D8       >120  :nocar   cld               ; \ Back to binary.
0C29: A5 C8    >121           lda   instptr     ; Inc 'instptr' by 6
0C2B: 69 06    >122           adc   #6
0C2D: 85 C8    >123           sta   instptr
0C2F: 90 02    >124           bcc   :nocarry
0C31: E6 C9    >125           inc   instptr+1
0C33: 60       >126  :nocarry rts
```

```
                >128  * B220 error routines
                >129
0C34: A9 CF     >130  OPerr     lda   #"O"       ; OPcode error
0C36: D0 14     >131            bne   ]err       ; (always)
                >132
0C38: A9 D6     >133  OFLerr    lda   #"V"       ; Overflow error
0C3A: D0 10     >134            bne   ]err       ; (always)
                >135
0C3C: A9 C6     >136  FIELDerr  lda   #"F"       ; Field error
0C3E: D0 0C     >137            bne   ]err       ; (always)
                >138
0C40: A9 C1     >139  ADDRerr   lda   #"A"       ; Address error
0C42: D0 08     >140            bne   ]err       ; (always)
                >141
0C44: 85 00     >142  IOerr     sta   0          ; Save I/O err code
0C46: A9 C9     >143            lda   #"I"       ; I/O error
0C48: D0 02     >144            bne   ]err
                >145
0C4A: A9 D8     >146  UNDIGerr  lda   #"X"       ; Non-BCD digit error
0C4C: 8D 67 05  >147  ]err      sta   ERRlab     ; Show on screen.
0C4F: 85 C1     >148            sta   ERR        ; Set error indicator,
0C51: 20 DD FB  >149            jsr   BEEP       ;  sound beep,
0C54: 4C 0D 08  >150            jmp   ]stop      ;   and stop...
```

```
                >152  ************************************************************
                >153  *                                                          *
                >154  *                    Initialize B220                       *
                >155  *                                                          *
                >156  ************************************************************
                >157
0C57: A0 C8     >158  init     ldy   #fnend-fnametbl ; Move fnames to $300.
0C59: B9 82 20  >159  :fnloop  lda   fnametbl,y
0C5C: 99 00 03  >160           sta   fnames,y
0C5F: 88        >161           dey
0C60: C0 FF     >162           cpy   #$FF       ; Loop until
0C62: D0 F5     >163           bne   :fnloop    ;  Y underflows.
0C64: A9 D0     >164           lda   #<MEM      ; Initialize B220 memory to 0
0C66: 85 CA     >165           sta   memptr
0C68: A9 20     >166           lda   #>MEM
0C6A: 85 CB     >167           sta   memptr+1
0C6C: A0 00     >168           ldy   #0
0C6E: 98        >169  :loop    tya
0C6F: 91 CA     >170  :pagloop sta   (memptr),y
0C71: C8        >171           iny
0C72: D0 FB     >172           bne   :pagloop
0C74: E6 CB     >173           inc   memptr+1
0C76: A5 CB     >174           lda   memptr+1
0C78: C9 96     >175           cmp   #>$9600
0C7A: 90 F2     >176           bcc   :loop
0C7C: A2 36     >177  reset    ldx   #B220end-B220strt-1 ; Clear B220 state
0C7E: A9 00     >178           lda   #0
0C80: 95 90     >179  :regloop sta   B220strt,x
0C82: CA        >180           dex
0C83: 10 FB     >181           bpl   :regloop
0C85: A2 13     >182           ldx   #IOstend-IOstate-1 ; Rewind paper
0C87: 9D DD 1D  >183  :offlp   sta   IOstate,x  ; tapes and mag tapes.
0C8A: CA        >184           dex
0C8B: 10 FA     >185           bpl   :offlp
                >186           seti  OvHlt      ; Set Ovflow Halt mode.
0C8D: A9 FF     >186           lda   #$FF
0C8F: 85 C7     >186           sta   OvHlt      ; Set non-zero.
                >186           eom
0C91: 20 04 0D  >187  restart  jsr   disppanl   ; Init screen for B220
0C94: 20 28 0F  >188           jsr   display    ;  panel & display state.
0C97: 4C 54 0B  >189           jmp   newP       ; Start simulation.
```

```
 184            put   B220PANEL
>1     ************************************************************
>2     *                                                          *
>3     *           B220 front panel display routines              *
>4     *                                                          *
>5     ************************************************************
>6
>7     off      equ   " "        ; blank (neon off)
>8     on       equ   "*"        ; asterisk (neon on)
>9
>10    AR8      equ   $580       ; Line 4
>11    AR4      equ   $600       ; Line 5
>12    AR2      equ   $680       ; Line 6
>13    AR1      equ   $700       ; Line 7
>14    ARv      equ   $428       ; Line 9
>15    BPC8     equ   $5A8       ; Line 12
>16    BPC4     equ   $628       ; Line 13
>17    BPC2     equ   $6A8       ; Line 14
>18    BPC1     equ   $728       ; Line 15
>19    BPCv     equ   $450       ; Line 17
>20    STATlin  equ   $550       ; Line 19
>21
>22    B220col  equ   13-1       ; Leftmost title column
>23    Acol     equ   6-1        ; Leftmost digit column of A
>24    Rcol     equ   24-1       ; Leftmost digit column of R
>25    Bcol     equ   6-1        ; Leftmost digit column of B
>26    Pcol     equ   14-1       ; Leftmost digit column of P
>27    Ccol     equ   22-1       ; Leftmost digit column of C
>28    SW1col   equ   7-1        ; SW 1 column
>29    RUNcol   equ   18-1       ; RUN column
>30    ERRcol   equ   22-1       ; ERR column
>31    COMPcol  equ   26-1       ; COMP column
>32    OFLcol   equ   32-1       ; OFL column
>33    RPTcol   equ   35-1       ; RPT column
>34
>35    * Register label addresses
>36
>37    Alab     equ   AR8+3
>38    Rlab     equ   AR8+21
>39    Blab     equ   BPC8+3
>40    Plab     equ   BPC8+11
>41    Clab     equ   BPC8+19
>42    SWlab    equ   STATlin+3
>43    ERRlab   equ   STATlin+ERRcol+2 ; Error type character
```

```
                        >45   * Register front panel attributes
                        >46
0C9A: 2D 04 05          >47   Aattr    dw    ARv+Acol,AR1+Acol,AR2+Acol,AR4+Acol,AR8+Acol
0CA4: A3                >48           db    rA+5        ; Low byte of rA
0CA5: 0B                >49           db    12-1        ; Display columns - 1
0CA6: 01 00 01          >50           db    1,0,1,1,1,1,1,1,1,1,1,1 ; Column mask
0CB2: 3F 04 17          >51   Rattr    dw    ARv+Rcol,AR1+Rcol,AR2+Rcol,AR4+Rcol,AR8+Rcol
0CBC: A9                >52           db    rR+5        ; Low byte of rR
0CBD: 0B                >53           db    12-1        ; Display columns - 1
0CBE: 01 00 01          >54           db    1,0,1,1,1,1,1,1,1,1,1,1 ; Column mask
0CCA: 55 04 2D          >55   Battr    dw    BPCv+Bcol,BPC1+Bcol,BPC2+Bcol,BPC4+Bcol,BPC8+Bcol
0CD4: 95                >56           db    rB+1        ; Low byte of rB
0CD5: 03                >57           db    4-1         ; Display columns - 1
0CD6: 01 01 01          >58           db    1,1,1,1     ; Column mask
0CDA: 5D 04 35          >59   Pattr    dw    BPCv+Pcol,BPC1+Pcol,BPC2+Pcol,BPC4+Pcol,BPC8+Pcol
0CE4: 97                >60           db    rP+1        ; Low byte of rP
0CE5: 03                >61           db    4-1         ; Display columns - 1
0CE6: 01 01 01          >62           db    1,1,1,1     ; Column mask
0CEA: 65 04 3D          >63   Cattr    dw    BPCv+Ccol,BPC1+Ccol,BPC2+Ccol,BPC4+Ccol,BPC8+Ccol
0CF4: 9D                >64           db    rC+5        ; Low byte of rC
0CF5: 0D                >65           db    14-1        ; Display columns - 1
0CF6: 01 00 01          >66           db    1,0,1,1,1,1,0,1,1,0,1,1,1,1 ; Column mask
```

```
                >68   ************************************************************
                >69   *                                                          *
                >70   *              Initialize B220 Front Panel                 *
                >71   *                                                          *
                >72   ************************************************************
                >73
0D04: D8        >74   disppanl cld             ; Force binary mode.
0D05: A9 15     >75             lda   #21       ; Disable 80-col firmware
0D07: 20 ED FD  >76             jsr   COUT
0D0A: A9 00     >77             lda   #0
0D0C: 85 22     >78             sta   WNDTOP    ; Set full-screen window.
0D0E: 20 58 FC  >79             jsr   HOME      ; Clear 40-col screen
0D11: 8D 0F C0  >80             sta   ALTCHAR   ; Select alternate charset
0D14: A2 0B     >81             ldx   #B220col-1
0D16: 20 4A F9  >82             jsr   PRBL2     ; Space to starting column
0D19: A0 00     >83             ldy   #0
0D1B: B9 BF 0D  >84   :titloop  lda   B220msg,y ; Display title and AR top border
0D1E: F0 06     >85             beq   :AR
0D20: 20 ED FD  >86             jsr   COUT
0D23: C8        >87             iny
0D24: D0 F5     >88             bne   :titloop  ; (always)
                >89
0D26: 20 95 0D  >90   :AR       jsr   disARmid  ; Display 8-bit line
0D29: 20 95 0D  >91             jsr   disARmid  ; Display 4-bit line
0D2C: 20 95 0D  >92             jsr   disARmid  ; Display 2-bit line
0D2F: 20 95 0D  >93             jsr   disARmid  ; Display 1-bit line
0D32: A0 00     >94             ldy   #0
0D34: B9 D4 0D  >95   :ARborlp  lda   ARbord,y  ; Display AR bottom border
0D37: F0 06     >96             beq   :BPC
0D39: 20 ED FD  >97             jsr   COUT
0D3C: C8        >98             iny
0D3D: D0 F5     >99             bne   :ARborlp  ; (always)
                >100
0D3F: 20 8D 0D  >101  :BPC      jsr   blanklin  ; <blank line for reg values>
0D42: 20 8D 0D  >102            jsr   blanklin  ; <blank line>
0D45: 20 A3 0D  >103            jsr   disBPCbo  ; Display BPC top border
0D48: 20 B1 0D  >104            jsr   disBPCmi  ; Display 8-bit line
0D4B: 20 B1 0D  >105            jsr   disBPCmi  ; Display 4-bit line
0D4E: 20 B1 0D  >106            jsr   disBPCmi  ; Display 2-bit line
0D51: 20 B1 0D  >107            jsr   disBPCmi  ; Display 1-bit line
0D54: 20 A3 0D  >108            jsr   disBPCbo  ; Display BPC bottom border
0D57: 20 8D 0D  >109            jsr   blanklin  ; <blank line for values>
0D5A: 20 8D 0D  >110            jsr   blanklin  ; <blank line>
0D5D: A0 00     >111            ldy   #0        ; Display Status & Help lines
0D5F: B9 6C 0E  >112  :STATlp   lda   STAT,y
0D62: F0 06     >113            beq   :finish
0D64: 20 ED FD  >114            jsr   COUT
0D67: C8        >115            iny
0D68: D0 F5     >116            bne   :STATlp   ; (always)
                >117
0D6A: A9 81     >118  :finish   lda   #$81      ; "A" label
0D6C: 8D 83 05  >119            sta   Alab
0D6F: A9 82     >120            lda   #$82      ; "B" label
0D71: 8D AB 05  >121            sta   Blab
0D74: A9 83     >122            lda   #$83      ; "C" label
0D76: 8D BB 05  >123            sta   Clab
0D79: A9 90     >124            lda   #$90      ; "P" label
0D7B: 8D B3 05  >125            sta   Plab
0D7E: A9 92     >126            lda   #$92      ; "R" label
0D80: 8D 95 05  >127            sta   Rlab
0D83: A9 93     >128            lda   #$93      ; "S" of "Sw"
0D85: 8D 53 05  >129            sta   SWlab
0D88: A9 14     >130            lda   #20       ; Window is last 4 lines.
0D8A: 85 22     >131            sta   WNDTOP
0D8C: 60        >132            rts
                >133
0D8D: A9 A0     >134  blanklin  lda   #"        " ; Separate CRs with blank
```

```
0D8F: 20 ED FD >135           jsr   COUT
0D92: 4C 8E FD >136           jmp   CROUT
               >137
0D95: A0 00    >138  disARmid ldy   #0         ; Display AR middle line
0D97: B9 FA 0D >139  :loop    lda   ARmid,y
0D9A: F0 06    >140           beq   :rts
0D9C: 20 ED FD >141           jsr   COUT
0D9F: C8       >142           iny
0DA0: D0 F5    >143           bne   :loop      ; (always)
               >144
0DA2: 60       >145  :rts     rts
               >146
0DA3: A0 00    >147  disBPCbo ldy   #0         ; Display BPC border
0DA5: B9 20 0E >148  :loop    lda   BPCbord,y
0DA8: F0 06    >149           beq   :rts
0DAA: 20 ED FD >150           jsr   COUT
0DAD: C8       >151           iny
0DAE: D0 F5    >152           bne   :loop      ; (always)
               >153
0DB0: 60       >154  :rts     rts
               >155
0DB1: A0 00    >156  disBPCmi ldy   #0         ; Display BPC middle line
0DB3: B9 46 0E >157  :loop    lda   BPCmid,y
0DB6: F0 06    >158           beq   :rts
0DB8: 20 ED FD >159           jsr   COUT
0DBB: C8       >160           iny
0DBC: D0 F5    >161           bne   :loop      ; (always)
               >162
0DBE: 60       >163  :rts     rts
               >164
0DBF: C2 F5 F2 >165  B220msg  asc   "Burroughs 220 v1.1"8DA08D
0DD4: A0 A0 A0 >166  ARbord   asc   "    +-+----------+    +-+----------+",8D00
0DFA: A0 A0 A0 >167  ARmid    asc   "    | |          |    | |          |",8D00
0E20: A0 A0 A0 >168  BPCbord  asc   "    +----+  +----+  +-+----+--+----+",8D00
0E46: A0 A0 A0 >169  BPCmid   asc   "    |    |  |    |  | |    |  |    |",8D00
0E6C: A0 A0 A0 >170  STAT     asc   "   Sw 0123456789 Run Err < = > Ov Rp",8DA08D
0E93: A0 D3 F4 >171  Help1    asc   " Stop/Step: <space>, Go: G, Reset: Z",8D
0EB8: A0 D3 E5 >172  Help2    asc   " Set reg: A/R/B/P/C + digits + Return",8D
0EDE: A0 D4 EF >173  Help3    asc   " Toggle switch: S + digit, Help: ?",8D
0F01: A0 C9 AF >174  Help4    asc   " I/O Configuration: I",00
               >175
0F17: 20 58 FC >176  disphelp jsr   HOME        ; Display help lines.
0F1A: A0 00    >177           ldy   #0         ; (window is last 4 lines)
0F1C: B9 93 0E >178  :helplp  lda   Help1,y
0F1F: F0 06    >179           beq   :done
0F21: 20 ED FD >180           jsr   COUT
0F24: C8       >181           iny
0F25: D0 F5    >182           bne   :helplp    ; (always)
               >183
0F27: 60       >184  :done    rts
```

```
                        >186  ************************************************************
                        >187  *                                                          *
                        >188  *                    Display B220 State                    *
                        >189  *                                                          *
                        >190  ************************************************************
                        >191
0F28: 20 3A 0F          >192  display  jsr   dispA      ; Display A
0F2B: 20 41 0F          >193           jsr   dispR      ; Display R
0F2E: 20 48 0F          >194           jsr   dispB      ; Display B
0F31: 20 4F 0F          >195           jsr   dispP      ; Display P
0F34: 20 56 0F          >196           jsr   dispC      ; Display C
0F37: 4C 5D 0F          >197           jmp   dispSTAT   ; Disp Status & return.
                        >198
0F3A: A9 9A             >199  dispA    lda   #<Aattr    ; Register A attributes
0F3C: A0 0C             >200           ldy   #>Aattr
0F3E: 4C EA 0F          >201           jmp   dispreg    ; Display the register.
                        >202
0F41: A9 B2             >203  dispR    lda   #<Rattr    ; Register R attributes
0F43: A0 0C             >204           ldy   #>Rattr
0F45: 4C EA 0F          >205           jmp   dispreg    ; Display the register.
                        >206
0F48: A9 CA             >207  dispB    lda   #<Battr    ; Register B attributes
0F4A: A0 0C             >208           ldy   #>Battr
0F4C: 4C EA 0F          >209           jmp   dispreg    ; Display the register.
                        >210
0F4F: A9 DA             >211  dispP    lda   #<Pattr    ; Register P attributes
0F51: A0 0C             >212           ldy   #>Pattr
0F53: 4C EA 0F          >213           jmp   dispreg    ; Display the register.
                        >214
0F56: A9 EA             >215  dispC    lda   #<Cattr    ; Register C attributes
0F58: A0 0C             >216           ldy   #>Cattr
0F5A: 4C EA 0F          >217           jmp   dispreg    ; Display the register.
                        >218
0F5D: A9 50             >219  dispSTAT lda   #<STATlin  ; Set ptr to STATlin
0F5F: 85 CC             >220           sta   ptr
0F61: A9 05             >221           lda   #>STATlin
0F63: 85 CD             >222           sta   ptr+1
0F65: A2 00             >223           ldx   #0
0F67: A0 06             >224           ldy   #SW1col    ; Start at switch 1
0F69: B5 B6             >225  :swloop  lda   CSW,x      ; Is it on?
0F6B: 20 C1 0F          >226           jsr   INDshow    ; Display it's state
0F6E: E8                >227           inx              ; Next switch
0F6F: E0 0A             >228           cpx   #10        ; Until done...
0F71: 90 F6             >229           bcc   :swloop
0F73: A0 11             >230           ldy   #RUNcol
0F75: A5 C0             >231           lda   RUN
0F77: 20 C1 0F          >232           jsr   INDshow
0F7A: A0 15             >233           ldy   #ERRcol
0F7C: A5 C1             >234           lda   ERR
0F7E: 20 C1 0F          >235           jsr   INDshow
0F81: A0 19             >236           ldy   #COMPcol
0F83: A5 C2             >237           lda   COMP       ; <0, 0, >0: < = >
0F85: 30 07             >238           bmi   :lt
0F87: F0 0A             >239           beq   :eq
0F89: A2 0C             >240           ldx   #:gtstr-:ltstr ; Point to > string
0F8B: 4C 95 0F          >241           jmp   :show
                        >242
0F8E: A2 00             >243  :lt      ldx   #:ltstr-:ltstr ; Point to < string
0F90: 4C 95 0F          >244           jmp   :show
                        >245
0F93: A2 06             >246  :eq      ldx   #:eqstr-:ltstr ; Point to = string
0F95: BD AF 0F          >247  :show    lda   :ltstr,x
0F98: F0 06             >248           beq   :next
0F9A: 91 CC             >249           sta   (ptr),y
0F9C: C8                >250           iny
0F9D: E8                >251           inx
0F9E: D0 F5             >252           bne   :show      ; (always)
```

```
                >253
0FA0: A0 1F     >254  :next     ldy   #OFLcol
0FA2: A5 C3     >255           lda   Ov         ; Overflow indicator
0FA4: 20 C1 0F >256           jsr   INDshow
0FA7: A0 22     >257           ldy   #RPTcol
0FA9: A5 C4     >258           lda   Rp         ; Repeat indicator
0FAB: 20 C1 0F >259           jsr   INDshow
0FAE: 60        >260           rts
                >261
0FAF: 3C        >262  :ltstr    asc   '<' ; Inverse
0FB0: A0 BD A0 >263           asc   " = >",00
0FB5: BC A0     >264  :eqstr    asc   "< "
0FB7: 3D        >265           asc   '=' ; Inverse
0FB8: A0 BE 00 >266           asc   " >",00
0FBB: BC A0 BD >267  :gtstr    asc   "< = "
0FBF: 3E 00     >268           asc   '>',00 ; inverse
                >269
                >270  ************************************************************
                >271  *                                                          *
                >272  *     Flip indicator to on (inverse) or off (normal)       *
                >273  *                                                          *
                >274  * A = indicator: 0 is OFF, >0 is ON                        *
                >275  * Y = leftmost column of indicator - 1                     *
                >276  * Exits with Y pointing 1 past last column of indicator    *
                >277  *                                                          *
                >278  ************************************************************
                >279
0FC1: 18        >280  INDshow  clc              ; >0 ==> inv, 0 ==> norm
0FC2: 69 FF     >281           adc   #$FF       ; Set C if >0, reset if 0
0FC4: B1 CC     >282  :loop    lda   (ptr),y    ; Get indicator char
0FC6: 29 20     >283           and   #$20       ; Is it Upper Case?
0FC8: D0 06     >284           bne   :notuc     ; -No, leave it alone.
0FCA: B1 CC     >285           lda   (ptr),y    ; -Yes, turn off $40 bit
0FCC: 29 BF     >286           and   #$BF       ;   to avoid mousetext.
0FCE: D0 02     >287           bne   :switch    ; (always)
                >288
0FD0: B1 CC     >289  :notuc   lda   (ptr),y    ; Recover original char
0FD2: 90 04     >290  :switch  bcc   :norm      ; Set to normal
0FD4: 29 7F     >291           and   #$7F       ; Set to inverse
0FD6: B0 02     >292           bcs   :store     ; (always)
                >293
0FD8: 09 80     >294  :norm    ora   #$80       ; Set to normal
0FDA: 91 CC     >295  :store   sta   (ptr),y
0FDC: C8        >296           iny              ; Advance to next char
0FDD: B1 CC     >297           lda   (ptr),y    ;  and examine it.
0FDF: 09 80     >298           ora   #$80       ; Force normal
0FE1: 49 A0     >299           eor   #"         " ; Space?
0FE3: F0 04     >300           beq   :done      ; -Yes, done.
0FE5: 29 E0     >301           and   #$E0       ; -No, digit?
0FE7: D0 DB     >302           bne   :loop      ; -No, keep going.
0FE9: 60        >303  :done    rts              ; -Yes, done.
```

```
                >305  ************************************************************
                >306  *                                                          *
                >307  *          Display a B220 register on front panel          *
                >308  *                                                          *
                >309  * Address of register attributes block is loaded in A,Y    *
                >310  *                                                          *
                >311  ************************************************************
                >312
0FEA: 85 CC     >313  dispreg  sta   ptr        ; Set register attribute ptr
0FEC: 84 CD     >314           sty   ptr+1
0FEE: A0 00     >315           ldy   #0
0FF0: B1 CC     >316  :cpyattr lda   (ptr),y    ; Copy reg attributes to page 0
0FF2: 99 D3 00  >317           sta   linev,y
0FF5: C8        >318           iny
0FF6: C0 0A     >319           cpy   #10
0FF8: 90 F6     >320           bcc   :cpyattr
0FFA: B1 CC     >321           lda   (ptr),y    ; Addr of low byte of register
0FFC: 8D 0F 10  >322           sta   :reg+1
0FFF: C8        >323           iny
1000: B1 CC     >324           lda   (ptr),y
1002: A8        >325           tay              ; Set Y = rightmost column
1003: 18        >326           clc
1004: A5 CC     >327           lda   ptr        ; Advance ptr to digit mask
1006: 69 0C     >328           adc   #12
1008: 85 CC     >329           sta   ptr
100A: 90 02     >330           bcc   :reg
100C: E6 CD     >331           inc   ptr+1
100E: A5 00     >332  :reg     lda   0*0        ; Load register byte
1010: CE 0F 10  >333           dec   :reg+1     ;  and move to next highest.
1013: 85 D0     >334           sta   t1         ; Save current reg byte
1015: 20 28 10  >335           jsr   dispdig    ; Display lo digit of reg byte
1018: 88        >336           dey              ; Move left one column.
1019: 30 0C     >337           bmi   :done      ; Quit if done...
101B: 20 28 10  >338           jsr   dispdig    ; Display hi digit of reg byte
101E: 88        >339  :skip    dey              ; Move left.
101F: 30 06     >340           bmi   :done      ; -Display complete.
1021: B1 CC     >341           lda   (ptr),y    ; Check mask
1023: F0 F9     >342           beq   :skip      ; -Skip this screen column
1025: D0 E7     >343           bne   :reg       ; -Keep going...
                >344
1027: 60        >345  :done    rts
                >346
```

```
                >348  ************************************************************
                >349  *                                                          *
                >350  *           Display one digit of B220 register             *
                >351  *                                                          *
                >352  ************************************************************
                >353
1028: A5 D0     >354  dispdig  lda   t1         ; Get (shifted) reg byte.
102A: 29 0F     >355           and   #$0F       ; Mask low digit,
102C: 09 B0     >356           ora   #$B0       ;  make ASCII digit,
102E: 91 D3     >357           sta   (linev),y  ;   and store it on screen.
1030: 46 D0     >358           lsr   t1         ; 1-bit to Carry
1032: A9 A0     >359           lda   #off
1034: 90 02     >360           bcc   :st1
1036: A9 AA     >361           lda   #on
1038: 91 D5     >362  :st1     sta   (line1),y  ; Store 1-bit state to screen
103A: 46 D0     >363           lsr   t1         ; 2-bit to Carry
103C: A9 A0     >364           lda   #off
103E: 90 02     >365           bcc   :st2
1040: A9 AA     >366           lda   #on
1042: 91 D7     >367  :st2     sta   (line2),y  ; Store 2-bit state to screen
1044: 46 D0     >368           lsr   t1         ; 4-bit to Carry
1046: A9 A0     >369           lda   #off
1048: 90 02     >370           bcc   :st4
104A: A9 AA     >371           lda   #on
104C: 91 D9     >372  :st4     sta   (line4),y  ; Store 4-bit state to screen
104E: 46 D0     >373           lsr   t1         ; 8-bit to Carry
1050: A9 A0     >374           lda   #off
1052: 90 02     >375           bcc   :st8
1054: A9 AA     >376           lda   #on
1056: 91 DB     >377  :st8     sta   (line8),y  ; Store 8-bit state to screen
1058: 60        >378           rts
```

```
                185             put   B220EXEC1
                >1      * OPcode execute phase dispatch table
                >2
                >3      optabl  equ   *         ; Low byte of execute routines
1059: 0D        >4              db    <HLT      ; S ---- 00 ---- HaLT
105A: 0D        >5              db    <NOP      ; S ---- 01 ---- No OP
105B: 34        >6              db    <OPerr    ;        02
105C: 10        >7              db    <PRD      ; S unnv 03 ADDR Pap tape RD
105D: 16        >8              db    <PRB      ; S u--v 04 ADDR Pap tape Rd, Br
105E: AD        >9              db    <PRI      ; S unnv 05 ADDR Pap tape Rd, Inv
105F: B0        >10             db    <PWR      ; S unn- 06 ADDR Pap tape WR
1060: DD        >11             db    <PWI      ; S u--- 07 ADDR Pap tape Wr, Int
1061: DD        >12             db    <KAD      ; S ---- 08 ---- Keyboard ADd
1062: E0        >13             db    <SPO      ; S dnnv 09 ADDR Sup Print Out
1063: 34 34 34  >14             db    <OPerr,<OPerr,<OPerr,<OPerr,<OPerr,<OPerr
1069: 64        >15             db    <CAD      ; S ---v 10 ADDR Clear ADd (Abs)
106A: 7B        >16             db    <CSU      ; S ---v 11 ADDR Clear SUb (Abs)
106B: 83        >17             db    <ADD      ; S ---v 12 ADDR ADD (Abs)
106C: 0F        >18             db    <SUB      ; S ---v 13 ADDR SUBtract (Abs)
106D: 25        >19             db    <MUL      ; S ---- 14 ADDR MULtiply
106E: A6        >20             db    <DIV      ; S ---- 15 ADDR DIVide
106F: 23        >21             db    <RND      ; S ---- 16 ---- RouND
1070: 45        >22             db    <EXT      ; S ---- 17 ADDR EXTract
1071: 6D        >23             db    <CFA      ; S sLfv 18 ADDR Comp Fld A (R)
1072: F3        >24             db    <ADL      ; S ---- 19 ADDR ADd to Location
1073: 34 34 34  >25             db    <OPerr,<OPerr,<OPerr,<OPerr,<OPerr,<OPerr
1079: 9C        >26             db    <IBB      ; S nnnn 20 ADDR Increase B, Br
107A: AF        >27             db    <DBB      ; S nnnn 21 ADDR Decrease B, Br
107B: 42        >28             db    <FAD      ; S n--v 22 ADDR Float ADd (Abs)
107C: 2D        >29             db    <FSU      ; S n--v 23 ADDR Float SUb (Abs)
107D: 31        >30             db    <FMU      ; S ---- 24 ADDR Float MUltiply
107E: C3        >31             db    <FDV      ; S ---- 25 ADDR Float DiVide
107F: 34        >32             db    <IFL      ; S sLnn 26 ADDR Inc Fld Loc
1080: 7A        >33             db    <DFL      ; S sLnn 27 ADDR Dec Fld Loc
1081: 8A        >34             db    <DLB      ; S sLnn 28 ADDR Dec fld loc,Ld B
1082: 36        >35             db    <RTF      ; S -nn- 29 ADDR Record TransFer
1083: 34 34 34  >36             db    <OPerr,<OPerr,<OPerr,<OPerr,<OPerr,<OPerr
1089: 05        >37             db    <BUN      ; S ---- 30 ADDR Branch UNcond
108A: C2        >38             db    <BOF      ; S ---- 31 ADDR Branch OverFlow
108B: CF        >39             db    <BRP      ; S ---- 32 ADDR Branch RePeat
108C: D5        >40             db    <BSA      ; S ---n 33 ADDR Branch Sign A
108D: DF        >41             db    <BCH      ; S ---v 34 ADDR Br Comp Hi (Lo)
108E: F3        >42             db    <BCE      ; S ---v 35 ADDR Br Comp Eq (Un)
108F: 1C        >43             db    <BFA      ; S sLnn 36 ADDR Branch Field A
1090: 18        >44             db    <BFR      ; S sLnn 37 ADDR Branch Field R
1091: 6B        >45             db    <BCS      ; S u--- 38 ADDR Br Control Sw
1092: 78        >46             db    <SOR      ; S ---V 39 ---- Set Ov Remember
1093: 34 34 34  >47             db    <OPerr,<OPerr,<OPerr,<OPerr,<OPerr,<OPerr
1099: 8C        >48             db    <STA      ; S sLfv 40 ADDR STore A (R/B)
109A: F3        >49             db    <LDR      ; S ---- 41 ADDR LoaD R
109B: FF        >50             db    <LDB      ; S ---v 42 ADDR LoaD B (Comp)
109C: 25        >51             db    <LSA      ; S ---n 43 ---- Load Sign A
109D: 2E        >52             db    <STP      ; S ---- 44 ADDR STore P
109E: 43        >53             db    <CLA      ; S ---v 45 ---- CLr A/R/AR/B/AB/T
109F: 64        >54             db    <CLL      ; S ---- 46 ADDR CLear Location
10A0: 34        >55             db    <OPerr    ;        47
10A1: 6F        >56             db    <SRA      ; S ---v 48 --nn Shft Rt A (AR/AS)
10A2: D7        >57             db    <SLA      ; S ---v 49 --nn Shft Lt A (AR/AS)
10A3: 34 34 34  >58             db    <OPerr,<OPerr,<OPerr,<OPerr,<OPerr,<OPerr
10A9: 45        >59             db    <MTS      ; S uhhv 50 addr Mag Tape Search
10AA: 77        >60             db    <MTC      ; S uhhK 51 addr Mag Tape sCan
10AB: 7A        >61             db    <MRD      ; S un-v 52 addr Mag tape ReaD
10AC: 85        >62             db    <MRR      ; S un-v 53 addr Mt Read Record
10AD: 88        >63             db    <MIW      ; S unkk 54 addr Mt Init Write
```

```
10AE: 90       >64           db    <MIR       ; S un-- 55 addr Mt Init wr Rec
10AF: 93       >65           db    <MOW       ; S unkk 56 addr Mt OverWrite
10B0: 9D       >66           db    <MOR       ; S un-- 57 addr Mt Overwr Rec
10B1: A0       >67           db    <MPF       ; S un-v 58 ---- Mt Pos Fwd
10B2: EA       >68           db    <MIB       ; S u--v 59 addr Mt Interr Branch
```

```
                >70   noAD     equ   $8000      ; Hi bit means "ignore ADDR"
                >71   operr    equ   OPerr+noAD ; Ignore ADDR on illegal OPs.
                >72
                >73   optabh   equ   *          ; High byte of execute routines
10B3: 91        >74            db    >HLT+noAD  ; S ---- 00 ---- HaLT
10B4: 91        >75            db    >NOP+noAD  ; S ---- 01 ---- No OP
10B5: 8C        >76            db    >operr     ;        02
10B6: 11        >77            db    >PRD       ; S unnv 03 ADDR Pap tape RD
10B7: 11        >78            db    >PRB       ; S u--v 04 ADDR Pap tape Rd, Br
10B8: 11        >79            db    >PRI       ; S unnv 05 ADDR Pap tape Rd, Inv
10B9: 11        >80            db    >PWR       ; S unn- 06 ADDR Pap tape WR
10BA: 12        >81            db    >PWI       ; S u--- 07 ADDR Pap tape Wr, Int
10BB: 92        >82            db    >KAD+noAD  ; S ---- 08 ---- Keyboard ADd
10BC: 12        >83            db    >SPO       ; S dnnv 09 ADDR Sup Print Out
10BD: 8C 8C 8C >84             db    >operr,>operr,>operr,>operr,>operr,>operr
10C3: 13        >85            db    >CAD       ; S ---v 10 ADDR Clear ADd (Abs)
10C4: 13        >86            db    >CSU       ; S ---v 11 ADDR Clear SUbtr (Abs)
10C5: 13        >87            db    >ADD       ; S ---v 12 ADDR ADD (Abs)
10C6: 14        >88            db    >SUB       ; S ---v 13 ADDR SUBtract (Abs)
10C7: 14        >89            db    >MUL       ; S ---- 14 ADDR MULtiply
10C8: 14        >90            db    >DIV       ; S ---- 15 ADDR DIVide
10C9: 95        >91            db    >RND+noAD  ; S ---- 16 ---- RouND
10CA: 15        >92            db    >EXT       ; S ---- 17 ADDR EXTract
10CB: 15        >93            db    >CFA       ; S sLfv 18 ADDR Comp Fld A (R)
10CC: 13        >94            db    >ADL       ; S ---- 19 ADDR ADd to Location
10CD: 8C 8C 8C >95             db    >operr,>operr,>operr,>operr,>operr,>operr
10D3: 19        >96            db    >IBB       ; S nnnn 20 ADDR Increase B, Br
10D4: 19        >97            db    >DBB       ; S nnnn 21 ADDR Decrease B, Br
10D5: 16        >98            db    >FAD       ; S n--v 22 ADDR Float ADd (Abs)
10D6: 16        >99            db    >FSU       ; S n--v 23 ADDR Float SUb (Abs)
10D7: 17        >100           db    >FMU       ; S ---- 24 ADDR Float MUltiply
10D8: 17        >101           db    >FDV       ; S ---- 25 ADDR Float DiVide
10D9: 18        >102           db    >IFL       ; S sLnn 26 ADDR Inc Fld Loc
10DA: 18        >103           db    >DFL       ; S sLnn 27 ADDR Dec Fld Loc
10DB: 18        >104           db    >DLB       ; S sLnn 28 ADDR Dec fld loc,Ld B
10DC: 19        >105           db    >RTF       ; S -nn- 29 ADDR Record TransFer
10DD: 8C 8C 8C >106            db    >operr,>operr,>operr,>operr,>operr,>operr
10E3: 1A        >107           db    >BUN       ; S ---- 30 ADDR Branch UNcond
10E4: 19        >108           db    >BOF       ; S ---- 31 ADDR Branch OverFlow
10E5: 19        >109           db    >BRP       ; S ---- 32 ADDR Branch RePeat
10E6: 19        >110           db    >BSA       ; S ---n 33 ADDR Branch Sign A
10E7: 19        >111           db    >BCH       ; S ---v 34 ADDR Br Comp Hi (Lo)
10E8: 19        >112           db    >BCE       ; S ---v 35 ADDR Br Comp Eq (Un)
10E9: 1A        >113           db    >BFA       ; S sLnn 36 ADDR Branch Field A
10EA: 1A        >114           db    >BFR       ; S sLnn 37 ADDR Branch Field R
10EB: 1A        >115           db    >BCS       ; S u--- 38 ADDR Br Control Sw
10EC: 1A        >116           db    >SOR       ; S ---v 39 ---- Set Ov Remember
10ED: 8C 8C 8C >117            db    >operr,>operr,>operr,>operr,>operr,>operr
10F3: 1A        >118           db    >STA       ; S sLfv 40 ADDR STore A (R/B)
10F4: 1A        >119           db    >LDR       ; S ---- 41 ADDR LoaD R
10F5: 1A        >120           db    >LDB       ; S ---v 42 ADDR LoaD B (Comp)
10F6: 9B        >121           db    >LSA+noAD  ; S ---n 43 ---- Load Sign A
10F7: 1B        >122           db    >STP       ; S ---- 44 ADDR STore P
10F8: 9B        >123           db    >CLA+noAD  ; S ---v 45 ---- CLr A/R/AR/B/AB/T
10F9: 1B        >124           db    >CLL       ; S ---- 46 ADDR CLear Location
10FA: 8C        >125           db    >operr     ;        47
10FB: 9B        >126           db    >SRA+noAD  ; S ---v 48 --nn Shft Rt A (AR/AS)
10FC: 9B        >127           db    >SLA+noAD  ; S ---v 49 --nn Shft Lt A (AR/AS)
10FD: 8C 8C 8C >128            db    >operr,>operr,>operr,>operr,>operr,>operr
1103: 1C        >129           db    >MTS       ; S uhhv 50 addr Mag Tape Search
1104: 1C        >130           db    >MTC       ; S uhhk 51 addr Mag Tape sCan
1105: 1C        >131           db    >MRD       ; S un-v 52 addr Mag tape ReaD
1106: 1C        >132           db    >MRR       ; S un-v 53 addr Mt Read Record
```

```
1107: 1C       >133          db    >MIW       ; S unkk 54 addr Mt Init Write
1108: 1C       >134          db    >MIR       ; S un-- 55 addr Mt Init wr Rec
1109: 1C       >135          db    >MOW       ; S unkk 56 addr Mt OverWrite
110A: 1C       >136          db    >MOR       ; S un-- 57 addr Mt Overwr Rec
110B: 9C       >137          db    >MPF+noAD  ; S un-v 58 ---- Mt Pos Fwd
110C: 1C       >138          db    >MIB       ; S u--v 59 addr Mt Interr Branch
```

```
                        >140  ************************************************************
                        >141  *                                                          *
                        >142  *            B220 Instruction Execute Routines             *
                        >143  *                                                          *
                        >144  * For all OPs with ADDR = memory address, Y = 0            *
                        >145  *  and A and rD+S = sign of MEM operand.                   *
                        >146  *                                                          *
                        >147  ************************************************************
                        >148
                        >149  HLT      equ   *          ; Halt is executed in 'fetch'.
                        >150
110D: 4C 72 0B          >151  NOP      jmp   fetch      ; Do nothing.
                        >152
1110: 20 2A 11          >153  PRD      jsr   ]prd       ; Paper tape ReaD
1113: 4C 72 0B          >154          jmp   fetch
                        >155
1116: A5 99             >156  PRB      lda   rC+sL      ; Paper tape Read & Branch
1118: 29 F0             >157          and   #$F0       ; Fake NN = 00 (100 words)
111A: 85 99             >158          sta   rC+sL
111C: A5 9A             >159          lda   rC+VV
111E: 29 0F             >160          and   #$0F
1120: 09 01             >161          ora   #$01       ;  and xeq sign 6/7.
1122: 85 9A             >162          sta   rC+VV
1124: 20 2A 11          >163  :read    jsr   ]prd       ; Read "tape" until
1127: 4C 24 11          >164          jmp   :read      ;  sign 6/7 xeq.
                        >165
112A: 20 BB 11          >166  ]prd     jsr   ptread     ; Read disk into MEM
112D: A5 9A             >167          lda   rC+VV      ; Examine variant digit
112F: 29 08             >168          and   #$08       ; 8-bit on?
1131: 85 D3             >169          sta   linev      ; Set B-mod mask.
1133: A5 9A             >170          lda   rC+VV      ; Variant again...
1135: A0 00             >171          ldy   #0
1137: 29 01             >172          and   #$01       ; Execute 6/7 sign?
1139: F0 02             >173          beq   :noxeq     ; -No, ignore 6/7 sign.
113B: A0 06             >174          ldy   #6         ; -Yes, set xeq mask.
113D: 84 D4             >175  :noxeq   sty   linev+1
113F: A6 D0             >176  :scanlp  ldx   t1         ; Index to unit offset.
1141: 18                >177          clc              ; Advance unit offset.
1142: BD DF 1D          >178          lda   rdroff+2,x ; Lo byte
1145: 69 06             >179          adc   #6
1147: 9D DF 1D          >180          sta   rdroff+2,x
114A: 90 08             >181          bcc   :scan      ; No carry.
114C: FE DE 1D          >182          inc   rdroff+1,x ; Carry into mid byte.
114F: D0 03             >183          bne   :scan      ; No carry.
1151: FE DD 1D          >184          inc   rdroff,x   ; Carry into hi byte.
1154: A0 00             >185  :scan    ldy   #0         ; Scan sign digits
1156: B1 CA             >186          lda   (memptr),y ;  for 8/9 or 6/7.
1158: 25 D3             >187          and   linev      ; Variant 8-bit
115A: F0 0C             >188          beq   :noBmod    ; No B modification
115C: B1 CA             >189          lda   (memptr),y ; B modify ADDR.
115E: 29 01             >190          and   #$01       ; Turn off 8-bit
1160: 91 CA             >191          sta   (memptr),y
1162: 20 8B 11          >192          jsr   Bmodmem    ; B-modify address.
1165: 4C 70 11          >193          jmp   :cont
                        >194
1168: B1 CA             >195  :noBmod  lda   (memptr),y ; Re-fetch sign digit
116A: 25 D4             >196          and   linev+1    ; Apply xeq mask (0/6)
116C: C9 06             >197          cmp   #6         ; Sign = 6 or 7?
116E: F0 08             >198          beq   :xeq       ; -Yes, execute it.
1170: 20 A1 11          >199  :cont    jsr   incmem     ; Advance memptr.
1173: C6 D1             >200          dec   NN         ; More words?
1175: D0 C8             >201          bne   :scanlp    ; -Yes, continue scan.
1177: 60                >202          rts              ; -No, return.
                        >203
1178: A2 00             >204  :xeq     ldx   #0         ; Execute input word.
117A: B1 CA             >205  :xeqlp   lda   (memptr),y
117C: 95 98             >206          sta   rC,x
```

```
117E: C8       >207           iny
117F: E8       >208           inx
1180: E0 06    >209           cpx   #6
1182: D0 F6    >210           bne   :xeqlp
1184: 86 C6    >211           stx   skipincP   ; Don't inc P reg.
1186: 68       >212           pla              ; No return.
1187: 68       >213           pla
1188: 4C 93 0B >214           jmp   execute    ; Execute instruction.
               >215
118B: C8       >216  Bmodmem  iny              ; Advance to
118C: C8       >217           iny              ;  ADDR field.
118D: C8       >218           iny
118E: C8       >219           iny
118F: C8       >220           iny
1190: F8       >221           sed              ; / Decimal mode.
1191: 18       >222           clc
1192: B1 CA    >223           lda   (memptr),y
1194: 65 95    >224           adc   rB+1
1196: 91 CA    >225           sta   (memptr),y
1198: 88       >226           dey
1199: B1 CA    >227           lda   (memptr),y
119B: 65 94    >228           adc   rB
119D: 91 CA    >229           sta   (memptr),y
119F: D8       >230           cld              ; \ Binary mode.
11A0: 60       >231           rts
               >232
11A1: 18       >233  incmem   clc              ; Advance memptr
11A2: A5 CA    >234           lda   memptr     ;  to next word.
11A4: 69 06    >235           adc   #6
11A6: 85 CA    >236           sta   memptr
11A8: 90 02    >237           bcc   :nocarry
11AA: E6 CB    >238           inc   memptr+1   ; Propagate carry.
11AC: 60       >239  :nocarry rts
               >240
11AD: 4C 34 0C >241  PRI      jmp   OPerr      ; Unimplemented
```

```
                >243  zeroff   equ   line1      ; Zero offset flag
                >244  cmdfnx   equ   line2+1    ; File name index
                >245
11B0: 84 D5     >246  PWR      sty   zeroff     ; New file if offset=0.
11B2: 20 C3 11  >247           jsr   ptwrite    ; Paper tape WRite
11B5: 20 C4 12  >248           jsr   incoff     ; Increment unit offset
11B8: 4C 72 0B  >249           jmp   fetch
                >250
11BB: A9 00     >251  ptread   lda   #0         ; PTRDR device class
11BD: 20 D0 11  >252           jsr   setread    ; Start read command
11C0: 4C C8 11  >253           jmp   ptrdwrt    ;  and do the I/O.
                >254
11C3: A9 02     >255  ptwrite  lda   #2         ; PTPCH device class
11C5: 20 DF 11  >256           jsr   setwrite   ; Start write command.
11C8: 20 B5 1D  >257  ptrdwrt  jsr   midNN      ; Get word count
11CB: 85 D1     >258           sta   NN         ;  in binary.
11CD: 4C F2 11  >259           jmp   doio       ; Do the I/O.
                >260
11D0: 85 D0     >261  setread  sta   t1         ; Set device class (0/2/4)
11D2: A0 03     >262           ldy   #3         ; Set I/O cmd to read file.
11D4: B9 4A 12  >263  :loadlp  lda   load,y
11D7: 99 53 12  >264           sta   Bxxxx+1,y
11DA: 88        >265           dey
11DB: 10 F7     >266           bpl   :loadlp
11DD: 30 0D     >267           bmi   getfnx     ; (always)
                >268
11DF: 85 D0     >269  setwrite sta   t1         ; Set device class (0/2/4)
11E1: A0 03     >270           ldy   #3         ; Set I/O cmd to write file.
11E3: B9 4E 12  >271  :savelp  lda   save,y
11E6: 99 53 12  >272           sta   Bxxxx+1,y
11E9: 88        >273           dey
11EA: 10 F7     >274           bpl   :savelp
11EC: 20 69 12  >275  getfnx   jsr   getfnxt1   ; Y = fnx, t1 ==> offset
11EF: 84 D8     >276           sty   cmdfnx     ; Save fnx.
11F1: 60        >277           rts
                >278
11F2: A2 00     >279  doio     ldx   #0         ; New ProDOS command.
11F4: A9 52     >280           lda   #<Bxxxx    ; Start with command.
11F6: A0 12     >281           ldy   #>Bxxxx
11F8: 20 53 20  >282           jsr   putpdcmd
11FB: A4 D8     >283           ldy   cmdfnx     ; Y = file name index.
11FD: B9 F9 1D  >284           lda   fnxfn,y
1200: A0 03     >285           ldy   #>fnames
1202: 20 53 20  >286           jsr   putpdcmd   ; Add file name.
1205: A9 59     >287           lda   #<Aparm
1207: A0 12     >288           ldy   #>Aparm
1209: 20 53 20  >289           jsr   putpdcmd   ; Add ",A$".
120C: A5 CB     >290           lda   memptr+1
120E: A4 CA     >291           ldy   memptr
1210: 20 90 12  >292           jsr   puthx      ; Add hex address...
1213: A9 5D     >293           lda   #<Lparm
1215: A0 12     >294           ldy   #>Lparm
1217: 20 53 20  >295           jsr   putpdcmd   ; Add ",L$"
121A: A5 D1     >296           lda   NN         ; Binary word count
121C: 0A        >297           asl             ; NN * 2
121D: 65 D1     >298           adc   NN         ; NN * 3
121F: 85 CE     >299           sta   inptr
1221: A9 00     >300           lda   #0
1223: 69 00     >301           adc   #0         ; Hi byte of NN * 3
1225: 26 CE     >302           rol   inptr      ; Lo byte of NN * 6
1227: 2A        >303           rol
1228: 85 CF     >304           sta   inptr+1    ; Hi byte of NN * 6
122A: A4 CE     >305           ldy   inptr
122C: 20 90 12  >306           jsr   puthx      ; Add hex length "xxxx"
122F: 86 D6     >307           stx   savex      ; Save X before "B" param
1231: A9 61     >308           lda   #<Bparm
1233: A0 12     >309           ldy   #>Bparm
```

```
1235: 20 53 20 >310           jsr   putpdcmd   ; Add ",B$"
1238: 20 AC 12 >311           jsr   putoff     ; Add hex offset "xxxxxx"
123B: A5 D5    >312           lda   zeroff     ; Create file on write?
123D: D0 02    >313           bne   :useB      ; -No, use B$offset.
123F: A6 D6    >314           ldx   savex      ; -Yes, no B param.
1241: 20 6A 20 >315  :useB    jsr   pdosxeq    ; Execute ProDOS command.
1244: 90 03    >316           bcc   :ok        ; No error.
1246: 4C 44 0C >317           jmp   IOerr      ; I/O error.
1249: 60       >318  :ok      rts
               >319
124A: CC CF C1 >320  load     asc   "LOAD"
124E: D3 C1 D6 >321  save     asc   "SAVE"
1252: C2 F8 F8 >322  Bxxxx    asc   "Bxxxx ",00
1259: AC C1 A4 >323  Aparm    asc   ",A$",00
125D: AC CC A4 >324  Lparm    asc   ",L$",00
1261: AC C2 A4 >325  Bparm    asc   ",B$",00
               >326
               >327  * Get file name index (fnx) and offset displacement (t1)
               >328  *  Entry: t1 = fnx base (0:RDR, 2:PCH, 4:MTape)
               >329  *  Exit: A, t1 = Displacement to unit offset (0..15)
               >330  *        Y = file name index (0..7)
               >331  *        X unchanged.
               >332
1265: A9 04    >333  getMTt1  lda   #4         ; Mag tape fnx base
1267: 85 D0    >334           sta   t1
1269: A5 99    >335  getfnxt1 lda   rC+sL      ; Get unit #
126B: 29 E0    >336           and   #$E0       ; Unit = 0 or 1?
126D: D0 1E    >337           bne   :ioerr     ; -No, I/O error.
126F: A5 99    >338           lda   rC+sL      ; -Yes, isolate
1271: 29 10    >339           and   #$10       ;   unit #.
1273: F0 02    >340           beq   :zero      ; Unit 0.
1275: A9 01    >341           lda   #1         ; Unit 1.
1277: 18       >342  :zero    clc              ; Add fnx base: 0 (PTRDR),
1278: 65 D0    >343           adc   t1         ;  2 (PTPCH), 4 (MT unit).
127A: A8       >344           tay
127B: C9 04    >345           cmp   #4         ; Mag tape? (4 or 5)
127D: 90 08    >346           bcc   :fnx       ; -No, A = file name index.
127F: C9 05    >347           cmp   #5         ; -Yes, if MT unit = 1,
1281: 69 00    >348           adc   #0         ;   add 1.
1283: 79 EB 1D >349           adc   mtlane-4,y ; Add lane 0/1.
1286: A8       >350           tay
1287: B9 F1 1D >351  :fnx     lda   fnxoff,y   ; Disp to unit offset
128A: 85 D0    >352           sta   t1         ;  in t1.
128C: 60       >353           rts
               >354
128D: 4C 44 0C >355  :ioerr   jmp   IOerr      ; I/O error relay.
               >356
1290: 20 94 12 >357  puthx    jsr   putbyte    ; Put first byte in hex
1293: 98       >358           tya              ;  and fall into putbyte.
1294: 48       >359  putbyte  pha              ; Save byte
1295: 4A       >360           lsr
1296: 4A       >361           lsr
1297: 4A       >362           lsr
1298: 4A       >363           lsr
1299: 20 9D 12 >364           jsr   :stdig     ; Put hi hex digit
129C: 68       >365           pla              ;  and then lo dig.
129D: 29 0F    >366  :stdig   and   #$0F       ; Isolate digit
129F: 09 B0    >367           ora   #"0"       ; Or in zone
12A1: C9 BA    >368           cmp   #$BA       ; >9?
12A3: 90 02    >369           bcc   :store     ; -No, store it.
12A5: 69 06    >370           adc   #6         ; -Yes, cvt to A..F
12A7: 9D 00 02 >371  :store   sta   IN,x       ; Add char to IN buffer.
12AA: E8       >372           inx
12AB: 60       >373           rts
               >374
12AC: A4 D0    >375  putoff   ldy   t1         ; Index to unit offset.
12AE: A9 03    >376           lda   #3         ; 3-byte binary offset
```

```
12B0: 85 CC    >377           sta   ptr        ; 3-byte offset.
12B2: B9 DD 1D >378  :offlp   lda   rdroff,y
12B5: 48       >379           pha
12B6: 05 D5    >380           ora   zeroff     ; Update zero
12B8: 85 D5    >381           sta   zeroff     ;  offset flag.
12BA: 68       >382           pla
12BB: 20 94 12 >383           jsr   putbyte
12BE: C8       >384           iny              ; Inc byte index
12BF: C6 CC    >385           dec   ptr        ; More bytes?
12C1: D0 EF    >386           bne   :offlp     ; -Yes, go again.
12C3: 60       >387           rts
               >388
12C4: A6 D0    >389  incoff   ldx   t1         ; Unit offset index.
12C6: 18       >390           clc
12C7: BD DF 1D >391           lda   rdroff+2,x ; Lo byte
12CA: 65 CE    >392           adc   inptr      ; Add length * 6
12CC: 9D DF 1D >393           sta   rdroff+2,x
12CF: BD DE 1D >394           lda   rdroff+1,x ; Mid byte
12D2: 65 CF    >395           adc   inptr+1
12D4: 9D DE 1D >396           sta   rdroff+1,x
12D7: 90 03    >397           bcc   :rts       ; Carry out?
12D9: FE DD 1D >398           inc   rdroff,x   ; -Yes, inc hi byte.
12DC: 60       >399  :rts     rts              ; -No, return.
               >400
               >401  PWI
               >402  KAD
12DD: 4C 34 0C >403           jmp   OPerr      ; Unimplemented
```

```
12E0: 20 B5 1D >405  SPO      jsr   midNN      ; Get count (NN) in A
12E3: 85 D1    >406           sta   NN         ; NN = binary word count.
12E5: A0 00    >407  :nxword  ldy   #0
12E7: B1 CA    >408           lda   (memptr),y ; Get sign
12E9: C9 02    >409           cmp   #2         ; Alphanumeric?
12EB: D0 3A    >410           bne   :num       ; -No, numeric.
12ED: C8       >411  :nxchar  iny              ; -Yes, print alpha.
12EE: B1 CA    >412           lda   (memptr),y ; Get next char
12F0: C9 26    >413           cmp   #$26       ; "Tab" code?
12F2: F0 11    >414           beq   :tab       ; -Yes, do tab.
12F4: C9 02    >415           cmp   #$02       ; -No, "Ignore" code?
12F6: F0 07    >416           beq   :ignore    ; -Yes, skip it.
12F8: AA       >417           tax              ; -No, translate B220
12F9: BD 01 1E >418           lda   b220asc,x  ;       char to ASCII.
12FC: 20 ED FD >419           jsr   COUT       ;        and print it.
12FF: C0 05    >420  :ignore  cpy   #5         ; Word complete?
1301: D0 EA    >421           bne   :nxchar    ; -No, keep going.
1303: F0 4E    >422           beq   :done      ; -Yes, word done (always)
               >423
1305: A2 00    >424  :tab     ldx   #0
1307: A5 24    >425           lda   CH
1309: DD 5F 13 >426  :nxtab   cmp   tabs,x     ; Find first tab
130C: 90 07    >427           bcc   :gottab    ;  greater than CH.
130E: E8       >428           inx
130F: E0 05    >429           cpx   #5
1311: D0 F6    >430           bne   :nxtab
1313: F0 EA    >431           beq   :ignore    ; (always) Skip if past tabs.
               >432
1315: 84 D0    >433  :gottab  sty   t1         ; Save Y
1317: BC 5F 13 >434           ldy   tabs,x     ; Get target tab position.
131A: A9 A0    >435  :prtblnk lda   #"         "
131C: 20 ED FD >436           jsr   COUT       ; Print blanks until at
131F: C4 24    >437           cpy   CH         ;  target tab position.
1321: D0 F7    >438           bne   :prtblnk
1323: A4 D0    >439           ldy   t1         ; Restore Y
1325: D0 D8    >440           bne   :ignore    ;  and continue. (always)
               >441
1327: A2 A0    >442  :num     ldx   #"         " ; Print blank if sign 0
1329: C9 00    >443           cmp   #0
132B: F0 09    >444           beq   :prtsign
132D: A2 AD    >445           ldx   #"-"       ; Print - if sign 1
132F: C9 01    >446           cmp   #1
1331: F0 03    >447           beq   :prtsign
1333: 09 B0    >448           ora   #"0"       ; Else print sign digit.
1335: AA       >449           tax
1336: 8A       >450  :prtsign txa
1337: 20 ED FD >451           jsr   COUT
133A: C8       >452  :nxbyte  iny              ; Print rest of number.
133B: B1 CA    >453           lda   (memptr),y
133D: 48       >454           pha
133E: 4A       >455           lsr
133F: 4A       >456           lsr
1340: 4A       >457           lsr
1341: 4A       >458           lsr              ; Hi digit it A
1342: 09 B0    >459           ora   #"0"       ; OR in zone
1344: 20 ED FD >460           jsr   COUT       ;  and print digit.
1347: 68       >461           pla              ; Recover low digit
1348: 29 0F    >462           and   #$0F       ; Isolate it
134A: 09 B0    >463           ora   #"0"       ;  add zone
134C: 20 ED FD >464           jsr   COUT       ;   and print it.
134F: C0 05    >465           cpy   #5         ; End of word?
1351: D0 E7    >466           bne   :nxbyte    ; -No, continue.
1353: C6 D1    >467  :done    dec   NN         ; -Yes, more words?
1355: F0 05    >468           beq   :quit      ; -No, all done.
1357: 20 A1 11 >469           jsr   incmem     ; -Yes, increment memptr.
135A: D0 89    >470           bne   :nxword    ; (always)
               >471
```

```
135C: 4C 72 0B >472  :quit    jmp   fetch
               >473
135F: 09 11 19 >474  tabs     db    9,17,25,33,41 ; SPO tab table
               >475
1364: A5 9A    >476  CAD      lda   rC+VV      ; CAD/CAA
1366: 29 01    >477           and   #$01       ; 0 = CAD, 1 = CAA
1368: 49 0F    >478           eor   #$0F       ; $0F = CAD, $0E = CAA
136A: 25 AA    >479           and   rD+S       ; Unchanged or Abs
136C: 85 9E    >480           sta   rA+S
136E: A0 05    >481           ldy   #5
1370: B1 CA    >482  :cpyloop lda   (memptr),y
1372: 99 9E 00 >483           sta   rA,y
1375: 88       >484           dey
1376: D0 F8    >485           bne   :cpyloop
1378: 4C 72 0B >486           jmp   fetch
               >487
137B: A5 AA    >488  CSU      lda   rD+S       ; CSU, CSA
137D: 49 01    >489           eor   #$01       ; Flip sign and
137F: 85 AA    >490           sta   rD+S       ;  do CAD/CAA.
1381: D0 E1    >491           bne   CAD        ; (always)
```

```
1383: A5 9A    >493  ADD      lda   rC+VV      ; ADD, ADA
1385: 29 0F    >494           and   #$0F
1387: C9 01    >495           cmp   #1         ; ADA?
1389: D0 04    >496           bne   :add       ; -No, ADD.
138B: A9 00    >497           lda   #0         ; -Yes, force MEM sign +
138D: 85 AA    >498           sta   rD+S
138F: 20 95 13 >499  :add     jsr   ]add       ; Do the add.
1392: 4C 72 0B >500           jmp   fetch
               >501
1395: A5 9E    >502  ]add     lda   rA+S
1397: 29 01    >503           and   #$01
1399: 85 9E    >504           sta   rA+S       ; Force sign 0 (+) or 1 (-)
139B: 45 AA    >505           eor   rD+S       ; Signs same or different?
139D: 29 01    >506           and   #$01
139F: D0 18    >507           bne   :subtr     ; -Different, subtract.
13A1: A0 05    >508           ldy   #5         ; -Same, add.
13A3: F8       >509           sed              ; / Decimal mode.
13A4: 18       >510           clc
13A5: B9 9E 00 >511  :addloop lda   rA,y       ; Do the addition...
13A8: 71 CA    >512           adc   (memptr),y
13AA: 99 9E 00 >513           sta   rA,y
13AD: 88       >514           dey
13AE: D0 F5    >515           bne   :addloop
13B0: D8       >516           cld              ; \ Back to binary.
13B1: 90 3F    >517           bcc   :done      ; Done.
               >518           seti  Ov         ; Signal Overflow
13B3: A9 FF    >518           lda   #$FF
13B5: 85 C3    >518           sta   Ov         ; Set non-zero.
               >518           eom
13B7: D0 39    >519           bne   :done      ; (always)
               >520
13B9: A0 01    >521  :subtr   ldy   #1         ; Compare magnitudes.
13BB: B9 9E 00 >522  :comloop lda   rA,y
13BE: D1 CA    >523           cmp   (memptr),y
13C0: F0 04    >524           beq   :cont      ; Equal, keep comparing.
13C2: B0 07    >525           bcs   :Abig      ; rA is bigger
13C4: 90 16    >526           bcc   :Asmall    ; rA is smaller
               >527
13C6: C8       >528  :cont    iny
13C7: C0 06    >529           cpy   #6
13C9: D0 F0    >530           bne   :comloop   ; If =, fall into :Abig.
13CB: A0 05    >531  :Abig    ldy   #5         ; Subtract MEM from rA.
13CD: F8       >532           sed              ; / Decimal mode.
13CE: B9 9E 00 >533  :subloop lda   rA,y
13D1: F1 CA    >534           sbc   (memptr),y
13D3: 99 9E 00 >535           sta   rA,y
13D6: 88       >536           dey
13D7: D0 F5    >537           bne   :subloop
13D9: D8       >538           cld              ; \ Back to binary.
13DA: F0 16    >539           beq   :done      ; (always)
               >540
13DC: A5 AA    >541  :Asmall  lda   rD+S       ; MEM - rA ==> rA
13DE: 29 01    >542           and   #$01       ; rA sign = MEM sign.
13E0: 85 9E    >543           sta   rA+S
13E2: A0 05    >544           ldy   #5
13E4: F8       >545           sed              ; / Decimal mode.
13E5: 38       >546           sec
13E6: B1 CA    >547  :sloop   lda   (memptr),y
13E8: F9 9E 00 >548           sbc   rA,y
13EB: 99 9E 00 >549           sta   rA,y
13EE: 88       >550           dey
13EF: D0 F5    >551           bne   :sloop
13F1: D8       >552           cld              ; \ Back to binary.
13F2: 60       >553  :done    rts
```

```
13F3: A2 FA    >555  ADL      ldx   #-6         ; MEM + rA ==> MEM
13F5: B5 A4    >556  :pushlp  lda   rA+6,x      ; Push rA
13F7: 48       >557           pha
13F8: E8       >558           inx
13F9: D0 FA    >559           bne   :pushlp
13FB: 20 95 13 >560           jsr   ]add        ; rA + MEM ==> rA
13FE: A0 05    >561           ldy   #5          ; rA ==> MEM
1400: B9 9E 00 >562  :mvloop  lda   rA,y
1403: 91 CA    >563           sta   (memptr),y
1405: 68       >564           pla               ;  and pop rA.
1406: 99 9E 00 >565           sta   rA,y
1409: 88       >566           dey
140A: 10 F4    >567           bpl   :mvloop
140C: 4C 72 0B >568           jmp   fetch
               >569
140F: A5 9A    >570  SUB      lda   rC+VV       ; SUB, SUA
1411: 29 0F    >571           and   #$0F
1413: C9 01    >572           cmp   #1          ; SUA?
1415: F0 06    >573           beq   :setsign    ; -Yes, set MEM -
1417: A5 AA    >574  :sub     lda   rD+S        ; -No, SUB.
1419: 29 01    >575           and   #$01        ; Invert MEM sign
141B: 49 01    >576           eor   #$01
141D: 85 AA    >577  :setsign sta   rD+S
141F: 20 95 13 >578           jsr   ]add        ;  and add.
1422: 4C 72 0B >579           jmp   fetch
```

```
1425: 20 2B 14 >581  MUL      jsr   multiply   ; Multiply
1428: 4C 72 0B >582           jmp   fetch
               >583
142B: 45 9E    >584  multiply eor   rA+S       ; Multiply subroutine
142D: 29 01    >585           and   #$01
142F: 48       >586           pha               ; Save result sign
1430: A2 00    >587           ldx   #0
1432: A0 05    >588           ldy   #5
1434: B1 CA    >589  :init    lda   (memptr),y ; rD = multiplicand
1436: 99 AA 00 >590           sta   rD,y
1439: 99 B0 00 >591           sta   rD10,y     ; rD10 = multiplicand
143C: B9 9E 00 >592           lda   rA,y       ; rR = multiplier
143F: 99 A4 00 >593           sta   rR,y
1442: 96 9E    >594           stx   rA,y       ; rA = 0 (including sign)
1444: 88       >595           dey
1445: 10 ED    >596           bpl   :init
1447: 86 AA    >597           stx   rD+S       ; Clear rD sign
1449: 86 B0    >598           stx   rD10+S     ;  and rD10 sign.
144B: A0 04    >599           ldy   #4         ; 4 bits/digit.
144D: 18       >600  :shloop  clc               ; Shift in zeros.
144E: 26 B5    >601           rol   rD10+5     ; Multiply rD10 by 10.
1450: 26 B4    >602           rol   rD10+4
1452: 26 B3    >603           rol   rD10+3
1454: 26 B2    >604           rol   rD10+2
1456: 26 B1    >605           rol   rD10+1
1458: 26 B0    >606           rol   rD10
145A: 88       >607           dey
145B: D0 F0    >608           bne   :shloop
145D: A9 05    >609           lda   #5          ; Set multiplier byte
145F: 85 D0    >610           sta   t1          ;  count = 5.
1461: F8       >611           sed               ; / Decimal mode.
1462: A5 A9    >612  :ckadd1  lda   rR+5
1464: 29 0F    >613           and   #$0F        ; Low digit of multiplier
1466: F0 10    >614           beq   :ckadd10   ; Skip add1 if zero.
1468: A8       >615           tay               ; Y = add1 count.
1469: A2 05    >616  :add1    ldx   #5
146B: 18       >617           clc               ; rA = rA + rD
146C: B5 9E    >618  :add1lp  lda   rA,x
146E: 75 AA    >619           adc   rD,x
1470: 95 9E    >620           sta   rA,x
1472: CA       >621           dex
1473: 10 F7    >622           bpl   :add1lp
1475: 88       >623           dey               ; More adds?
1476: D0 F1    >624           bne   :add1      ; -Yes.
1478: A5 A9    >625  :ckadd10 lda   rR+5       ; Low multiplier byte
147A: 29 F0    >626           and   #$F0        ; High digit of byte
147C: F0 14    >627           beq   :shift     ; Skip add10 if zero.
147E: 4A       >628           lsr
147F: 4A       >629           lsr
1480: 4A       >630           lsr
1481: 4A       >631           lsr
1482: A8       >632           tay               ; Y = add10 count.
1483: A2 05    >633  :add10   ldx   #5
1485: 18       >634           clc               ; rA = rA + rD10
1486: B5 9E    >635  :add10lp lda   rA,x
1488: 75 B0    >636           adc   rD10,x
148A: 95 9E    >637           sta   rA,x
148C: CA       >638           dex
148D: 10 F7    >639           bpl   :add10lp
148F: 88       >640           dey               ; More adds?
1490: D0 F1    >641           bne   :add10     ; -Yes.
1492: 20 5D 1D >642  :shift   jsr   srT2        ; -No, shift |rA| & |rR|
1495: A5 9E    >643           lda   rA+S        ;  right 2 digits
1497: 85 9F    >644           sta   rA+1        ;   including rA sign.
1499: 86 9E    >645           stx   rA+S        ; Clear rA sign.
149B: C6 D0    >646           dec   t1          ; Keep going if more
149D: D0 C3    >647           bne   :ckadd1    ;  multiplier digits.
```

```
149F: D8       >648          cld              ; \ Back to binary.
14A0: 68       >649          pla              ; Recover product sign
14A1: 85 9E    >650          sta   rA+S       ;  and set rA & rR signs.
14A3: 85 A4    >651          sta   rR+S
14A5: 60       >652          rts
```

```
14A6: 20 AC 14 >654  DIV      jsr   divide     ; DIVide
14A9: 4C 72 0B >655           jmp   fetch
               >656
14AC: 45 9E    >657  divide   eor   rA+S
14AE: 29 01    >658           and   #$01
14B0: 48       >659           pha               ; Sign of quotient
14B1: A5 9E    >660           lda   rA+S
14B3: 85 A4    >661           sta   rR+S        ; Sign of remainder
14B5: C8       >662           iny               ; Y = 1: skip signs.
14B6: B9 9E 00 >663  :comp    lda   rA,y        ; Compare rA magnitude
14B9: D1 CA    >664           cmp   (memptr),y ;  with divisor magnitude.
14BB: 90 0F    >665           bcc   :divide     ; rA < MEM, so divide.
14BD: D0 05    >666           bne   :oflow      ; rA > MEM, overflow.
14BF: C8       >667           iny
14C0: C0 06    >668           cpy   #6
14C2: D0 F2    >669           bne   :comp
               >670  :oflow   seti  Ov          ; Signal overflow
14C4: A9 FF    >670           lda   #$FF
14C6: 85 C3    >670           sta   Ov          ; Set non-zero.
               >670           eom
14C8: 68       >671           pla               ; Drop result sign.
14C9: 4C 72 0B >672           jmp   fetch
               >673
14CC: A0 0A    >674  :divide  ldy   #10         ; Quotient digit count = 10.
14CE: 84 D0    >675           sty   t1
14D0: A0 05    >676           ldy   #5
14D2: B1 CA    >677  :div2rD  lda   (memptr),y ; Move divisor to rD
14D4: 99 AA 00 >678           sta   rD,y
14D7: 88       >679           dey
14D8: D0 F8    >680           bne   :div2rD
14DA: 84 9E    >681           sty   rA+S        ; Clear signs of rA
14DC: 84 AA    >682           sty   rD+S        ;  and rD.
14DE: F8       >683           sed               ; / Decimal mode.
14DF: A0 04    >684  :shift   ldy   #4          ; 4 bits/digit.
14E1: 18       >685  :shiftlp clc               ; Shift AR left 1 digit
14E2: 20 71 1D >686           jsr   slT         ;  shifting in zeros.
14E5: 26 9E    >687           rol   rA+S        ;   (include sign in A)
14E7: 88       >688           dey
14E8: D0 F7    >689           bne   :shiftlp
14EA: A2 00    >690           ldx   #0
14EC: B5 9E    >691  :complp  lda   rA,x        ; Compare A with divisor
14EE: D5 AA    >692           cmp   rD,x
14F0: 90 25    >693           bcc   :zero       ; Speed up quotient zeros.
14F2: D0 05    >694           bne   :sub        ; A > divisor
14F4: E8       >695           inx
14F5: E0 06    >696           cpx   #6
14F7: D0 F3    >697           bne   :complp
14F9: A2 05    >698  :sub     ldx   #5          ; A(ext) = A(ext) - D(ext).
14FB: 38       >699           sec
14FC: B5 9E    >700  :sublp   lda   rA,x
14FE: F5 AA    >701           sbc   rD,x
1500: 95 9E    >702           sta   rA,x
1502: CA       >703           dex
1503: 10 F7    >704           bpl   :sublp
1505: 90 04    >705           bcc   :restore    ; Restore if underflow
1507: E6 A9    >706           inc   rR+5        ; Increment quotient digit.
1509: D0 EE    >707           bne   :sub        ; (always)
               >708
150B: A2 05    >709  :restore ldx   #5          ; Add divisor back to A.
150D: 18       >710           clc
150E: B5 9E    >711  :restlp  lda   rA,x
1510: 75 AA    >712           adc   rD,x
1512: 95 9E    >713           sta   rA,x
1514: CA       >714           dex
1515: 10 F7    >715           bpl   :restlp
1517: C6 D0    >716  :zero    dec   t1          ; Quotient complete?
1519: D0 C4    >717           bne   :shift      ; -No, keep dividing.
```

```
151B: 20 86 1D >718          jsr   exchAR     ; -Yes, exchange A and R
151E: D8       >719          cld              ; \ Back to binary.
151F: 68       >720          pla
1520: 85 9E    >721          sta   rA+S       ; Set quotient sign.
1522: 60       >722          rts
```

```
1523: A5 A5    >724  RND      lda   rR+1       ; Hi digit of rR
1525: C9 50    >725           cmp   #$50       ; >= 5?
1527: 90 19    >726           bcc   :done      ; -No, done.
1529: A2 A4    >727           ldx   #rR        ; -Yes, clear rR
152B: 20 A8 1D >728           jsr   clear
152E: F8       >729           sed             ; / Decimal mode.
152F: 38       >730           sec             ; Add 1 to rA.
1530: A2 05    >731           ldx   #5
1532: B5 9E    >732  :rndloop lda   rA,x
1534: 69 00    >733           adc   #0
1536: 95 9E    >734           sta   rA,x
1538: CA       >735           dex
1539: D0 F7    >736           bne   :rndloop
153B: D8       >737           cld             ; \ Back to binary.
153C: 90 04    >738           bcc   :done
               >739           seti  Ov         ; Signal Overflow.
153E: A9 FF    >739           lda   #$FF
1540: 85 C3    >739           sta   Ov         ; Set non-zero.
               >739           eom
1542: 4C 72 0B >740  :done    jmp   fetch
               >741
1545: A0 05    >742  EXT      ldy   #5         ; Extract digits from rA
1547: B1 CA    >743  :extlp   lda   (memptr),y ;  where MEM digits are odd.
1549: 29 11    >744           and   #$11       ; Isolate odd bits
154B: AA       >745           tax             ; $00, $01, $10, $11.
154C: BD 5B 15 >746           lda   :exttbl,x  ; $00, $0F, $F0, $FF.
154F: 39 9E 00 >747           and   rA,y       ; Mask rA digits
1552: 99 9E 00 >748           sta   rA,y
1555: 88       >749           dey
1556: 10 EF    >750           bpl   :extlp
1558: 4C 72 0B >751           jmp   fetch
               >752
155B: 00 0F    >753  :exttbl  db    $00,$0F    ; Indices $00, $01 used
155D: 03 02 01 >754  signtbl  db    3,2,1,0,7,6,5,4,8,9 ; CFx sign order
1567: 00 00 00 >755           db    0,0,0,0    ; (filler)
156B: F0 FF    >756           db    $F0,$FF    ; Indices $10, $11 used.
               >757
156D: A5 9A    >758  CFA      lda   rC+VV      ; CFA, CFR
156F: 29 0F    >759           and   #$0F
1571: A2 A4    >760           ldx   #rR
1573: C9 01    >761           cmp   #1         ; CFR?
1575: F0 02    >762           beq   :cfr       ; -Yes.
1577: A2 9E    >763           ldx   #rA        ; No, CFA.
1579: A5 9A    >764  :cfr     lda   rC+VV      ; Reload variant
157B: 29 10    >765           and   #$10       ; Partial field bit
157D: A8       >766           tay             ;  to Y.
157E: A9 D0    >767           lda   #BNEop     ; Do signed compare.
1580: 20 8F 15 >768           jsr   compare
1583: 85 C2    >769           sta   COMP       ; Set COMPare indicator
1585: A5 C1    >770           lda   ERR        ; Error detected?
1587: D0 03    >771           bne   :err       ; -Yes, report it.
1589: 4C 72 0B >772           jmp   fetch
               >773
158C: 4C 4C 0C >774  :err     jmp   ]err
```

```
                        >776  ************************************************************
                        >777  *                                                          *
                        >778  * Compare register with (memptr), whole or partial field.*
                        >779  *                                                          *
                        >780  * Entry: X = Register addr, (memptr) = comparand addr      *
                        >781  *        Y = Whole (0) or partial (not 0)                  *
                        >782  *        A = BNE (signed comp) or BCS (unsigned comp)      *
                        >783  *                                                          *
                        >784  *  Exit: A = COMP indicator state (<0, 0, >0)            *
                        >785  *                                                          *
                        >786  ************************************************************
                        >787
158F: 8D B9 15          >788  compare  sta   :magonly   ; Signed/unsigned (BNE, BCS)
1592: B5 00             >789           lda   0,x        ; Save register sign
1594: 8D BC 15          >790           sta   :cmpsign+1 ;  for compare.
1597: 8E EB 15          >791           stx   :comp1+1   ; And save register
159A: 8E 16 16          >792           stx   :comp2+1   ;  address for loads.
159D: 8E 21 16          >793           stx   :byte+1
15A0: 84 CC             >794           sty   ptr        ; Save whole/partial.
15A2: C0 00             >795           cpy   #0         ; Whole/partial (0, not 0)
15A4: D0 06             >796           bne   :partial   ; -Yes.
15A6: A9 00             >797           lda   #0         ; -No, fake 0:0 field
15A8: A2 0B             >798           ldx   #11        ;   and compare signs.
15AA: D0 0F             >799           bne   :cmpsign   ; (always)
                        >800
15AC: 20 94 1D          >801  :partial jsr   splitsL    ; Split sL: A = s and X = L.
15AF: 18                >802           clc              ; A = low digit, 1..10
15B0: 69 01             >803           adc   #1         ;     low dig + 1, 2..11
15B2: 38                >804           sec
15B3: 86 D0             >805           stx   t1         ; Digit length
15B5: E5 D0             >806           sbc   t1         ; A = hi digit #
15B7: 90 18             >807           bcc   :flderr    ; <0 ==> Field error.
15B9: D0 1F             >808  :magonly bne   :comp      ; >0 ==> Comp magnitudes.
15BB: A0 00             >809  :cmpsign ldy   #0*0       ; =0 ==> Compare signs.
15BD: C4 AA             >810           cpy   rD+S       ; Reg sign = MEM sign?
15BF: F0 15             >811           beq   :nosign    ; -Yes, comp magnitudes.
15C1: B9 5D 15          >812           lda   signtbl,y  ; -No, translate reg sign
15C4: A4 AA             >813           ldy   rD+S       ; MEM sign
15C6: BE 5D 15          >814           ldx   signtbl,y  ;  translated.
15C9: 86 D0             >815           stx   t1
15CB: C5 D0             >816           cmp   t1         ; Compare signs.
15CD: E6 CC             >817           inc   ptr        ; Force no flip.
15CF: D0 26             >818           bne   :neql      ; (always) Sign determines.
                        >819
15D1: A5 C6             >820  :flderr  lda   "F"        ; Signal Field error.
15D3: 85 C1             >821           sta   ERR
15D5: 60                >822           rts
                        >823
15D6: 18                >824  :nosign  clc              ; Exclude sign from field
15D7: 69 01             >825           adc   #1         ; Field start + 1
15D9: CA                >826           dex              ; Field length - 1
15DA: 18                >827  :comp    clc
15DB: 69 01             >828           adc   #1
15DD: 4A                >829           lsr              ; A = hi byte for compare
15DE: A8                >830           tay              ; Y = hi byte index
15DF: B0 2E             >831           bcs   :lodigit   ; C ==> lo digit of hi byte.
15E1: CA                >832  :hidigit dex              ; Next digit, too?
15E2: D0 3C             >833           bne   :byte      ; -Yes, comp whole byte.
15E4: B1 CA             >834           lda   (memptr),y ; MEM byte
15E6: 29 F0             >835           and   #$F0       ; -No, final digit.
15E8: 85 D0             >836           sta   t1
15EA: B9 00 00          >837  :comp1   lda   0*0,y      ; Reg byte
15ED: 29 F0             >838           and   #$F0       ; Hi digit
15EF: C5 D0             >839  :final   cmp   t1         ; Compare final digit.
15F1: D0 04             >840  :done    bne   :neql      ; =?
15F3: A9 00             >841           lda   #0         ; -Yes, A = 0.
15F5: F0 06             >842           beq   :fin       ; (always)
```

```
                >843
15F7: A9 01     >844  :neql     lda   #1
15F9: B0 02     >845            bcs   :fin       ; >
15FB: A9 FF     >846            lda   #-1        ; <
15FD: A4 CC     >847  :fin      ldy   ptr        ; Recover whole/partial
15FF: D0 0D     >848            bne   :noflip    ; Partial ==> no flip
1601: A6 AA     >849            ldx   rD+S       ; Original sign
1603: F0 09     >850            beq   :noflip    ; + if 0.
1605: E0 04     >851            cpx   #4         ; Collate as + or -?
1607: B0 05     >852            bcs   :noflip    ; + if >= 4.
1609: AA        >853            tax              ; - if 1, 2, or 3.
160A: F0 02     >854            beq   :noflip    ; Comp =, no flip.
160C: 49 80     >855            eor   #$80       ; Exchange > and <.
160E: 60        >856  :noflip   rts
                >857
160F: B1 CA     >858  :lodigit  lda   (memptr),y ; MEM byte
1611: 29 0F     >859            and   #$0F       ; Lo digit
1613: 85 D0     >860            sta   t1         ; Save for compare.
1615: B9 00 00  >861  :comp2    lda   0*0,y      ; Reg byte
1618: 29 0F     >862            and   #$0F       ; Lo digit
161A: C5 D0     >863            cmp   t1         ; Compare digits.
161C: D0 D3     >864            bne   :done      ; Done if unequal.
161E: F0 07     >865            beq   :nxbyte    ; Else continue (always)
                >866
1620: B9 00 00  >867  :byte     lda   0*0,y      ; Reg byte
1623: D1 CA     >868            cmp   (memptr),y ; Compare w MEM.
1625: D0 CA     >869            bne   :done      ; Done if unequal.
1627: C8        >870  :nxbyte   iny              ; Advance byte index and
1628: CA        >871            dex              ;  decrement digit count
1629: D0 B6     >872            bne   :hidigit   ; Continue if digits left,
162B: F0 C4     >873            beq   :done      ;  else done. (always)
```

```
                186           put   B220EXEC2
162D: 29 01    >1    FSU      and   #$01        ; Standardize sign of
162F: 85 AA    >2             sta   rD+S        ;  MEM operand (0/1).
1631: A5 9A    >3             lda   rC+VV       ; FSU or FSA?
1633: 29 0F    >4             and   #$0F
1635: C9 01    >5             cmp   #1
1637: F0 04    >6             beq   :setneg     ; -FSA, set operand -.
1639: A5 AA    >7             lda   rD+S        ; -FSU.
163B: 49 01    >8             eor   #$01        ;   Complement sign
163D: 85 AA    >9    :setneg  sta   rD+S        ;    of operand,
163F: 4C 50 16 >10            jmp   ]fad        ;     and do FAD.
               >11
1642: 29 01    >12   FAD      and   #$01        ; Standardize sign of
1644: 85 AA    >13            sta   rD+S        ;  MEM operand (0/1).
1646: A5 9A    >14            lda   rC+VV       ; FAD or FAA?
1648: 29 0F    >15            and   #$0F
164A: 49 01    >16            eor   #$01
164C: D0 02    >17            bne   ]fad        ; -FAD, continue.
164E: 85 AA    >18            sta   rD+S        ; -FAA, force +.
1650: A5 99    >19   ]fad     lda   rC+sL       ; Get normalization limit.
1652: 4A       >20            lsr
1653: 4A       >21            lsr
1654: 4A       >22            lsr
1655: 4A       >23            lsr
1656: D0 02    >24            bne   :nonzero
1658: A9 0A    >25            lda   #10
165A: 85 D1    >26   :nonzero sta   NN          ; Save binary norm limit.
165C: A5 9E    >27            lda   rA+S        ; Standardize rA sign (0/1)
165E: 29 01    >28            and   #$01
1660: 85 9E    >29            sta   rA+S
1662: A0 05    >30            ldy   #5          ; Copy MEM operand to rD.
1664: B1 CA    >31   :mem2rD  lda   (memptr),y
1666: 99 AA 00 >32            sta   rD,y
1669: 88       >33            dey
166A: D0 F8    >34            bne   :mem2rD     ; (rD sign already set)
166C: 84 D0    >35            sty   t1          ; Init t1 = 0
166E: A2 01    >36            ldx   #EXP        ; Compare rA & rD magnitudes
1670: B5 9E    >37   :complp  lda   rA,x
1672: D5 AA    >38            cmp   rD,x
1674: 90 39    >39            bcc   :Alt        ; rA < rD.
1676: D0 05    >40            bne   :Age        ; rA > rD.
1678: E8       >41            inx               ; rA = rD so far...
1679: E0 06    >42            cpx   #6
167B: D0 F3    >43            bne   :complp
167D: F8       >44   :Age     sed               ; / Decimal mode.
167E: A5 9F    >45            lda   rA+EXP      ; rA >= rD. C = 1.
1680: E5 AB    >46            sbc   rD+EXP      ; Operand misalignment
1682: F0 3A    >47            beq   :doarith    ; Misalignment = 0, go.
1684: C9 08    >48            cmp   #8          ; Is misalignment > 7?
1686: B0 60    >49            bcs   :done       ; -Yes, rA unchanged.
1688: 4A       >50            lsr               ; -No, div by 2, C = odd.
1689: 90 0E    >51            bcc   :bytesh     ; Even, so shift bytes.
168B: A2 04    >52            ldx   #4          ; Odd.  4 bits / digit.
168D: 18       >53   :digsh   clc               ; Shift rD right 1 digit.
168E: 66 AC    >54            ror   rD+MANT
1690: 66 AD    >55            ror   rD+MANT+1
1692: 66 AE    >56            ror   rD+MANT+2
1694: 66 AF    >57            ror   rD+MANT+3
1696: CA       >58            dex
1697: D0 F4    >59            bne   :digsh
1699: A8       >60   :bytesh  tay               ; Byte shift count
169A: F0 22    >61            beq   :doarith    ; -Ready to go.
169C: A5 AE    >62            lda   rD+MANT+2   ; -Shift right 2 digits
169E: 85 AF    >63            sta   rD+MANT+3
16A0: A5 AD    >64            lda   rD+MANT+1
16A2: 85 AE    >65            sta   rD+MANT+2
16A4: A5 AC    >66            lda   rD+MANT
```

```
16A6: 85 AD    >67            sta   rD+MANT+1
16A8: A9 00    >68            lda   #0
16AA: 85 AC    >69            sta   rD+MANT
16AC: 88       >70            dey
16AD: D0 EA    >71            bne   :bytesh    ; (always)
               >72
16AF: A2 05    >73   :Alt     ldx   #5         ; Exchange rA and rD
16B1: B5 9E    >74   :exchAD  lda   rA,x       ;  so |rA| > |rD|.
16B3: B4 AA    >75            ldy   rD,x
16B5: 94 9E    >76            sty   rA,x
16B7: 95 AA    >77            sta   rD,x
16B9: CA       >78            dex
16BA: 10 F5    >79            bpl   :exchAD
16BC: 30 BF    >80            bmi   :Age       ; Now |rA| >= |rD|.
               >81
16BE: A5 9E    >82   :doarith lda   rA+S       ; Compare signs.
16C0: C5 AA    >83            cmp   rD+S
16C2: D0 28    >84            bne   :subtr     ; -Different, subtract.
16C4: A2 03    >85            ldx   #3         ; -Same, add.
16C6: 18       >86            clc
16C7: B5 A0    >87   :add     lda   rA+MANT,x  ; rA mantissa =
16C9: 75 AC    >88            adc   rD+MANT,x  ;  rA mantissa +
16CB: 95 A0    >89            sta   rA+MANT,x  ;   rD mantissa.
16CD: CA       >90            dex
16CE: 10 F7    >91            bpl   :add
16D0: 90 16    >92            bcc   :done      ; -No carry out.
16D2: A2 04    >93            ldx   #4         ; 4 bits / digit.
16D4: 20 44 1D >94   :srloop  jsr   srAM       ; -Shift mant 1 dig right.
16D7: 18       >95            clc              ; Shift in zeroes.
16D8: CA       >96            dex
16D9: D0 F9    >97            bne   :srloop
16DB: 18       >98            clc
16DC: A5 9F    >99            lda   rA+EXP     ; Increment rA exponent.
16DE: 69 01    >100           adc   #1
16E0: 85 9F    >101           sta   rA+EXP
16E2: 90 04    >102           bcc   :done      ; -No overflow.
               >103           seti  Ov         ; -Signal exponent overflow.
16E4: A9 FF    >103           lda   #$FF
16E6: 85 C3    >103           sta   Ov         ; Set non-zero.
               >103           eom
16E8: D8       >104  :done    cld              ; \ Back to binary.
16E9: 4C 72 0B >105           jmp   fetch
               >106
16EC: A2 03    >107  :subtr   ldx   #3         ; Subtract.
16EE: 38       >108           sec
16EF: B5 A0    >109  :sub     lda   rA+MANT,x  ; rA mantissa =
16F1: F5 AC    >110           sbc   rD+MANT,x  ;  rA mantissa -
16F3: 95 A0    >111           sta   rA+MANT,x  ;   rD mantissa.
16F5: 05 D0    >112           ora   t1         ; Summarize zero
16F7: 85 D0    >113           sta   t1         ;  mantissa.
16F9: CA       >114           dex
16FA: 10 F3    >115           bpl   :sub
16FC: A5 D0    >116           lda   t1         ; Result mantissa = 0?
16FE: D0 04    >117           bne   :norm      ; -No, normalize.
1700: 85 9F    >118           sta   rA+EXP     ; -Yes, exponent = 0.
1702: F0 E4    >119           beq   :done      ; (always)
               >120
1704: A5 A0    >121  :norm    lda   rA+MANT    ; Normalize result.
1706: 29 F0    >122           and   #$F0       ; Hi digit = 0?
1708: D0 DE    >123           bne   :done      ; -No, all done.
170A: A2 04    >124           ldx   #4         ; -Yes, shift left 1 dig.
170C: 18       >125  :diglp   clc              ; Shift in zeroes.
170D: 26 A3    >126           rol   rA+MANT+3
170F: 26 A2    >127           rol   rA+MANT+2
1711: 26 A1    >128           rol   rA+MANT+1
1713: 26 A0    >129           rol   rA+MANT
1715: CA       >130           dex
```

```
1716: D0 F4    >131           bne   :diglp
1718: C6 D1    >132           dec   NN          ; Norm limit exceeded?
171A: 10 04    >133           bpl   :ok         ; -No, continue.
               >134           resi  RUN         ; -Limit exceeded, halt.
171C: A9 00    >134           lda   #0
171E: 85 C0    >134           sta   RUN         ; Zero indicator.
               >134           eom
1720: 38       >135  :ok      sec
1721: A5 9F    >136           lda   rA+EXP      ; Decrement rA exponent
1723: E9 01    >137           sbc   #1
1725: 85 9F    >138           sta   rA+EXP
1727: D0 DB    >139           bne   :norm
1729: A2 9E    >140           ldx   #rA         ; Exponent underflow,
172B: 20 A8 1D >141           jsr   clear       ;  clear rA.
172E: 4C E8 16 >142           jmp   :done
```

```
1731: 18       >144  FMU      clc             ; Floating MUltiply
1732: C8       >145          iny             ; Y = 1 (exponent)
1733: F8       >146          sed             ; / Decimal mode.
1734: B1 CA    >147          lda   (memptr),y ; Operand exponent
1736: 85 CC    >148          sta   ptr        ; Save for restoration.
1738: 65 9F    >149          adc   rA+EXP     ;  + rA exponent
173A: 90 06    >150          bcc   :notov     ; No overflow.
173C: C9 50    >151          cmp   #$50       ; Sum < 150?
173E: 90 06    >152          bcc   :ok        ; -Yes, no overflow.
1740: B0 60    >153          bcs   :ovflow    ; -No, overflow (always)
               >154
1742: C9 50    >155  :notov   cmp   #$50       ; Sum < 50?
1744: 90 6C    >156          bcc   :unflow    ; -Yes, underflow.
1746: 38       >157  :ok      sec             ; -No, subtract extra
1747: E9 50    >158          sbc   #$50       ;   excess 50 and
1749: 85 D1    >159          sta   NN         ;    save result exponent.
174B: A9 00    >160          lda   #0         ; Clear operand and
174D: 91 CA    >161          sta   (memptr),y ;  rA exponents.
174F: 85 9F    >162          sta   rA+EXP
1751: A5 A0    >163          lda   rA+MANT    ; Is rA unnormalized?
1753: 29 F0    >164          and   #$F0
1755: F0 5B    >165          beq   :unflow    ; -Yes, underflow.
1757: C8       >166          iny             ; Y = 2 (mantissa)
1758: B1 CA    >167          lda   (memptr),y ; Is memory operand
175A: 29 F0    >168          and   #$F0       ;  unnormalized?
175C: F0 54    >169          beq   :unflow    ; -Yes, underflow.
175E: A5 AA    >170          lda   rD+S       ; Recover operand sign.
1760: 20 2B 14 >171          jsr   multiply   ; Do the multiply.
1763: A2 02    >172          ldx   #2         ; Shift rA & rR left
1765: B5 9F    >173  :shloop  lda   rA+1,x     ;  one byte.
1767: 95 9E    >174          sta   rA,x
1769: E8       >175          inx
176A: E0 06    >176          cpx   #6         ; Skip rR sign byte.
176C: D0 05    >177          bne   :notsign
176E: A5 A5    >178          lda   rR+1
1770: 85 A3    >179          sta   rA+5
1772: E8       >180          inx
1773: E0 0B    >181  :notsign cpx   #11        ; Done?
1775: D0 EE    >182          bne   :shloop    ; -No, continue.
1777: A9 00    >183          lda   #0         ; -Yes, clear
1779: 85 A9    >184          sta   rR+5       ;   low byte of rR.
177B: A5 A0    >185          lda   rA+MANT    ; Is rA normalized?
177D: 29 F0    >186          and   #$F0
177F: D0 13    >187          bne   :normal    ; -Yes.
1781: A0 04    >188          ldy   #4         ; -No, shift rA & rR
1783: 18       >189  :shdig   clc             ;   left one digit.
1784: 20 71 1D >190          jsr   slT
1787: 88       >191          dey
1788: D0 F9    >192          bne   :shdig
178A: A5 D1    >193          lda   NN         ; Recover result exp
178C: F0 24    >194          beq   :unflow    ; Underflow if 0.
178E: F8       >195          sed             ; / Decimal mode.
178F: 38       >196          sec
1790: E9 01    >197          sbc   #1         ; Compensate for shift.
1792: 85 D1    >198          sta   NN
1794: A5 D1    >199  :normal  lda   NN
1796: 85 9F    >200          sta   rA+EXP     ; Set result exponent.
1798: D8       >201  :done    cld             ; \ Binary mode.
1799: A0 01    >202          ldy   #1         ; Restore memory
179B: A5 CC    >203          lda   ptr        ;  operand's exponent.
179D: 91 CA    >204          sta   (memptr),y
179F: 4C 72 0B >205          jmp   fetch
               >206
               >207  :ovflow  seti  Ov         ; Set overflow indicator
17A2: A9 FF    >207          lda   #$FF
17A4: 85 C3    >207          sta   Ov         ; Set non-zero.
               >207          eom
```

```
17A6: A2 9E    >208           ldx   #rA        ; Clear rA.
17A8: 20 A8 1D >209           jsr   clear
17AB: 85 A4    >210           sta   rR+S       ; Clear sign and
17AD: 85 A5    >211           sta   rR+1       ;  top byte of rR.
17AF: 4C 98 17 >212           jmp   :done
               >213
17B2: 20 B8 17 >214  :unflow  jsr   clearAR    ; Clear rA and rR
17B5: 4C 98 17 >215           jmp   :done      ;  and clean up.
               >216
17B8: A2 9E    >217  clearAR  ldx   #rA        ; Clear rA.
17BA: 20 A8 1D >218           jsr   clear
17BD: A2 A4    >219           ldx   #rR        ; Clear rR.
17BF: 20 A8 1D >220           jsr   clear
17C2: 60       >221           rts
```

```
17C3: C8       >223  FDV      iny             ; Floating DiVide (Y==>EXP)
17C4: B1 CA    >224           lda   (memptr),y ; Save MEM exponent
17C6: 85 CC    >225           sta   ptr        ;  for restoration
17C8: A9 00    >226           lda   #0         ;   and clear it for
17CA: 91 CA    >227           sta   (memptr),y ;    for divide.
17CC: C8       >228           iny             ; Y ==> MEM mantissa
17CD: B1 CA    >229           lda   (memptr),y ; Hi byte of mant
17CF: 29 F0    >230           and   #$F0       ; Divisor normalized?
17D1: F0 54    >231           beq   :ovflo     ; -No, overflow.
17D3: A5 A0    >232           lda   rA+MANT    ; Hi byte of rA mant
17D5: 29 F0    >233           and   #$F0       ; Dividend normalized?
17D7: F0 55    >234           beq   :unflo     ; -No, underflow.
17D9: F8       >235           sed             ; /Decimal mode.
17DA: 38       >236           sec
17DB: A5 9F    >237           lda   rA+EXP     ; Dividend exponent
17DD: E5 CC    >238           sbc   ptr        ;  - divisor exponent.
17DF: B0 07    >239           bcs   :chkov     ; *dend >= *isor, ck ovflo.
17E1: 38       >240           sec             ; *dend <  *isor, ck unflo.
17E2: E9 50    >241           sbc   #$50       ; Restore excess-50
17E4: 90 48    >242           bcc   :unflo     ; Exponent underflow.
17E6: B0 05    >243           bcs   :ok        ; (always)
               >244
17E8: 18       >245  :chkov   clc
17E9: 69 50    >246           adc   #$50       ; Restore excess-50
17EB: B0 3A    >247           bcs   :ovflo     ; Exponent overflow.
17ED: 85 D1    >248  :ok      sta   NN         ; Save result exponent.
17EF: A9 00    >249           lda   #0         ; Clear rA exponent
17F1: 85 9F    >250           sta   rA+EXP     ;  for divide.
17F3: A0 04    >251           ldy   #4         ; 4 bits/digit.
17F5: 18       >252  :shrt    clc             ; Shift in zeros.
17F6: 20 4F 1D >253           jsr   srAMR      ; Shift rA mant & rR
17F9: 88       >254           dey             ;  right one digit.
17FA: D0 F9    >255           bne   :shrt
17FC: A5 AA    >256           lda   rD+S       ; Y=0, A=MEM sign
17FE: 20 AC 14 >257           jsr   divide     ; Do the divide.
1801: A5 9F    >258           lda   rA+1       ; Hi byte of quotient.
1803: 29 F0    >259           and   #$F0       ; Is hi digit = 0?
1805: D0 0B    >260           bne   :shrT2     ; -No, shift right 2 digs.
1807: A0 04    >261           ldy   #4         ; -Yes, shift right 1 dig.
1809: 18       >262  :shloop  clc             ; Shift in zeros.
180A: 20 4D 1D >263           jsr   srT        ; Shift |rA| & |rR|
180D: 88       >264           dey             ;  right one digit.
180E: D0 F9    >265           bne   :shloop
1810: F0 07    >266           beq   :setexp    ; (always)
               >267
1812: 20 5D 1D >268  :shrT2   jsr   srT2       ; Make room for exponent
1815: E6 D1    >269           inc   NN         ; EXP = EXP + 1
1817: F0 0E    >270           beq   :ovflo     ; Exponent overflow
1819: A5 D1    >271  :setexp  lda   NN         ; Set quotient exponent.
181B: 85 9F    >272           sta   rA+EXP
181D: A5 CC    >273  :done    lda   ptr        ; Recover MEM exponent
181F: A0 01    >274           ldy   #1         ;  and put it back into
1821: 91 CA    >275           sta   (memptr),y ;   divisor in memory.
1823: D8       >276           cld             ; \ Binary mode.
1824: 4C 72 0B >277           jmp   fetch
               >278
               >279  :ovflo   seti  Ov         ; Set Overflow indicator
1827: A9 FF    >279           lda   #$FF
1829: 85 C3    >279           sta   Ov         ; Set non-zero.
               >279           eom
182B: 4C 1D 18 >280           jmp   :done      ;  and finish up.
               >281
182E: 20 B8 17 >282  :unflo   jsr   clearAR    ; Clear rA and rR
1831: 4C 1D 18 >283           jmp   :done      ;  and finish up.
```

```
1834: A9 18    >285  IFL      lda   #CLCop     ; Patch ]dfl for IFL
1836: 8D D3 18 >286          sta   ]clc
1839: A9 65    >287          lda   #ADCZop
183B: 8D E2 18 >288          sta   ]adc
183E: A9 C9    >289          lda   #CMPIop
1840: 8D E4 18 >290          sta   ]cmp
1843: A9 EA    >291          lda   #NOPop
1845: 8D 0C 19 >292          sta   ]nop
1848: A9 79    >293          lda   #ADCYop
184A: 8D 0F 19 >294          sta   ]sub
184D: A9 C3    >295          lda   #Ov
184F: 8D 2E 19 >296          sta   ]Ov+3
1852: 20 9F 18 >297          jsr   ]dfl       ; Do the IFL.
1855: A9 C4    >298          lda   #Rp         ; Patch ]dfl back.
1857: 8D 2E 19 >299          sta   ]Ov+3
185A: A9 F9    >300          lda   #SBCYop
185C: 8D 0F 19 >301          sta   ]sub
185F: A9 38    >302          lda   #SECop
1861: 8D 0C 19 >303          sta   ]nop
1864: A9 24    >304          lda   #BITZop
1866: 8D E4 18 >305          sta   ]cmp
1869: A9 E5    >306          lda   #SBCZop
186B: 8D E2 18 >307          sta   ]adc
186E: A9 EA    >308          lda   #NOPop
1870: 8D D3 18 >309          sta   ]clc
1873: A5 C1    >310          lda   ERR         ; Error detected?
1875: D0 10    >311          bne   ]errpt      ; -Yes, report it.
1877: 4C 72 0B >312  ]fetch4  jmp   fetch
               >313
               >314  DFL      resi  Rp          ; Reset Repeat indicator.
187A: A9 00    >314          lda   #0
187C: 85 C4    >314          sta   Rp          ; Zero indicator.
               >314          eom
187E: 20 9F 18 >315          jsr   ]dfl        ; Decrease FieLd
1881: A5 C1    >316          lda   ERR         ; Error detected?
1883: D0 02    >317          bne   ]errpt      ; -Yes, report it.
1885: F0 F0    >318          beq   ]fetch4     ; (always)
               >319
1887: 4C 4C 0C >320  ]errpt   jmp   ]err
               >321
               >322  DLB      resi  Rp          ; Reset Repeat indicator.
188A: A9 00    >322          lda   #0
188C: 85 C4    >322          sta   Rp          ; Zero indicator.
               >322          eom
188E: 20 9F 18 >323          jsr   ]dfl        ; Decrease Field
1891: A5 AD    >324          lda   rD+3        ; Load rB from rD 8:4.
1893: 85 94    >325          sta   rB
1895: A5 AE    >326          lda   rD+4
1897: 85 95    >327          sta   rB+1
1899: A5 C1    >328          lda   ERR         ; Error detected?
189B: D0 EA    >329          bne   ]errpt      ; -Yes, report it.
189D: F0 D8    >330          beq   ]fetch4     ; (always)
```

```
189F: A2 AA    >332  ]dfl     ldx   #rD        ; Clear rD.
18A1: 20 A8 1D >333           jsr   clear
18A4: A2 B0    >334           ldx   #rD10      ; Clear rD10.
18A6: 20 A8 1D >335           jsr   clear
18A9: 20 94 1D >336           jsr   splitsL    ; A = s, X = L
18AC: 18       >337           clc
18AD: 69 01    >338           adc   #1         ; A = s + 1
18AF: 4A       >339           lsr              ; A = (s+1)/2, C = even dig
18B0: 08       >340           php              ; Push Carry status.
18B1: A8       >341           tay              ; Y = low byte index
18B2: A5 9A    >342           lda   rC+VV      ; NN
18B4: 99 B0 00 >343           sta   rD10,y     ; rD10 = subtrahend
18B7: B0 16    >344           bcs   :subtr     ; Even dig first, no shift.
18B9: 86 D0    >345           stx   t1         ; Save X
18BB: 98       >346           tya              ; Move Y to X.
18BC: AA       >347           tax
18BD: 16 B0    >348           asl   rD10,x     ; Odd dig first, shift
18BF: 36 AF    >349           rol   rD10-1,x   ;  1 digit left.
18C1: 16 B0    >350           asl   rD10,x
18C3: 36 AF    >351           rol   rD10-1,x
18C5: 16 B0    >352           asl   rD10,x
18C7: 36 AF    >353           rol   rD10-1,x
18C9: 16 B0    >354           asl   rD10,x
18CB: 36 AF    >355           rol   rD10-1,x
18CD: A6 D0    >356           ldx   t1         ; Restore X.
18CF: 28       >357  :subtr   plp              ; Pop C.
18D0: F8       >358           sed              ; / Decimal mode.
18D1: 90 39    >359           bcc   ]nop       ; Not C = odd dig first.
18D3: EA       >360  ]clc     nop              ; <Patch to CLC for IFL>
18D4: CA       >361  :evendig dex              ; Both even and odd digs?
18D5: D0 36    >362           bne   :byte      ; -Yes, subtr whole byte.
18D7: B9 B0 00 >363           lda   rD10,y     ; -No, subtr final digit.
18DA: 29 0F    >364           and   #$0F       ; Isolate even digit
18DC: 85 D0    >365           sta   t1         ;  and save for subtract.
18DE: B1 CA    >366           lda   (memptr),y ; MEM byte
18E0: 29 0F    >367           and   #$0F       ; Isolate even digit
18E2: E5 D0    >368  ]adc     sbc   t1         ;  & subtr. <ADC for IFL>
18E4: 24 10    >369  ]cmp     bit   $10        ; CMP# if IFL (to set C)
18E6: 29 0F    >370           and   #$0F       ; Mask result
18E8: 85 D0    >371           sta   t1         ;  and save it.
18EA: B1 CA    >372           lda   (memptr),y ; Recover MEM byte,
18EC: 29 F0    >373           and   #$F0       ;  mask out even digit,
18EE: 05 D0    >374           ora   t1         ;   OR in difference,
18F0: 91 CA    >375           sta   (memptr),y ;    and put it back.
18F2: A4 AE    >376           ldy   rD+4       ; Save high 4 digits of
18F4: 84 AF    >377           sty   rD+5       ;  difference in rD 8:4.
18F6: A4 AD    >378           ldy   rD+3
18F8: 84 AE    >379           sty   rD+4
18FA: 85 AD    >380           sta   rD+3
18FC: 08       >381           php              ; Push Carry status.
18FD: A2 04    >382           ldx   #4         ; 4 bits/digit
18FF: 26 AF    >383  :shlp    rol   rD+5       ; Shift rD left 1 digit
1901: 26 AE    >384           rol   rD+4       ;  to line up with rB.
1903: 26 AD    >385           rol   rD+3
1905: CA       >386           dex
1906: D0 F7    >387           bne   :shlp
1908: 28       >388           plp              ; Pop Carry status.
1909: 4C 28 19 >389           jmp   :done
               >390
190C: 38       >391  ]nop     sec              ; <Patch to NOP for IFL>
190D: B1 CA    >392  :byte    lda   (memptr),y ; MEM byte
190F: F9 B0 00 >393  ]sub     sbc   rD10,y     ;  minus subtrahend
1912: 91 CA    >394           sta   (memptr),y ;   back to MEM.
1914: 84 D0    >395           sty   t1         ; Save Y
1916: A4 AE    >396           ldy   rD+4       ; Save 4 hi digits of
1918: 84 AF    >397           sty   rD+5       ;  difference in rD 8:4.
191A: A4 AD    >398           ldy   rD+3
```

```
191C: 84 AE    >399           sty   rD+4
191E: 85 AD    >400           sta   rD+3
1920: A4 D0    >401           ldy   t1          ; Restore Y
1922: 88       >402           dey
1923: 30 0B    >403           bmi   :flderr     ; Field error.
1925: CA       >404           dex               ; More digits?
1926: D0 AC    >405           bne   :evendig    ; -Yes, keep subtracting.
1928: D8       >406  :done    cld               ; \ -No. Back to binary.
1929: 90 04    >407           bcc   :noRpt      ; Underflow ==> no Rpt
               >408  ]Ov      seti  Rp          ; Set Rpt <Ov for IFL>
192B: A9 FF    >408           lda   #$FF
192D: 85 C4    >408           sta   Rp          ; Set non-zero.
               >408           eom
192F: 60       >409  :noRpt   rts
               >410
1930: A9 C6    >411  :flderr  lda   #"F"        ; Signal Field error
1932: 85 C1    >412           sta   ERR
1934: D8       >413           cld               ; Clear decimal mode.
1935: 60       >414           rts
```

```
1936: 84 CF    >416  RTF      sty   inptr+1    ; 'inptr+1' = 0
1938: 84 D0    >417           sty   t1         ; 't1' = 0
193A: 20 B5 1D >418           jsr   midNN      ; Extract NN (word count)
193D: 85 CE    >419           sta   inptr      ; Save binary NN (1..100)
193F: A6 95    >420           ldx   rB+1       ; Convert rB to MEM
1941: E0 9A    >421           cpx   #$99+1     ;  address in 'ptr'.
1943: B0 51    >422           bcs   :underr    ; Undigit error.
1945: A4 94    >423           ldy   rB
1947: C0 4A    >424           cpy   #$49+1
1949: B0 4E    >425           bcs   :addrerr   ; Address error.
194B: BD 8B 1E >426           lda   BCDLadrl,x
194E: 79 BF 1F >427           adc   BCDHadrl,y
1951: 85 CC    >428           sta   ptr
1953: BD 25 1F >429           lda   BCDLadrh,x
1956: 79 09 20 >430           adc   BCDHadrh,y
1959: B0 3B    >431           bcs   :underr    ; Carry out ==> undigit.
195B: 85 CD    >432           sta   ptr+1      ; 'ptr' = dest MEM addr.
195D: A5 CE    >433           lda   inptr      ; Binary NN
195F: 0A       >434           asl              ; NN * 2 (2..200)
1960: 65 CE    >435           adc   inptr      ; NN * 3 (3..300)
1962: 26 CF    >436           rol   inptr+1    ; Capture high bit.
1964: 0A       >437           asl
1965: 26 CF    >438           rol   inptr+1    ; NN * 6 (6..600)
1967: AA       >439           tax              ; Byte count lo
1968: A0 00    >440           ldy   #0
196A: B1 CA    >441  :movelp  lda   (memptr),y ; Move bytes upward.
196C: 91 CC    >442           sta   (ptr),y
196E: CA       >443           dex              ; Dec byte count lo
196F: F0 09    >444           beq   :ckhi      ; If 0, chk hi byte.
1971: C8       >445  :cont    iny
1972: D0 F6    >446           bne   :movelp
1974: E6 CB    >447           inc   memptr+1   ; Advance ptr pages
1976: E6 CD    >448           inc   ptr+1
1978: D0 F0    >449           bne   :movelp    ; (always)
               >450
197A: C6 CF    >451  :ckhi    dec   inptr+1    ; Dec byte count hi
197C: 10 F3    >452           bpl   :cont      ; Continue if >= 0.
197E: A5 D1    >453           lda   NN         ; NN = 00 (100)?
1980: D0 02    >454           bne   :lt100     ; -No, less than 100.
1982: E6 D0    >455           inc   t1         ; -Yes, set 100.
1984: F8       >456  :lt100   sed              ; / Decimal mode.
1985: 18       >457           clc
1986: A5 95    >458           lda   rB+1       ; rB = rB + NN
1988: 65 D1    >459           adc   NN
198A: 85 95    >460           sta   rB+1
198C: A5 94    >461           lda   rB
198E: 65 D0    >462           adc   t1         ; 1 if NN = 0, else 0.
1990: 85 94    >463           sta   rB
1992: D8       >464           cld              ; \ Back to binary.
1993: 4C 72 0B >465           jmp   fetch
               >466
1996: 4C 4A 0C >467  :underr  jmp   UNDIGerr   ; Relay jump.
1999: 4C 40 0C >468  :addrerr jmp   ADDRerr    ; Relay jump.
```

```
199C: F8       >470  IBB      sed             ; / Decimal mode.
199D: 18       >471           clc
199E: A5 95    >472           lda   rB+1      ; rB = rB + rC(4:4)
19A0: 65 9A    >473           adc   rC+VV
19A2: 85 95    >474           sta   rB+1
19A4: A5 94    >475           lda   rB
19A6: 65 99    >476           adc   rC+sL
19A8: 85 94    >477           sta   rB
19AA: D8       >478           cld             ; \ Back to binary.
19AB: 90 58    >479           bcc   BUN       ; No overflow ==> branch
19AD: B0 66    >480           bcs   ]fetch3   ; Overflow ==> continue
               >481
19AF: F8       >482  DBB      sed             ; / Decimal mode.
19B0: 38       >483           sec
19B1: A5 95    >484           lda   rB+1      ; rB = rB - rC(4:4)
19B3: E5 9A    >485           sbc   rC+VV
19B5: 85 95    >486           sta   rB+1
19B7: A5 94    >487           lda   rB
19B9: E5 99    >488           sbc   rC+sL
19BB: 85 94    >489           sta   rB
19BD: D8       >490           cld             ; \ Back to binary.
19BE: B0 45    >491           bcs   BUN       ; No underflow ==> branch
19C0: 90 53    >492           bcc   ]fetch3   ; Underflow. (always)
```

```
19C2: A5 C3    >494  BOF      lda   Ov         ; Overflow indicator set?
19C4: D0 02    >495           bne   :ovflo     ; -Yes, clear it and branch.
19C6: F0 4D    >496           beq   ]fetch3    ; (always)
               >497
               >498  :ovflo   resi  Ov         ; Reset Overflow indicator
19C8: A9 00    >498           lda   #0
19CA: 85 C3    >498           sta   Ov         ; Zero indicator.
               >498           eom
19CC: 4C 05 1A >499           jmp   BUN        ;  and take the branch.
               >500
19CF: A5 C4    >501  BRP      lda   Rp         ; Repeat indicator set?
19D1: D0 32    >502           bne   BUN        ; -Yes, branch.
19D3: F0 40    >503           beq   ]fetch3    ; (always)
               >504
19D5: A5 9A    >505  BSA      lda   rC+VV      ; Get comparand digit
19D7: 29 0F    >506           and   #$0F
19D9: C5 9E    >507           cmp   rA+S       ; Equal to rA sign?
19DB: F0 28    >508           beq   BUN        ; -Yes, take branch.
19DD: D0 36    >509           bne   ]fetch3    ; (always)
               >510
19DF: A5 9A    >511  BCH      lda   rC+VV      ; BCH or BCL?
19E1: 29 01    >512           and   #$01
19E3: F0 06    >513           beq   :bch       ; -BCH.
19E5: A5 C2    >514           lda   COMP       ; -BCL.
19E7: 30 1C    >515           bmi   BUN        ; Branch if Lo
19E9: 10 2A    >516           bpl   ]fetch3    ; (always)
               >517
19EB: A5 C2    >518  :bch     lda   COMP
19ED: F0 26    >519           beq   ]fetch3    ; Equal.
19EF: 10 14    >520           bpl   BUN        ; Branch if Hi
19F1: 30 22    >521           bmi   ]fetch3    ; (always)
               >522
19F3: A5 9A    >523  BCE      lda   rC+VV      ; BCE or BCU?
19F5: 29 01    >524           and   #$01
19F7: F0 06    >525           beq   :bce       ; BCE.
19F9: A5 C2    >526           lda   COMP
19FB: D0 08    >527           bne   BUN        ; Branch if unequal.
19FD: F0 16    >528           beq   ]fetch3    ; (always)
               >529
19FF: A5 C2    >530  :bce     lda   COMP
1A01: F0 02    >531           beq   BUN        ; Branch if equal.
1A03: D0 10    >532           bne   ]fetch3    ; (always)
               >533
1A05: A5 9C    >534  BUN      lda   rC+ADDR    ; Set new P reg
1A07: 85 96    >535           sta   rP
1A09: A5 9D    >536           lda   rC+ADDR+1
1A0B: 85 97    >537           sta   rP+1
1A0D: A5 CA    >538           lda   memptr     ;  and instptr.
1A0F: 85 C8    >539           sta   instptr
1A11: A5 CB    >540           lda   memptr+1
1A13: 85 C9    >541           sta   instptr+1
1A15: 4C 72 0B >542  ]fetch3  jmp   fetch
```

```
1A18: A2 A4    >544  BFR      ldx   #rR        ; X points to rR
1A1A: D0 02    >545           bne   ]bfr
               >546
1A1C: A2 9E    >547  BFA      ldx   #rA        ; X points to rA
1A1E: A4 9A    >548  ]bfr     ldy   rC+VV      ; Y = 2-digit comparand
1A20: A5 99    >549           lda   rC+sL
1A22: 29 10    >550           and   #$10       ; s even or odd?
1A24: F0 0E    >551           beq   :even      ; -Even, no digit swap.
1A26: 98       >552           tya              ; -Odd, swap digits.
1A27: C9 80    >553           cmp   #$80       ; Hi bit to C
1A29: 2A       >554           rol              ;  and rotate 1 bit.
1A2A: C9 80    >555           cmp   #$80       ; Hi bit to C
1A2C: 2A       >556           rol              ;  and rotate 1 bit.
1A2D: C9 80    >557           cmp   #$80       ; Hi bit to C
1A2F: 2A       >558           rol              ;  and rotate 1 bit.
1A30: C9 80    >559           cmp   #$80       ; Hi bit to C
1A32: 2A       >560           rol              ;  and rotate 1 bit.
1A33: A8       >561           tay
1A34: 84 B5    >562  :even    sty   rD10+5     ; Expand comparand
1A36: 84 B4    >563           sty   rD10+4     ;  to full width in rD10.
1A38: 84 B3    >564           sty   rD10+3
1A3A: 84 B2    >565           sty   rD10+2
1A3C: 84 B1    >566           sty   rD10+1
1A3E: 98       >567           tya
1A3F: 29 0F    >568           and   #$0F       ; Mask off hi sign digit.
1A41: 85 B0    >569           sta   rD10
1A43: A5 CB    >570           lda   memptr+1   ; Push 'memptr' on stack.
1A45: 48       >571           pha
1A46: A5 CA    >572           lda   memptr
1A48: 48       >573           pha
1A49: A9 B0    >574           lda   #rD10      ; Point 'memptr' at rD10
1A4B: 85 CA    >575           sta   memptr
1A4D: A9 00    >576           lda   #0
1A4F: 85 CB    >577           sta   memptr+1
               >578
1A51: A0 01    >579           ldy   #1         ; Partial field compare
1A53: A9 B0    >580           lda   #BCSop     ; Unsigned compare
1A55: 20 8F 15 >581           jsr   compare
1A58: AA       >582           tax              ; Save A
1A59: 68       >583           pla              ; Pop 'memptr'
1A5A: 85 CA    >584           sta   memptr
1A5C: 68       >585           pla
1A5D: 85 CB    >586           sta   memptr+1
1A5F: A5 C1    >587           lda   ERR        ; Error detected?
1A61: D0 05    >588           bne   :err       ; -Yes, report it.
1A63: 8A       >589           txa              ; Recover COMP flags
1A64: F0 9F    >590           beq   BUN        ; -Branch if equal.
1A66: D0 6E    >591           bne   ]fetch2    ; -Else NOP. (always)
               >592
1A68: 4C 4C 0C >593  :err     jmp   ]err
               >594
1A6B: A5 99    >595  BCS      lda   rC+sL      ; Get switch #
1A6D: 4A       >596           lsr
1A6E: 4A       >597           lsr
1A6F: 4A       >598           lsr
1A70: 4A       >599           lsr
1A71: AA       >600           tax
1A72: B5 B6    >601           lda   CSW,x      ; Get switch state
1A74: D0 8F    >602           bne   BUN        ; -True, take branch.
1A76: F0 5E    >603           beq   ]fetch2    ; -False, no branch.
```

```
1A78: A5 9A    >605  SOR      lda   rC+VV      ; SOR / SOH / IOM?
1A7A: 29 0F    >606           and   #$0F
1A7C: C9 02    >607           cmp   #2         ; IOM?
1A7E: F0 05    >608           beq   :iom       ; -Yes.
1A80: 85 C7    >609           sta   OvHlt      ; -No, set Ovflo mode.
1A82: 4C 72 0B >610  :fetch   jmp   fetch
               >611
1A85: A5 C7    >612  :iom     lda   OvHlt
1A87: F0 F9    >613           beq   :fetch     ; No branch if SOR mode.
1A89: 4C 05 1A >614           jmp   BUN        ; Branch if SOH mode.
               >615
1A8C: A5 9A    >616  STA      lda   rC+VV      ; STA, STR, STB?
1A8E: 29 0F    >617           and   #$0F       ; Isolate reg variant.
1A90: A2 A4    >618           ldx   #rR
1A92: C9 01    >619           cmp   #1         ; STR?
1A94: F0 08    >620           beq   :store     ; -Yes.
1A96: A2 90    >621           ldx   #rBx
1A98: C9 02    >622           cmp   #2         ; STB?
1A9A: F0 02    >623           beq   :store     ; -Yes.
1A9C: A2 9E    >624           ldx   #rA        ; STA
1A9E: A5 9A    >625  :store   lda   rC+VV      ; Partial field :store?
1AA0: 29 10    >626           and   #$10
1AA2: D0 0F    >627           bne   :stfield   ; -Yes, do it.
1AA4: 8E AA 1A >628           stx   :stloop+1  ; -No, full word store.
1AA7: A0 05    >629           ldy   #5
1AA9: B9 00 00 >630  :stloop  lda   0*0,y      ; Store the register.
1AAC: 91 CA    >631           sta   (memptr),y
1AAE: 88       >632           dey
1AAF: 10 F8    >633           bpl   :stloop
1AB1: 30 23    >634           bmi   ]fetch2    ; (always)
               >635
1AB3: 8E C4 1A >636  :stfield stx   :evendig+1 ; Save register
1AB6: 8E DA 1A >637           stx   :odddig+1  ;  address...
1AB9: 20 94 1D >638           jsr   splitsL    ; Split sL: A = s and X = L
1ABC: 18       >639           clc
1ABD: 69 01    >640           adc   #1         ; A = s + 1
1ABF: 4A       >641           lsr              ; A = (s+1)/2, C = even dig
1AC0: A8       >642           tay              ; Y = byte offset
1AC1: 90 16    >643           bcc   :odddig    ; -Start digit is odd.
1AC3: B9 00 00 >644  :evendig lda   0*0,y      ; -Start digit is even.
1AC6: CA       >645           dex              ; Both even & odd digits?
1AC7: D0 1D    >646           bne   :byte      ; -Yes, move full byte.
1AC9: E8       >647           inx              ; -No, restore dig counter.
1ACA: 29 0F    >648           and   #$0F       ; Isolate even digit
1ACC: 85 D0    >649           sta   t1         ;  and save it.
1ACE: B1 CA    >650           lda   (memptr),y ; Get MEM byte,
1AD0: 29 F0    >651           and   #$F0       ;  clear target digit,
1AD2: 05 D0    >652           ora   t1         ;   OR in new digit,
1AD4: 91 CA    >653           sta   (memptr),y ;    and put it back.
1AD6: 4C 72 0B >654  ]fetch2  jmp   fetch      ; All done.
               >655
1AD9: B9 00 00 >656  :odddig  lda   0*0,y      ; Start digit is odd.
1ADC: 29 F0    >657           and   #$F0       ; Isolate reg digit
1ADE: 85 D0    >658           sta   t1         ;  and save it.
1AE0: B1 CA    >659           lda   (memptr),y ; Get MEM byte,
1AE2: 29 0F    >660           and   #$0F       ;  clear target digit,
1AE4: 05 D0    >661           ora   t1         ;   OR in new digit,
1AE6: 91 CA    >662  :byte    sta   (memptr),y ;    and put it back.
1AE8: 88       >663           dey              ; Move byte index.
1AE9: 30 05    >664           bmi   :flderr    ; -Err if field too long.
1AEB: CA       >665           dex              ; More digits?
1AEC: D0 D5    >666           bne   :evendig   ; -Yes, continue.
1AEE: F0 E6    >667           beq   ]fetch2    ; -No, finished. (always)
               >668
1AF0: 4C 3C 0C >669  :flderr  jmp   FIELDerr   ; Report field error.
```

```
1AF3: A0 05    >671  LDR      ldy   #5         ; MEM(ADDR) ==> rR
1AF5: B1 CA    >672  :ldr     lda   (memptr),y
1AF7: 99 A4 00 >673           sta   rR,y
1AFA: 88       >674           dey
1AFB: 10 F8    >675           bpl   :ldr
1AFD: 30 41    >676           bmi   ]fetch1    ; (always)
               >677
1AFF: A5 9A    >678  LDB      lda   rC+VV      ; LDB, LBC
1B01: A0 05    >679           ldy   #5
1B03: 29 01    >680           and   #$01
1B05: D0 0C    >681           bne   :lbc       ; Load rB Complement
1B07: B1 CA    >682  :ldb     lda   (memptr),y
1B09: 85 95    >683           sta   rB+1
1B0B: 88       >684           dey
1B0C: B1 CA    >685           lda   (memptr),y
1B0E: 85 94    >686           sta   rB
1B10: 4C 72 0B >687           jmp   fetch      ; -Yes, done.
               >688
1B13: F8       >689  :lbc     sed              ; / Decimal mode
1B14: 38       >690           sec              ;  for 10's complement.
1B15: A9 00    >691  :ldbc    lda   #0
1B17: F1 CA    >692           sbc   (memptr),y
1B19: 85 95    >693           sta   rB+1
1B1B: 88       >694           dey
1B1C: A9 00    >695           lda   #0
1B1E: F1 CA    >696           sbc   (memptr),y
1B20: 85 94    >697           sta   rB
1B22: D8       >698           cld              ; \ -Yes, back to binary.
1B23: 90 1B    >699           bcc   ]fetch1    ; (always)
               >700
1B25: A5 9A    >701  LSA      lda   rC+VV      ; Load Sign A
1B27: 29 0F    >702           and   #$0F       ; Isolate new sign digit
1B29: 85 9E    >703           sta   rA+S       ;  and put into rA.
1B2B: 4C 72 0B >704           jmp   fetch
```

```
1B2E: A0 05    >706  STP      ldy   #5         ; rP + 1 ==> MEM(0:4)
1B30: F8       >707           sed              ; / Decimal mode
1B31: 18       >708           clc
1B32: A5 97    >709           lda   rP+1
1B34: 69 01    >710           adc   #1
1B36: 91 CA    >711           sta   (memptr),y
1B38: 88       >712           dey
1B39: A5 96    >713           lda   rP
1B3B: 69 00    >714           adc   #0
1B3D: 91 CA    >715           sta   (memptr),y
1B3F: D8       >716           cld              ; \ Back to binary
1B40: 4C 72 0B >717  ]fetch1  jmp   fetch      ; -Yes, done.
               >718
1B43: A5 9A    >719  CLA      lda   rC+VV      ; CLA/R/B
1B45: 4A       >720           lsr              ; 1-bit to C
1B46: 85 D0    >721           sta   t1         ; Save mask
1B48: 90 05    >722           bcc   :notA      ; rA not included.
1B4A: A2 9E    >723           ldx   #rA
1B4C: 20 A8 1D >724           jsr   clear      ; Clear rA.
1B4F: 46 D0    >725  :notA    lsr   t1         ; 2-bit to C
1B51: 90 05    >726           bcc   :notR      ; rR not included.
1B53: A2 A4    >727           ldx   #rR
1B55: 20 A8 1D >728           jsr   clear      ; Clear rR.
1B58: 46 D0    >729  :notR    lsr   t1         ; 4-bit to C.
1B5A: 90 05    >730           bcc   :fetch     ; rB not included.
1B5C: A2 90    >731           ldx   #rBx
1B5E: 20 A8 1D >732           jsr   clear      ; Clear rB.
1B61: 4C 72 0B >733  :fetch   jmp   fetch
               >734
1B64: A9 00    >735  CLL      lda   #0         ; CLear Location
1B66: A0 05    >736           ldy   #5
1B68: 91 CA    >737  :cllloop sta   (memptr),y
1B6A: 88       >738           dey
1B6B: 10 FB    >739           bpl   :cllloop
1B6D: 30 D1    >740           bmi   ]fetch1    ; (always)
```

```
1B6F: A5 9D    >742  SRA      lda   rC+ADDR+1  ; SRA, SRT, SRS nn
1B71: 29 1F    >743           and   #$1F       ; Isolate count 0..19
1B73: C9 10    >744           cmp   #$10       ; Greater than 9?
1B75: 90 02    >745           bcc   :nocor     ; -No, don't correct.
1B77: E9 06    >746           sbc   #6         ; -Yes, cnvrt to binary.
1B79: AA       >747  :nocor   tax              ; X = shift count.
1B7A: A5 9A    >748           lda   rC+VV      ; SRA, SRT, SRS
1B7C: 29 0F    >749           and   #$0F
1B7E: C9 01    >750           cmp   #1         ; SRT?
1B80: F0 17    >751           beq   :srt       ; -Yes, shift right AR
1B82: E0 00    >752           cpx   #0         ; SRA or SRS, is count = 0?
1B84: F0 10    >753           beq   :fetch     ; -Yes, done.
1B86: C9 02    >754           cmp   #2         ; SRS?
1B88: F0 3F    >755           beq   :srs       ; -Yes, shift right A + Sign
1B8A: A0 04    >756  :sra     ldy   #4         ; 4 bits/digit
1B8C: 18       >757  :nxbita  clc
1B8D: 20 42 1D >758           jsr   srA        ; Shift 1 bit, not sign.
1B90: 88       >759           dey
1B91: D0 F9    >760           bne   :nxbita
1B93: CA       >761           dex              ; Count exhausted?
1B94: D0 F4    >762           bne   :sra       ; -No, shift again.
1B96: 4C 72 0B >763  :fetch   jmp   fetch      ; -Yes, done. (always)
               >764
1B99: A5 9E    >765  :srt     lda   rA+S       ; Copy rA sign
1B9B: 85 A4    >766           sta   rR+S       ;  to rR sign.
1B9D: 8A       >767           txa              ; Is count = 0?
1B9E: F0 F6    >768           beq   :fetch     ; -Yes, done.
1BA0: E0 0A    >769           cpx   #10        ; -No, count >= 10?
1BA2: 90 17    >770           bcc   :nxdig     ; -No, do general case.
1BA4: 86 D0    >771           stx   t1         ; -Yes, special case SRT >= 10.
1BA6: A0 00    >772           ldy   #0         ; SRT 10: rR = rA, rA = 0.
1BA8: A2 05    >773           ldx   #5
1BAA: B5 9E    >774  :srt10   lda   rA,x
1BAC: 95 A4    >775           sta   rR,x
1BAE: 94 9E    >776           sty   rA,x
1BB0: CA       >777           dex
1BB1: D0 F7    >778           bne   :srt10
1BB3: A5 D0    >779           lda   t1         ; Recover shift count
1BB5: 38       >780           sec
1BB6: E9 0A    >781           sbc   #10        ; Is count = 10?
1BB8: F0 DC    >782           beq   :fetch     ; -Yes, done.
1BBA: AA       >783           tax              ; -No, keep shifting.
1BBB: A0 04    >784  :nxdig   ldy   #4         ; 4 bits/digit
1BBD: 18       >785  :nxbitt  clc
1BBE: 20 4D 1D >786           jsr   srT        ; Shift |rA| & |rR|
1BC1: 88       >787           dey              ;  right 1 digit.
1BC2: D0 F9    >788           bne   :nxbitt
1BC4: CA       >789           dex              ; Count exhausted?
1BC5: D0 F4    >790           bne   :nxdig     ; -No, shift again.
1BC7: F0 CD    >791           beq   :fetch     ; -Yes, done. (always)
               >792
1BC9: A0 04    >793  :srs     ldy   #4         ; SRS, 4 bits/digit
1BCB: 18       >794  :nxbits  clc
1BCC: 20 40 1D >795           jsr   srAS       ; Shift rA (with sign)
1BCF: 88       >796           dey
1BD0: D0 F9    >797           bne   :nxbits
1BD2: CA       >798           dex              ; Count exhausted?
1BD3: D0 F4    >799           bne   :srs       ; -No, shift again.
1BD5: F0 BF    >800           beq   :fetch     ; (always)
```

```
1BD7: A5 9D    >802  SLA      lda   rC+ADDR+1  ; SLA, SLT, SLS nn
1BD9: 29 1F    >803           and   #$1F       ; Isolate count 0..19
1BDB: C9 10    >804           cmp   #$10       ; Greater than 9?
1BDD: 90 02    >805           bcc   :nocor     ; -No, don't correct.
1BDF: E9 06    >806           sbc   #6         ; -Yes, cnvrt to binary.
1BE1: AA       >807  :nocor   tax              ; X = shift count.
1BE2: A5 9A    >808           lda   rC+VV      ; SLA, SLT, SLS?
1BE4: 29 0F    >809           and   #$0F
1BE6: C9 01    >810           cmp   #1         ; SLT?
1BE8: F0 19    >811           beq   :slt       ; -Yes, shift left AR
1BEA: E0 00    >812           cpx   #0         ; -No, check count.
1BEC: F0 12    >813           beq   :fetch     ; Done if count = 0.
1BEE: C9 02    >814           cmp   #2         ; SLS?
1BF0: F0 39    >815           beq   :sls       ; -Yes, shift left A + Sign
1BF2: A5 9F    >816           lda   rA+1       ; SLA, hi magnitude digit
1BF4: 2A       >817           rol              ; High bit to C.
1BF5: A0 04    >818  :sla     ldy   #4         ; 4 bits/digit
1BF7: 20 7B 1D >819  :nxbita  jsr   slA        ; Rotate A left 1 bit.
1BFA: 88       >820           dey              ; More bits?
1BFB: D0 FA    >821           bne   :nxbita    ; -Yes.
1BFD: CA       >822           dex              ; More digits?
1BFE: D0 F5    >823           bne   :sla       ; -Yes.
1C00: 4C 72 0B >824  :fetch   jmp   fetch
               >825
1C03: A5 A4    >826  :slt     lda   rR+S       ; Copy rR Sign
1C05: 85 9E    >827           sta   rA+S       ;  to rA Sign.
1C07: 8A       >828           txa              ; Is count = 0?
1C08: F0 F6    >829           beq   :fetch     ; -Yes, done.
1C0A: E0 0A    >830           cpx   #10        ; -No, count >= 10?
1C0C: 90 10    >831           bcc   :nxdig     ; -No, do general case.
1C0E: 86 D0    >832           stx   t1         ; -Yes, special case SLT >= 10.
1C10: 20 86 1D >833           jsr   exchAR     ; Exchange A and R magnitudes
1C13: A5 D0    >834           lda   t1         ; Recover count.
1C15: 38       >835           sec
1C16: E9 0A    >836           sbc   #10        ; Is count = 10?
1C18: F0 E6    >837           beq   :fetch     ; -Yes, done.
1C1A: AA       >838           tax              ; -No, keep shifting.
1C1B: A5 9F    >839           lda   rA+1       ; Hi magnitude digit.
1C1D: 2A       >840           rol              ; High bit to C
1C1E: A0 04    >841  :nxdig   ldy   #4         ; 4 bits/digit
1C20: 20 71 1D >842  :nxbitt  jsr   slT        ; Rotate AR left 1 bit.
1C23: 88       >843           dey              ; More bits?
1C24: D0 FA    >844           bne   :nxbitt    ; -Yes.
1C26: CA       >845           dex              ; More digits?
1C27: D0 F5    >846           bne   :nxdig     ; -Yes.
1C29: F0 D5    >847           beq   :fetch     ; (always)
               >848
1C2B: A5 9E    >849  :sls     lda   rA+S       ; SLS, use sign digit
1C2D: 29 0F    >850           and   #$0F       ;  and mask it.
1C2F: A0 04    >851  :nxdigt  ldy   #4         ; 4 bits/digit
1C31: C9 08    >852  :nxbit   cmp   #8         ; Hi bit of sign to C
1C33: 20 7B 1D >853           jsr   slA        ; Rotate A left 1 bit
1C36: A5 9E    >854           lda   rA+S       ;  then rotate sign.
1C38: 2A       >855           rol
1C39: 29 0F    >856           and   #$0F       ; Mask again
1C3B: 85 9E    >857           sta   rA+S       ;  and put it back.
1C3D: 88       >858           dey              ; More bits?
1C3E: D0 F1    >859           bne   :nxbit     ; -Yes.
1C40: CA       >860           dex              ; More digits?
1C41: D0 EC    >861           bne   :nxdigt    ; -Yes.
1C43: F0 BB    >862           beq   :fetch     ; (always)
```

```
                >864  * Mag Tape Op Codes
                >865
                >866  blkcnt   equ   line2      ; Temp block count.
                >867
1C45: 20 65 12  >868  MTS      jsr   getMTt1    ; Set t1 to offset disp.
1C48: A5 9A     >869           lda   rC+VV      ; MTS...MRW
1C4A: 29 0F     >870           and   #$0F       ; Isolate variant
1C4C: C9 04     >871           cmp   #4         ; MLS?
1C4E: F0 14     >872           beq   :mls       ; -Yes, lane select.
1C50: C9 08     >873           cmp   #8         ; -No, MRW?
1C52: F0 03     >874           beq   :mrw       ; -Yes, rewind.
1C54: 4C 34 0C  >875           jmp   OPerr      ; -No, unimplemented.
                >876
1C57: A6 D0     >877  :mrw     ldx   t1
1C59: A9 00     >878           lda   #0         ; Set unit position
1C5B: 9D DD 1D  >879           sta   rdroff,x   ;  to zero.
1C5E: 9D DE 1D  >880           sta   rdroff+1,x
1C61: 9D DF 1D  >881           sta   rdroff+2,x
1C64: A5 9A     >882  :mls     lda   rC+VV
1C66: 29 10     >883           and   #$10       ; Isolate lane low bit.
1C68: F0 02     >884           beq   :lane0
1C6A: A9 01     >885           lda   #1         ; Lane 1
1C6C: AA        >886  :lane0   tax              ; Save lane #
1C6D: 98        >887           tya              ; fnx
1C6E: 4A        >888           lsr              ; A = 2 (unit 0) or 3 (unit 1)
1C6F: A8        >889           tay
1C70: 8A        >890           txa              ; Recover lane #.
1C71: 99 ED 1D  >891           sta   mtlane-2,y ; Select lane.
1C74: 4C 72 0B  >892           jmp   fetch
                >893
1C77: 4C 34 0C  >894  MTC      jmp   OPerr      ; Not implemented
                >895
1C7A: A9 04     >896  MRD      lda   #4         ; Mag tape device class
1C7C: 20 D0 11  >897           jsr   setread    ; Mag tape ReaD
1C7F: 20 FB 1C  >898           jsr   mtrw       ; Do the I/O.
1C82: 4C 72 0B  >899           jmp   fetch
                >900
1C85: 4C 34 0C  >901  MRR      jmp   OPerr      ; Not implemented
                >902
1C88: 84 D5     >903  MIW      sty   zeroff     ; New file if offset=0.
1C8A: 20 F6 1C  >904           jsr   mtwrite    ; Go write...
1C8D: 4C 72 0B  >905           jmp   fetch
                >906
1C90: 4C 34 0C  >907  MIR      jmp   OPerr      ; Not implemented
                >908
1C93: A9 FF     >909  MOW      lda   #$FF       ; Rewrite file if
1C95: 85 D5     >910           sta   zeroff     ;  offset = 0.
1C97: 20 F6 1C  >911           jsr   mtwrite    ; Go write...
1C9A: 4C 72 0B  >912           jmp   fetch
                >913
1C9D: 4C 34 0C  >914  MOR      jmp   OPerr      ; Not implemented
                >915
1CA0: 20 94 1D  >916  MPF      jsr   splitsL
1CA3: 86 D7     >917           stx   blkcnt     ; # of blocks
1CA5: 20 65 12  >918           jsr   getMTt1    ; Set t1 = unit offset disp
1CA8: A9 58     >919           lda   #<600      ; Set offset increment
1CAA: 85 CE     >920           sta   inptr      ;  in bytes.
1CAC: A9 02     >921           lda   #>600
1CAE: 85 CF     >922           sta   inptr+1
1CB0: A5 9A     >923           lda   rC+VV      ; Check subop.
1CB2: 29 0F     >924           and   #$0F
1CB4: F0 0B     >925           beq   :mpf       ; Mag tape Position Fwd
1CB6: C9 01     >926           cmp   #1
1CB8: F0 11     >927           beq   :mpb       ; Mag tape Position Back
1CBA: C9 02     >928           cmp   #2
1CBC: F0 0A     >929           beq   :mpe       ; Mag tape Position End
1CBE: 4C 34 0C  >930           jmp   OPerr      ; Unimplemented.
```

```
                    >931
1CC1: 20 C4 12      >932  :mpf     jsr   incoff     ; Inc offset 100 words.
1CC4: C6 D7         >933           dec   blkcnt     ; More?
1CC6: D0 F9         >934           bne   :mpf       ; -Yes, repeat.
1CC8: 4C 72 0B      >935  :mpe     jmp   fetch      ; -No, done.
                    >936
1CCB: A6 D0         >937  :mpb     ldx   t1         ; Index to unit offset.
1CCD: 38            >938           sec
1CCE: BD DF 1D      >939           lda   rdroff+2,x ; Decrement offset
1CD1: E5 CE         >940           sbc   inptr      ;  by 100 words.
1CD3: 9D DF 1D      >941           sta   rdroff+2,x
1CD6: BD DE 1D      >942           lda   rdroff+1,x
1CD9: E5 CF         >943           sbc   inptr+1
1CDB: 9D DE 1D      >944           sta   rdroff+1,x
1CDE: B0 03         >945           bcs   :nobor
1CE0: DE DD 1D      >946           dec   rdroff,x
1CE3: C6 D7         >947  :nobor   dec   blkcnt     ; More blocks?
1CE5: D0 E4         >948           bne   :mpb       ; -Yes, repeat.
1CE7: 4C 72 0B      >949           jmp   fetch      ; -No, done.
                    >950
1CEA: A5 9A         >951  MIB      lda   rC+VV      ; MIB / MIE
1CEC: 29 01         >952           and   #$01       ; Isolate variant.
1CEE: D0 03         >953           bne   :mie
1CF0: 4C 05 1A      >954           jmp   BUN        ; MIB always branches.
                    >955
1CF3: 4C 72 0B      >956  :mie     jmp   fetch      ; MIE never branches.
                    >957
1CF6: A9 04         >958  mtwrite  lda   #4         ; Mag tape device class
1CF8: 20 DF 11      >959           jsr   setwrite   ; Set to write file.
1CFB: 20 94 1D      >960  mtrw     jsr   splitsL
1CFE: 86 D7         >961           stx   blkcnt     ; # of blocks
1D00: A9 64         >962  :nxblock lda   #100       ; Set block length
1D02: 85 D1         >963           sta   NN         ;  to 100 words.
1D04: 20 F2 11      >964           jsr   doio       ; Do the I/O operation.
1D07: 20 C4 12      >965           jsr   incoff     ; Increment unit offset.
1D0A: A5 9B         >966           lda   rC+OP      ; Check opcode.
1D0C: C9 52         >967           cmp   #$52       ; MRD?
1D0E: D0 1E         >968           bne   :noBmod    ; -No, skip B-mod scan.
1D10: A5 9A         >969           lda   rC+VV      ; -Yes, does variant
1D12: 29 08         >970           and   #$08       ;       specify B-mod?
1D14: F0 18         >971           beq   :noBmod    ; -No, skip it.
1D16: A0 00         >972  :Bmod    ldy   #0         ; Scan mem for B-mod.
1D18: B1 CA         >973           lda   (memptr),y ; Get sign digit.
1D1A: C9 08         >974           cmp   #$08       ; Is sign 8 or 9?
1D1C: 90 07         >975           bcc   :next      ; -No, skip word.
1D1E: 29 01         >976           and   #$01       ; -Yes, reset 8 bit.
1D20: 91 CA         >977           sta   (memptr),y
1D22: 20 8B 11      >978           jsr   Bmodmem    ; Modify ADDR field.
1D25: 20 A1 11      >979  :next    jsr   incmem     ; Advance memptr.
1D28: C6 D1         >980           dec   NN         ; More words?
1D2A: D0 EA         >981           bne   :Bmod      ; -Yes, continue.
1D2C: F0 0D         >982           beq   :more      ; -No, done. (always)
                    >983
1D2E: 18            >984  :noBmod  clc
1D2F: A5 CA         >985           lda   memptr     ; Increment memptr
1D31: 69 58         >986           adc   #<600      ;  by 6 * 100.
1D33: 85 CA         >987           sta   memptr
1D35: A5 CB         >988           lda   memptr+1
1D37: 69 02         >989           adc   #>600
1D39: 85 CB         >990           sta   memptr+1
1D3B: C6 D7         >991  :more    dec   blkcnt     ; More blocks?
1D3D: D0 C1         >992           bne   :nxblock   ; -Yes, do them.
1D3F: 60            >993           rts              ; -No, done.
```

```
                >995  ************************************************************
                >996  *                                                          *
                >997  *            Utility Shifting Subroutines                  *
                >998  *                                                          *
                >999  ************************************************************
                >1000
1D40: 66 9E     >1001 srAS     ror   rA         ; rA & sign right 1 bit
1D42: 66 9F     >1002 srA      ror   rA+1       ; Sign not included
1D44: 66 A0     >1003 srAM     ror   rA+2       ; FP mantissa
1D46: 66 A1     >1004          ror   rA+3
1D48: 66 A2     >1005          ror   rA+4
1D4A: 66 A3     >1006          ror   rA+5
1D4C: 60        >1007          rts
                >1008
1D4D: 66 9F     >1009 srT      ror   rA+1       ; |rA| & |rR| right 1 bit
1D4F: 20 44 1D  >1010 srAMR    jsr   srAM       ; Shift rA Mantissa & |rR|
1D52: 66 A5     >1011 srR      ror   rR+1       ; Shift |rR|
1D54: 66 A6     >1012          ror   rR+2
1D56: 66 A7     >1013          ror   rR+3
1D58: 66 A8     >1014          ror   rR+4
1D5A: 66 A9     >1015          ror   rR+5
1D5C: 60        >1016          rts
                >1017
1D5D: A2 0A     >1018 srT2     ldx   #10        ; |rA| & |rR| right
1D5F: B5 9E     >1019 :shloop  lda   rA,x       ;  2 digits (1 byte).
1D61: E0 05     >1020          cpx   #5         ; About to store in rR+S?
1D63: D0 04     >1021          bne   :cont      ; -No, continue.
1D65: 85 A5     >1022          sta   rR+1       ; -Yes, skip rR sign.
1D67: F0 02     >1023          beq   :next      ;   and on to next byte.
1D69: 95 9F     >1024 :cont    sta   rA+1,x
1D6B: CA        >1025 :next    dex
1D6C: D0 F1     >1026          bne   :shloop    ; Exclude rA sign.
1D6E: 86 9F     >1027          stx   rA+1       ; Shift in zeros.
1D70: 60        >1028          rts
                >1029
1D71: 26 A9     >1030 slT      rol   rR+5       ; Rotate |rR| & |rA| left
1D73: 26 A8     >1031          rol   rR+4       ;  one bit.
1D75: 26 A7     >1032          rol   rR+3
1D77: 26 A6     >1033          rol   rR+2
1D79: 26 A5     >1034          rol   rR+1       ; Fall into slA.
                >1035
1D7B: 26 A3     >1036 slA      rol   rA+5       ; Rotate |rA| left 1 bit
1D7D: 26 A2     >1037          rol   rA+4
1D7F: 26 A1     >1038          rol   rA+3
1D81: 26 A0     >1039          rol   rA+2
1D83: 26 9F     >1040          rol   rA+1
1D85: 60        >1041          rts
                >1042
1D86: A2 05     >1043 exchAR   ldx   #5         ; Exchange |rA| and |rR|
1D88: B5 9E     >1044 :exch    lda   rA,x       ; (equivalent to SLT 10)
1D8A: B4 A4     >1045          ldy   rR,x
1D8C: 95 A4     >1046          sta   rR,x
1D8E: 94 9E     >1047          sty   rA,x
1D90: CA        >1048          dex
1D91: D0 F5     >1049          bne   :exch
1D93: 60        >1050          rts
```

```
                >1052 ************************************************************
                >1053 *                                                          *
                >1054 *          Split sL field into A = s and X = L             *
                >1055 *                                                          *
                >1056 ************************************************************
                >1057
1D94: A5 99     >1058 splitsL  lda   rC+sL      ; Get field specifier
1D96: 29 0F     >1059          and   #$0F       ; L = digit count
1D98: D0 02     >1060          bne   :notz
1D9A: A9 0A     >1061          lda   #10        ; "0" ==> 10
1D9C: AA        >1062 :notz    tax             ; X = digit count (L)
1D9D: A5 99     >1063          lda   rC+sL
1D9F: 4A        >1064          lsr             ; Isolate field start s
1DA0: 4A        >1065          lsr
1DA1: 4A        >1066          lsr
1DA2: 4A        >1067          lsr
1DA3: D0 02     >1068          bne   :ret
1DA5: A9 0A     >1069          lda   #10        ; "0" ==> 10
1DA7: 60        >1070 :ret     rts             ; A = start digit (s)
                >1071
                >1072 ************************************************************
                >1073 *                                                          *
                >1074 *                    Clear Register                        *
                >1075 *                                                          *
                >1076 * At entry: X = Register address                           *
                >1077 * At exit:  A = 0, X = $FF                                 *
                >1078 *                                                          *
                >1079 ************************************************************
                >1080
1DA8: 8E B0 1D  >1081 clear    stx   :clrloop+1 ; Save reg address
1DAB: A2 05     >1082          ldx   #5
1DAD: A9 00     >1083          lda   #0
1DAF: 95 00     >1084 :clrloop sta   0*0,x      ; Clear the register.
1DB1: CA        >1085          dex
1DB2: 10 FB     >1086          bpl   :clrloop
1DB4: 60        >1087          rts
                >1088
                >1089 ************************************************************
                >1090 *                                                          *
                >1091 *            Extract NN from 3:2 field of rC               *
                >1092 *                                                          *
                >1093 * Returns NN in BCD in 'NN', in binary in A. X unchanged.*
                >1094 *                                                          *
                >1095 ************************************************************
                >1096
1DB5: A5 99     >1097 midNN    lda   rC+sL      ; Extract NN from xN Nx.
1DB7: 29 0F     >1098          and   #$0F       ; Return binary NN in A
1DB9: A8        >1099          tay
1DBA: 0A        >1100          asl
1DBB: 0A        >1101          asl
1DBC: 0A        >1102          asl
1DBD: 0A        >1103          asl
1DBE: 85 D1     >1104          sta   NN         ; N0
1DC0: A5 9A     >1105          lda   rC+VV      ; Nx (low digit)
1DC2: 4A        >1106          lsr
1DC3: 4A        >1107          lsr
1DC4: 4A        >1108          lsr
1DC5: 4A        >1109          lsr             ; 0N
1DC6: 05 D1     >1110          ora   NN
1DC8: 85 D1     >1111          sta   NN         ; 'NN' = BCD NN
1DCA: 38        >1112          sec
1DCB: F9 D3 1D  >1113          sbc   bcdcor,y   ; Convert to binary.
1DCE: D0 02     >1114          bne   :ret
1DD0: A9 64     >1115          lda   #100       ; "00" ==> 100
1DD2: 60        >1116 :ret     rts
                >1117
1DD3: 00 06 0C  >1118 bcdcor   db    0,6,12,18,24,30,36,42,48,54
```

```
                 187           put   B220TABLES
                >1    * Byte offsets for paper tape reader & punch files
                >2    * for units 0..1 and mag tape units 0..1.
                >3
                >4    IOstate  equ   *          ; Start of I/O state.
                >5
1DDD: 00 00 00  >6    rdroff   ds    2*3        ; 3 bytes * 2 unit (0..1)
1DE3: 00 00 00  >7    pchoff   ds    2*3        ; (Offsets are big-endian)
1DE9: 00 00 00  >8    mtoff    ds    2*3        ; 3 bytes x 2 units (0..1)
1DEF: 00 00     >9    mtlane   db    0,0        ; MT units lane state 0..1
                >10
                >11   IOstend  equ   *          ; End of I/O state.
                >12
1DF1: 00 03 06  >13   fnxoff   db    0,3,6,9,12,12,15,15 ; fnx ==> offset
1DF9: 00 19 32  >14   fnxfn    db    0,25,50,75,100,125,150,175 ; fnx ==> fn
                >15
                >16   * $00..$89 B220 character code to ASCII
                >17
                >18   b220asc  equ   *          ; B220 code to ASCII
1E01: A0        >19           db    $A0        ; $00 = Blank
1E02: 00        >20           ds    1          ; $01 skip
1E03: 00        >21           db    $00        ; $02 = Ignore
1E04: AE A9     >22           asc   ".)"       ; $03..$04
1E06: 00 00 00  >23           ds    11         ; $05..$0F skip
1E11: A8        >24           asc   "("        ; $10
1E12: 00 00     >25           ds    2          ; $11..$12 skip
1E14: AB AA     >26           asc   "+*"       ; $13..$14
1E16: 8C        >27           db    $8C        ; $15 = Eject
1E17: 8D        >28           db    $8D        ; $16 = CR
1E18: 00 00 00  >29           ds    3+6        ; $17..$1F skip
1E21: AD AF     >30           asc   "-/"       ; $20..$21
1E23: 00        >31           ds    1          ; $22 skip
1E24: AC        >32           asc   ","        ; $23
1E25: A5        >33           asc   "%"        ; $24 (For SNAP CR translation)
1E26: 00        >34           ds    1          ; $25 skip
1E27: 89        >35           db    $89        ; $26 = TAB
1E28: A4        >36           asc   "$"        ; $27
1E29: 00 00 00  >37           ds    2+6+2      ; $28..$31 skip
1E33: BF BD A7  >38           asc   "?='"      ; $32..$34
1E36: 00 00 00  >39           ds    5+6+1      ; $35..$40 skip
1E42: C1 C2 C3  >40           asc   "ABCDEFGHI" ; $41..$49
1E4B: 00 00 00  >41           ds    6+1        ; $4A..$50 skip
1E52: CA CB CC  >42           asc   "JKLMNOPQR" ; $51..$59
1E5B: 00 00 00  >43           ds    6+2        ; $5A..$61 skip
1E63: D3 D4 D5  >44           asc   "STUVWXYZ" ; $62..$69
1E6B: 00 00 00  >45           ds    6+16       ; $6A..$7F skip
1E81: B0 B1 B2  >46           asc   "0123456789" ; $80..$89
```

```
                >48   * 4-digit BCD to binary word address tables
                >49
                >50   BCDLadrl equ   *          ; BCD lo 2 dig --> addr lo byte
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1E8B: D0        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1E8C: D6        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1E8D: DC        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1E8E: E2        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1E8F: E8        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1E90: EE        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1E91: F4        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1E92: FA        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1E93: 00        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1E94: 06        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1E95: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1E96: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
```

```
1E97: 00       >57             db    0
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1E98: 00       >57             db    0
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1E99: 00       >57             db    0
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1E9A: 00       >57             db    0
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1E9B: 0C       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1E9C: 12       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1E9D: 18       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1E9E: 1E       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1E9F: 24       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1EA0: 2A       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1EA1: 30       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1EA2: 36       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1EA3: 3C       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1EA4: 42       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
```

```
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EA5: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EA6: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EA7: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EA8: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EA9: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EAA: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EAB: 48        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EAC: 4E        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EAD: 54        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EAE: 5A        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EAF: 60        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EB0: 66        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EB1: 6C        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
```

```
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EB2: 72        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EB3: 78        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EB4: 7E        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EB5: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EB6: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EB7: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EB8: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EB9: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EBA: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EBB: 84        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EBC: 8A        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EBD: 90        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EBE: 96        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EBF: 9C        >59           db    <]T*10+]U*6+MEM
```

```
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EC0: A2        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EC1: A8        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EC2: AE        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EC3: B4        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EC4: BA        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EC5: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EC6: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EC7: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EC8: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EC9: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1ECA: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1ECB: C0        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1ECC: C6        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
```

```
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1ECD: CC        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1ECE: D2        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1ECF: D8        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1ED0: DE        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1ED1: E4        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1ED2: EA        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1ED3: F0        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1ED4: F6        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1ED5: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1ED6: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1ED7: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1ED8: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1ED9: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
```

```
1EDA: 00       >57             db    0
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1EDB: FC       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1EDC: 02       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1EDD: 08       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1EDE: 0E       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1EDF: 14       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1EE0: 1A       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1EE1: 20       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1EE2: 26       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1EE3: 2C       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1EE4: 32       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1EE5: 00       >57             db    0
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1EE6: 00       >57             db    0
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1EE7: 00       >57             db    0
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
```

```
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EE8: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EE9: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EEA: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EEB: 38        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EEC: 3E        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EED: 44        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EEE: 4A        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EEF: 50        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EF0: 56        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EF1: 5C        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EF2: 62        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EF3: 68        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EF4: 6E        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
```

```
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EF5: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EF6: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EF7: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EF8: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EF9: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EFA: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EFB: 74        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EFC: 7A        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EFD: 80        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EFE: 86        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1EFF: 8C        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F00: 92        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F01: 98        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F02: 9E        >59           db    <]T*10+]U*6+MEM
```

```
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F03: A4        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F04: AA        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F05: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F06: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F07: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F08: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F09: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F0A: 00        >57            db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F0B: B0        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F0C: B6        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F0D: BC        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F0E: C2        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F0F: C8        >59            db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
```

```
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F10: CE        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F11: D4        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F12: DA        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F13: E0        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F14: E6        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F15: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F16: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F17: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F18: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F19: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F1A: 00        >57           db    0
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F1B: EC        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
1F1C: F2        >59           db    <]T*10+]U*6+MEM
                >52   ]Ax      equ   *-BCDLadrl ; ]Ax = index of table entry
                >53   ]T       equ   ]Ax/16     ; BCD tens digit
                >54   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >55   ]U       equ   ]Ax-]A0    ; BCD units digit
```

```
1F1D: F8       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1F1E: FE       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1F1F: 04       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1F20: 0A       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1F21: 10       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1F22: 16       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1F23: 1C       >59             db    <]T*10+]U*6+MEM
               >52   ]Ax       equ   *-BCDLadrl ; ]Ax = index of table entry
               >53   ]T        equ   ]Ax/16     ; BCD tens digit
               >54   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >55   ]U        equ   ]Ax-]A0    ; BCD units digit
1F24: 22       >59             db    <]T*10+]U*6+MEM
               >62
               >63   BCDLadrh  equ   *          ; BCD lo 2 dig --> addr hi byte
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F25: 20       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F26: 20       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F27: 20       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F28: 20       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F29: 20       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
```

```
1F2A: 20       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F2B: 20       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F2C: 20       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F2D: 21       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F2E: 21       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F2F: FF       >70             db    $FF        ; Force overflow on undigits.
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F30: FF       >70             db    $FF        ; Force overflow on undigits.
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F31: FF       >70             db    $FF        ; Force overflow on undigits.
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F32: FF       >70             db    $FF        ; Force overflow on undigits.
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F33: FF       >70             db    $FF        ; Force overflow on undigits.
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F34: FF       >70             db    $FF        ; Force overflow on undigits.
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F35: 21       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F36: 21       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F37: 21       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
```

```
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F38: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F39: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F3A: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F3B: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F3C: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F3D: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F3E: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F3F: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F40: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F41: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F42: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F43: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F44: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
```

```
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F45: 21        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F46: 21        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F47: 21        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F48: 21        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F49: 21        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F4A: 21        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F4B: 21        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F4C: 21        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F4D: 21        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F4E: 21        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F4F: FF        >70           db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F50: FF        >70           db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F51: FF        >70           db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F52: FF        >70           db    $FF        ; Force overflow on undigits.
```

```
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F53: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F54: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F55: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F56: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F57: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F58: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F59: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F5A: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F5B: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F5C: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F5D: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F5E: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F5F: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
```

```
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F60: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F61: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F62: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F63: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F64: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F65: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F66: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F67: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F68: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F69: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F6A: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F6B: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F6C: 21        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
```

```
1F6D: 21       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F6E: 21       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F6F: FF       >70             db    $FF        ; Force overflow on undigits.
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F70: FF       >70             db    $FF        ; Force overflow on undigits.
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F71: FF       >70             db    $FF        ; Force overflow on undigits.
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F72: FF       >70             db    $FF        ; Force overflow on undigits.
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F73: FF       >70             db    $FF        ; Force overflow on undigits.
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F74: FF       >70             db    $FF        ; Force overflow on undigits.
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F75: 21       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F76: 22       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F77: 22       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F78: 22       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F79: 22       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1F7A: 22       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
```

```
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F7B: 22        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F7C: 22        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F7D: 22        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F7E: 22        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F7F: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F80: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F81: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F82: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F83: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F84: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F85: 22        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F86: 22        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F87: 22        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
```

```
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F88: 22        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F89: 22        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F8A: 22        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F8B: 22        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F8C: 22        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F8D: 22        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F8E: 22        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F8F: FF        >70           db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F90: FF        >70           db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F91: FF        >70           db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F92: FF        >70           db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F93: FF        >70           db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F94: FF        >70           db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F95: 22        >72           db    >]T*10+]U*6+MEM
```

```
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F96: 22        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F97: 22        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F98: 22        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F99: 22        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F9A: 22        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F9B: 22        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F9C: 22        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F9D: 22        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F9E: 22        >72            db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1F9F: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1FA0: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1FA1: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1FA2: FF        >70            db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
```

```
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1FA3: FF        >70           db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1FA4: FF        >70           db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1FA5: 22        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1FA6: 22        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1FA7: 22        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1FA8: 22        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1FA9: 22        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1FAA: 22        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1FAB: 22        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1FAC: 22        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1FAD: 22        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1FAE: 22        >72           db    >]T*10+]U*6+MEM
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1FAF: FF        >70           db    $FF        ; Force overflow on undigits.
                >65   ]Ax      equ   *-BCDLadrh ; ]Ax = index of table entry
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
```

```
1FB0: FF       >70             db    $FF        ; Force overflow on undigits.
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1FB1: FF       >70             db    $FF        ; Force overflow on undigits.
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1FB2: FF       >70             db    $FF        ; Force overflow on undigits.
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1FB3: FF       >70             db    $FF        ; Force overflow on undigits.
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1FB4: FF       >70             db    $FF        ; Force overflow on undigits.
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1FB5: 22       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1FB6: 22       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1FB7: 22       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1FB8: 22       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1FB9: 23       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1FBA: 23       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1FBB: 23       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1FBC: 23       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
               >66   ]T        equ   ]Ax/16     ; BCD tens digit
               >67   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U        equ   ]Ax-]A0    ; BCD units digit
1FBD: 23       >72             db    >]T*10+]U*6+MEM
               >65   ]Ax       equ   *-BCDLadrh ; ]Ax = index of table entry
```

```
                >66   ]T       equ   ]Ax/16     ; BCD tens digit
                >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1FBE: 23        >72            db    >]T*10+]U*6+MEM
                >75
                >76   BCDHadrl equ   *          ; BCD Hi 2 dig --> bin lo byte
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FBF: 00        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FC0: 58        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FC1: B0        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FC2: 08        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FC3: 60        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FC4: B8        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FC5: 10        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FC6: 68        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FC7: C0        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FC8: 18        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FC9: 00        >83            db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FCA: 00        >83            db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
```

```
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FCB: 00        >83            db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FCC: 00        >83            db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FCD: 00        >83            db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FCE: 00        >83            db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FCF: 70        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FD0: C8        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FD1: 20        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FD2: 78        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FD3: D0        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FD4: 28        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FD5: 80        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FD6: D8        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FD7: 30        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
```

```
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FD8: 88        >85           db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FD9: 00        >83           db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FDA: 00        >83           db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FDB: 00        >83           db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FDC: 00        >83           db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FDD: 00        >83           db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FDE: 00        >83           db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FDF: E0        >85           db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FE0: 38        >85           db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FE1: 90        >85           db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FE2: E8        >85           db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FE3: 40        >85           db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FE4: 98        >85           db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FE5: F0        >85           db    <]T*10+]U*600
```

```
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FE6: 48        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FE7: A0        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FE8: F8        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FE9: 00        >83            db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FEA: 00        >83            db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FEB: 00        >83            db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FEC: 00        >83            db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FED: 00        >83            db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FEE: 00        >83            db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FEF: 50        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FF0: A8        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FF1: 00        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FF2: 58        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
```

```
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FF3: B0        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FF4: 08        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FF5: 60        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FF6: B8        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FF7: 10        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FF8: 68        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FF9: 00        >83            db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FFA: 00        >83            db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FFB: 00        >83            db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FFC: 00        >83            db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FFD: 00        >83            db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FFE: 00        >83            db    0
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
1FFF: C0        >85            db    <]T*10+]U*600
                >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
                >79   ]T       equ   ]Ax/16     ; BCD tens digit
                >80   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >81   ]U       equ   ]Ax-]A0    ; BCD units digit
```

```
2000: 18       >85             db    <]T*10+]U*600
               >78   ]Ax       equ   *-BCDHadrl ; ]Ax = index of table entry
               >79   ]T        equ   ]Ax/16     ; BCD tens digit
               >80   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >81   ]U        equ   ]Ax-]A0    ; BCD units digit
2001: 70       >85             db    <]T*10+]U*600
               >78   ]Ax       equ   *-BCDHadrl ; ]Ax = index of table entry
               >79   ]T        equ   ]Ax/16     ; BCD tens digit
               >80   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >81   ]U        equ   ]Ax-]A0    ; BCD units digit
2002: C8       >85             db    <]T*10+]U*600
               >78   ]Ax       equ   *-BCDHadrl ; ]Ax = index of table entry
               >79   ]T        equ   ]Ax/16     ; BCD tens digit
               >80   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >81   ]U        equ   ]Ax-]A0    ; BCD units digit
2003: 20       >85             db    <]T*10+]U*600
               >78   ]Ax       equ   *-BCDHadrl ; ]Ax = index of table entry
               >79   ]T        equ   ]Ax/16     ; BCD tens digit
               >80   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >81   ]U        equ   ]Ax-]A0    ; BCD units digit
2004: 78       >85             db    <]T*10+]U*600
               >78   ]Ax       equ   *-BCDHadrl ; ]Ax = index of table entry
               >79   ]T        equ   ]Ax/16     ; BCD tens digit
               >80   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >81   ]U        equ   ]Ax-]A0    ; BCD units digit
2005: D0       >85             db    <]T*10+]U*600
               >78   ]Ax       equ   *-BCDHadrl ; ]Ax = index of table entry
               >79   ]T        equ   ]Ax/16     ; BCD tens digit
               >80   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >81   ]U        equ   ]Ax-]A0    ; BCD units digit
2006: 28       >85             db    <]T*10+]U*600
               >78   ]Ax       equ   *-BCDHadrl ; ]Ax = index of table entry
               >79   ]T        equ   ]Ax/16     ; BCD tens digit
               >80   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >81   ]U        equ   ]Ax-]A0    ; BCD units digit
2007: 80       >85             db    <]T*10+]U*600
               >78   ]Ax       equ   *-BCDHadrl ; ]Ax = index of table entry
               >79   ]T        equ   ]Ax/16     ; BCD tens digit
               >80   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >81   ]U        equ   ]Ax-]A0    ; BCD units digit
2008: D8       >85             db    <]T*10+]U*600
               >88
               >89   BCDHadrh  equ   *          ; BCD Hi 2 dig --> bin Hi byte
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
2009: 00       >98             db    >]T*10+]U*600
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
200A: 02       >98             db    >]T*10+]U*600
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
200B: 04       >98             db    >]T*10+]U*600
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
200C: 07       >98             db    >]T*10+]U*600
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
```

```
200D: 09       >98             db    >]T*10+]U*600
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
200E: 0B       >98             db    >]T*10+]U*600
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
200F: 0E       >98             db    >]T*10+]U*600
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
2010: 10       >98             db    >]T*10+]U*600
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
2011: 12       >98             db    >]T*10+]U*600
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
2012: 15       >98             db    >]T*10+]U*600
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
2013: FF       >96             db    $FF        ; Force overflow on undigits.
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
2014: FF       >96             db    $FF        ; Force overflow on undigits.
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
2015: FF       >96             db    $FF        ; Force overflow on undigits.
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
2016: FF       >96             db    $FF        ; Force overflow on undigits.
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
2017: FF       >96             db    $FF        ; Force overflow on undigits.
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
2018: FF       >96             db    $FF        ; Force overflow on undigits.
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
2019: 17       >98             db    >]T*10+]U*600
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
201A: 19       >98             db    >]T*10+]U*600
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
```

```
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
201B: 1C        >98            db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
201C: 1E        >98            db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
201D: 20        >98            db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
201E: 23        >98            db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
201F: 25        >98            db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2020: 27        >98            db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2021: 2A        >98            db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2022: 2C        >98            db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2023: FF        >96            db    $FF        ; Force overflow on undigits.
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2024: FF        >96            db    $FF        ; Force overflow on undigits.
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2025: FF        >96            db    $FF        ; Force overflow on undigits.
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2026: FF        >96            db    $FF        ; Force overflow on undigits.
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2027: FF        >96            db    $FF        ; Force overflow on undigits.
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
```

```
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2028: FF        >96           db    $FF         ; Force overflow on undigits.
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2029: 2E        >98           db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
202A: 31        >98           db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
202B: 33        >98           db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
202C: 35        >98           db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
202D: 38        >98           db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
202E: 3A        >98           db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
202F: 3C        >98           db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2030: 3F        >98           db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2031: 41        >98           db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2032: 43        >98           db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2033: FF        >96           db    $FF         ; Force overflow on undigits.
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2034: FF        >96           db    $FF         ; Force overflow on undigits.
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2035: FF        >96           db    $FF         ; Force overflow on undigits.
```

```
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2036: FF        >96            db    $FF        ; Force overflow on undigits.
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2037: FF        >96            db    $FF        ; Force overflow on undigits.
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2038: FF        >96            db    $FF        ; Force overflow on undigits.
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2039: 46        >98            db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
203A: 48        >98            db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
203B: 4B        >98            db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
203C: 4D        >98            db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
203D: 4F        >98            db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
203E: 52        >98            db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
203F: 54        >98            db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2040: 56        >98            db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2041: 59        >98            db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2042: 5B        >98            db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
```

```
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2043: FF        >96           db    $FF        ; Force overflow on undigits.
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2044: FF        >96           db    $FF        ; Force overflow on undigits.
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2045: FF        >96           db    $FF        ; Force overflow on undigits.
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2046: FF        >96           db    $FF        ; Force overflow on undigits.
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2047: FF        >96           db    $FF        ; Force overflow on undigits.
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2048: FF        >96           db    $FF        ; Force overflow on undigits.
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
2049: 5D        >98           db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
204A: 60        >98           db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
204B: 62        >98           db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
204C: 64        >98           db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
204D: 67        >98           db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
204E: 69        >98           db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
204F: 6B        >98           db    >]T*10+]U*600
                >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
                >92   ]T       equ   ]Ax/16     ; BCD tens digit
                >93   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
                >94   ]U       equ   ]Ax-]A0    ; BCD units digit
```

```
2050: 6E       >98             db    >]T*10+]U*600
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
2051: 70       >98             db    >]T*10+]U*600
               >91   ]Ax       equ   *-BCDHadrh ; ]Ax = index of table entry
               >92   ]T        equ   ]Ax/16     ; BCD tens digit
               >93   ]A0       equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >94   ]U        equ   ]Ax-]A0    ; BCD units digit
2052: 72       >98             db    >]T*10+]U*600
```

```
                        >102  ************************************************************
                        >103  *                                                          *
                        >104  *                       PUTPDCMD                           *
                        >105  *                                                          *
                        >106  * Append null-terminated string at (A,Y) onto IN,X.       *
                        >107  * Command is in hi-ASCII.                                  *
                        >108  *                                                          *
                        >109  * Advances X, destroys A and Y.                           *
                        >110  *                                                          *
                        >111  ************************************************************
                        >112
2053: 85 CC             >113  putpdcmd sta   ptr        ; Set up string pointer
2055: 84 CD             >114           sty   ptr+1
2057: A0 00             >115           ldy   #0
2059: B1 CC             >116  :cmdloop lda   (ptr),y    ; Append command string
205B: F0 07             >117           beq   :rts       ;  until null
205D: 9D 00 02          >118           sta   IN,x       ;   to keyboard buffer.
2060: E8                >119           inx              ; Bump pointers.
2061: C8                >120           iny
2062: D0 F5             >121           bne   :cmdloop   ; (always)
                        >122
2064: 60                >123  :rts     rts              ; Return...
                        >124
                        >125  ************************************************************
                        >126  *                                                          *
                        >127  *                       PDOSCMD                            *
                        >128  *                                                          *
                        >129  * Execute null-terminated ProDOS command at (A,Y)         *
                        >130  * Command is in hi-ASCII.                                  *
                        >131  *                                                          *
                        >132  * Keyboard buffer, sptr, and Y are changed.               *
                        >133  * On error, C is set and A contains error code.           *
                        >134  *                                                          *
                        >135  ************************************************************
                        >136
2065: A2 00             >137  pdoscmd  ldx   #0         ; Empty kbd buffer.
2067: 20 53 20          >138           jsr   putpdcmd   ; Move in the command
                        >139                            ;  and fall into pdosxeq.
                        >140
                        >141  ************************************************************
                        >142  *                                                          *
                        >143  *                       PDOSXEQ                            *
                        >144  *                                                          *
                        >145  * Execute ProDOS command in keyboard buffer after         *
                        >146  * appending a carriage return.  Command is in hi-ASCII.   *
                        >147  *                                                          *
                        >148  * On error, C is set and A contains error code.           *
                        >149  *                                                          *
                        >150  ************************************************************
                        >151
206A: A9 8D             >152  pdosxeq  lda   #$8D       ; Carriage Return
206C: 9D 00 02          >153           sta   IN,x       ;  at end
206F: AE 42 BE          >154           ldx   BSSTATE    ; Save BASIC.SYSTEM
2072: 86 DB             >155           stx   line8      ;  'state' var & set it
2074: A2 FF             >156           ldx   #$FF       ;    to suppress blank
2076: 8E 42 BE          >157           stx   BSSTATE    ;     line.
2079: 20 03 BE          >158           jsr   DOSCMD     ; Then do it...
207C: A6 DB             >159           ldx   line8      ; Restore BASIC.SYSTEM
207E: 8E 42 BE          >160           stx   BSSTATE    ;  state variable.
2081: 60                >161           rts
                        >162
                        >163  simend   equ   *-1        ; End of B220SIM code
                        >164           err   simend/MEM ; Can't encroach on MEM area.
```

```
                >166  * File name table can initially overlap B220 MEM
                >167
                >168  fnames   equ   $300       ; Ultimate location
                >169
2082: D0 D4 D2  >170  fnametbl asc   "PTRDR0",00 ; Moved to $300 by 'init'.
2089: 00 00 00  >171           ds    18
209B: D0 D4 D2  >172           asc   "PTRDR1",00
20A2: 00 00 00  >173           ds    18
20B4: D0 D4 D0  >174           asc   "PTPCH0",00
20BB: 00 00 00  >175           ds    18
20CD: D0 D4 D0  >176           asc   "PTPCH1",00
20D4: 00 00 00  >177           ds    18
20E6: CD D4 D5  >178           asc   "MTU0L0",00
20ED: 00 00 00  >179           ds    18
20FF: CD D4 D5  >180           asc   "MTU0L1",00
2106: 00 00 00  >181           ds    18
2118: CD D4 D5  >182           asc   "MTU1L0",00
211F: 00 00 00  >183           ds    18
2131: CD D4 D5  >184           asc   "MTU1L1",00
2138: 00 00 00  >185           ds    18
                >186  fnend    equ   *


--End assembly, 6474 bytes, Errors: 0


Symbol table - alphabetical order:

   ADCYop  =$79        ADCZop  =$65        ADD     =$1383      ADDR    =$04
   ADDRerr =$0C40      ADDRerrR=$0B48      ADL     =$13F3      ALTCHAR =$C00F
   AR1     =$0700      AR2     =$0680      AR4     =$0600      AR8     =$0580
   ARbord  =$0DD4      ARmid   =$0DFA      ARv     =$0428      Aattr   =$0C9A
   Acol    =$05        Ain     =$091C      Alab    =$0583      Aparm   =$1259
?  B220SIM =$0800      B220col =$0C        B220end =$C7        B220msg =$0DBF
   B220strt=$90        BASCALC =$FBC1      BASL    =$28        BCDHadrh=$2009
   BCDHadrl=$1FBF      BCDLadrh=$1F25      BCDLadrl=$1E8B      BCE     =$19F3
   BCH     =$19DF      BCS     =$1A6B      BCSop   =$B0        BEEP    =$FBDD
   BFA     =$1A1C      BFR     =$1A18      BITZop  =$24        BNEop   =$D0
   BOF     =$19C2      BPC1    =$0728      BPC2    =$06A8      BPC4    =$0628
   BPC8    =$05A8      BPCbord =$0E20      BPCmid  =$0E46      BPCv    =$0450
   BRP     =$19CF      BSA     =$19D5      BSSTATE =$BE42      BUN     =$1A05
   Battr   =$0CCA      Bcol    =$05        Bin     =$0920      Blab    =$05AB
   Bmodmem =$118B      Bparm   =$1261      Bxxxx   =$1252      CAD     =$1364
   CFA     =$156D      CH      =$24        CLA     =$1B43      CLCop   =$18
   CLL     =$1B64      CMPIop  =$C9        COMP    =$C2        COMPcol =$19
   COUT    =$FDED      CROUT   =$FD8E      CSU     =$137B      CSW     =$B6
   Cattr   =$0CEA      Ccol    =$15        Cin     =$0924      Clab    =$05BB
   DBB     =$19AF      DFL     =$187A      DIV     =$14A6      DLB     =$188A
   DOSCMD  =$BE03      DOSCON  =$03D0      ERR     =$C1        ERRcol  =$15
   ERRlab  =$0567      EXP     =$01        EXT     =$1545      FAD     =$1642
   FDV     =$17C3      FIELDerr=$0C3C      FMU     =$1731      FSU     =$162D
   HLT     =$110D      HOME    =$FC58      Help1   =$0E93   ?  Help2   =$0EB8
?  Help3   =$0EDE   ?  Help4   =$0F01      IBB     =$199C      IFL     =$1834
   IN      =$0200      INDshow =$0FC1      IOerr   =$0C44      IOstate =$1DDD
   IOstend =$1DF1      KAD     =$12DD      KBD     =$C000      KBSTROBE=$C010
   LDB     =$1AFF      LDR     =$1AF3      LSA     =$1B25      Lparm   =$125D
   MANT    =$02        MEM     =$20D0      MIB     =$1CEA      MIR     =$1C90
   MIW     =$1C88      MOR     =$1C9D      MOW     =$1C93      MPF     =$1CA0
   MRD     =$1C7A      MRR     =$1C85      MTC     =$1C77      MTS     =$1C45
   MUL     =$1425      NN      =$D1        NOP     =$110D      NOPop   =$EA
   OFLcol  =$1F        OFLerr  =$0C38      OP      =$03        OPerr   =$0C34
   Ov      =$C3        OvHlt   =$C7        PRB     =$1116      PRBL2   =$F94A
   PRD     =$1110      PRI     =$11AD   ?  PRINTERR=$BE0C      PWI     =$12DD
   PWR     =$11B0      Pattr   =$0CDA      Pcol    =$0D        Pin     =$0928
   Plab    =$05B3   ?  RESTART =$0803      RND     =$1523      RPTcol  =$22
   RTF     =$1936      RUN     =$C0        RUNcol  =$11        Rattr   =$0CB2
   Rcol    =$17        Rin     =$092C      Rlab    =$0595      Rp      =$C4
```

```
   S       =$00        SBCYop  =$F9        SBCZop  =$E5        SECop   =$38
   SLA     =$1BD7      SOR     =$1A78      SPKR    =$C030      SPO     =$12E0
   SRA     =$1B6F      STA     =$1A8C      STAT    =$0E6C      STATlin =$0550
   STP     =$1B2E      SUB     =$140F      SW1col  =$06        SWlab   =$0553
?  TABV    =$FB5B      UNDIGerR=$0B4B      UNDIGerr=$0C4A      VV      =$02
   WNDTOP  =$22     V  ]A0     =$40     V  ]Ax     =$49     V? ]Ov     =$192B
V  ]T      =$04     V  ]U      =$09     V? ]adc    =$18E2   V  ]add    =$1395
V? ]bfr    =$1A1E   V? ]clc    =$18D3   V? ]cmp    =$18E4   V? ]contin =$0BC1
V? ]dfl    =$189F   V  ]err    =$0C4C   V  ]errpt  =$1887   V? ]fad    =$1650
V  ]fetch1 =$1B40   V  ]fetch2 =$1AD6   V? ]fetch3 =$1A15   V  ]fetch4 =$1877
V? ]nop    =$190C   V? ]prd    =$112A   V  ]stop   =$080D   V? ]sub    =$190F
   b220asc =$1E01      bcdcor  =$1DD3      beepget =$0954      blanklin=$0D8D
   blkcnt  =$D7        changed =$D9        clear   =$1DA8      clearAR =$17B8
   cmdfnx  =$D8        compare =$158F      cursor  =$57        delete  =$FF
   disARmid=$0D95      disBPCbo=$0DA3      disBPCmi=$0DB1      disiocfg=$0A1E
   dispA   =$0F3A      dispB   =$0F48      dispC   =$0F56      dispP   =$0F4F
   dispR   =$0F41      dispSTAT=$0F5D      dispcnt =$64        dispctr =$D2
   dispdig =$1028      disphelp=$0F17      display =$0F28      disppanl=$0D04
   dispreg =$0FEA      divide  =$14AC      dnarrow =$8A        doio    =$11F2
   ediocfg =$0A15      escape  =$9B        exchAR  =$1D86      execute =$0B93
   fetch   =$0B72      fnamecol=$0C        fnames  =$0300      fnametbl=$2082
   fnend   =$214A      fnx     =$D3        fnxfn   =$1DF9      fnxoff  =$1DF1
   getMTt1 =$1265      getdig  =$0957      getfnx  =$11EC      getfnxt1=$1269
   incP    =$0C14      incmem  =$11A1      incoff  =$12C4      init    =$0C57
   inptr   =$CE        instptr =$C8        intabl  =$091C      inverse =$0B3B
   iocfgstr=$096F      iocfgtt =$0B        keyin   =$0806      keyinR  =$0B4E
   line    =$D8        line1   =$D5        line2   =$D7        line4   =$D9
   line8   =$DB        linev   =$D3        load    =$124A      ltarrow =$88
   memb    =$7530      memptr  =$CA        midNN   =$1DB5      mtlane  =$1DEF
?  mtoff   =$1DE9      mtrw    =$1CFB      mtwrite =$1CF6      multiply=$142B
   newP    =$0B54      newp    =$C5        noAD    =$8000      off     =$A0
   on      =$AA        operr   =$8C34      optabh  =$10B3      optabl  =$1059
?  pchoff  =$1DE3   ?  pdoscmd =$2065      pdosxeq =$206A      ptr     =$CC
   ptrdwrt =$11C8      ptread  =$11BB      ptwrite =$11C3      putbyte =$1294
   puthx   =$1290      putoff  =$12AC      putpdcmd=$2053      rA      =$9E
   rB      =$94        rBx     =$90        rC      =$98        rD      =$AA
   rD10    =$B0        rP      =$96        rR      =$A4        rdroff  =$1DDD
   reset   =$0C7C   MD resi    =$8000      restart =$0C91      rtmargin=$04
   sL      =$01        save    =$124E      savex   =$D6        selBASL =$DB
   selch   =$D7        selected=$D4        selsave =$D5     MD seti    =$8000
   setread =$11D0      setwrite=$11DF      shleft1 =$0930      signtbl =$155D
   simend  =$2081      skipincP=$C6        slA     =$1D7B      slT     =$1D71
   splitsL =$1D94      srA     =$1D42      srAM    =$1D44      srAMR   =$1D4F
   srAS    =$1D40   ?  srR     =$1D52      srT     =$1D4D      srT2    =$1D5D
   stopR   =$0B51      t1      =$D0        tabs    =$135F      uparrow =$8B
   zeroff  =$D5
```

Symbol table - numerical order:

```
   S       =$00        sL      =$01        EXP     =$01        VV      =$02
   MANT    =$02        OP      =$03        ADDR    =$04        rtmargin=$04
V  ]T      =$04        Acol    =$05        Bcol    =$05        SW1col  =$06
V  ]U      =$09        iocfgtt =$0B        fnamecol=$0C        B220col =$0C
   Pcol    =$0D        RUNcol  =$11        Ccol    =$15        ERRcol  =$15
   Rcol    =$17        CLCop   =$18        COMPcol =$19        OFLcol  =$1F
   WNDTOP  =$22        RPTcol  =$22        BITZop  =$24        CH      =$24
   BASL    =$28        SECop   =$38     V  ]A0     =$40     V  ]Ax     =$49
   cursor  =$57        dispcnt =$64        ADCZop  =$65        ADCYop  =$79
   ltarrow =$88        dnarrow =$8A        uparrow =$8B        B220strt=$90
   rBx     =$90        rB      =$94        rP      =$96        rC      =$98
   escape  =$9B        rA      =$9E        off     =$A0        rR      =$A4
   rD      =$AA        on      =$AA        BCSop   =$B0        rD10    =$B0
   CSW     =$B6        RUN     =$C0        ERR     =$C1        COMP    =$C2
   Ov      =$C3        Rp      =$C4        newp    =$C5        skipincP=$C6
   B220end =$C7        OvHlt   =$C7        instptr =$C8        CMPIop  =$C9
   memptr  =$CA        ptr     =$CC        inptr   =$CE        BNEop   =$D0
```

```
   t1      =$D0        NN      =$D1        dispctr =$D2        linev   =$D3
   fnx     =$D3        selected=$D4        line1   =$D5        selsave =$D5
   zeroff  =$D5        savex   =$D6        line2   =$D7        selch   =$D7
   blkcnt  =$D7        line    =$D8        cmdfnx  =$D8        line4   =$D9
   changed =$D9        line8   =$DB        selBASL =$DB        SBCZop  =$E5
   NOPop   =$EA        SBCYop  =$F9        delete  =$FF        IN      =$0200
   fnames  =$0300      DOSCON  =$03D0      ARv     =$0428      BPCv    =$0450
   STATlin =$0550      SWlab   =$0553      ERRlab  =$0567      AR8     =$0580
   Alab    =$0583      Rlab    =$0595      BPC8    =$05A8      Blab    =$05AB
   Plab    =$05B3      Clab    =$05BB      AR4     =$0600      BPC4    =$0628
   AR2     =$0680      BPC2    =$06A8      AR1     =$0700      BPC1    =$0728
?  B220SIM =$0800   ?  RESTART =$0803      keyin   =$0806   V  ]stop   =$080D
   intabl  =$091C      Ain     =$091C      Bin     =$0920      Cin     =$0924
   Pin     =$0928      Rin     =$092C      shleft1 =$0930      beepget =$0954
   getdig  =$0957      iocfgstr=$096F      ediocfg =$0A15      disiocfg=$0A1E
   inverse =$0B3B      ADDRerrR=$0B48      UNDIGerR=$0B4B      keyinR  =$0B4E
   stopR   =$0B51      newP    =$0B54      fetch   =$0B72      execute =$0B93
V? ]contin =$0BC1      incP    =$0C14      OPerr   =$0C34      OFLerr  =$0C38
   FIELDerr=$0C3C      ADDRerr =$0C40      IOerr   =$0C44      UNDIGerr=$0C4A
V  ]err    =$0C4C      init    =$0C57      reset   =$0C7C      restart =$0C91
   Aattr   =$0C9A      Rattr   =$0CB2      Battr   =$0CCA      Pattr   =$0CDA
   Cattr   =$0CEA      disppanl=$0D04      blanklin=$0D8D      disARmid=$0D95
   disBPCbo=$0DA3      disBPCmi=$0DB1      B220msg =$0DBF      ARbord  =$0DD4
   ARmid   =$0DFA      BPCbord =$0E20      BPCmid  =$0E46      STAT    =$0E6C
   Help1   =$0E93   ?  Help2   =$0EB8   ?  Help3   =$0EDE   ?  Help4   =$0F01
   disphelp=$0F17      display =$0F28      dispA   =$0F3A      dispR   =$0F41
   dispB   =$0F48      dispP   =$0F4F      dispC   =$0F56      dispSTAT=$0F5D
   INDshow =$0FC1      dispreg =$0FEA      dispdig =$1028      optabl  =$1059
   optabh  =$10B3      HLT     =$110D      NOP     =$110D      PRD     =$1110
   PRB     =$1116   V? ]prd    =$112A      Bmodmem =$118B      incmem  =$11A1
   PRI     =$11AD      PWR     =$11B0      ptread  =$11BB      ptwrite =$11C3
   ptrdwrt =$11C8      setread =$11D0      setwrite=$11DF      getfnx  =$11EC
   doio    =$11F2      load    =$124A      save    =$124E      Bxxxx   =$1252
   Aparm   =$1259      Lparm   =$125D      Bparm   =$1261      getMTt1 =$1265
   getfnxt1=$1269      puthx   =$1290      putbyte =$1294      putoff  =$12AC
   incoff  =$12C4      PWI     =$12DD      KAD     =$12DD      SPO     =$12E0
   tabs    =$135F      CAD     =$1364      CSU     =$137B      ADD     =$1383
V  ]add    =$1395      ADL     =$13F3      SUB     =$140F      MUL     =$1425
   multiply=$142B      DIV     =$14A6      divide  =$14AC      RND     =$1523
   EXT     =$1545      signtbl =$155D      CFA     =$156D      compare =$158F
   FSU     =$162D      FAD     =$1642   V? ]fad    =$1650      FMU     =$1731
   clearAR =$17B8      FDV     =$17C3      IFL     =$1834   V  ]fetch4 =$1877
   DFL     =$187A   V  ]errpt  =$1887      DLB     =$188A   V? ]dfl    =$189F
V? ]clc    =$18D3   V? ]adc    =$18E2   V? ]cmp    =$18E4   V? ]nop    =$190C
V? ]sub    =$190F   V? ]Ov     =$192B      RTF     =$1936      IBB     =$199C
   DBB     =$19AF      BOF     =$19C2      BRP     =$19CF      BSA     =$19D5
   BCH     =$19DF      BCE     =$19F3      BUN     =$1A05   V? ]fetch3 =$1A15
   BFR     =$1A18      BFA     =$1A1C   V? ]bfr    =$1A1E      BCS     =$1A6B
   SOR     =$1A78      STA     =$1A8C   V  ]fetch2 =$1AD6      LDR     =$1AF3
   LDB     =$1AFF      LSA     =$1B25      STP     =$1B2E   V  ]fetch1 =$1B40
   CLA     =$1B43      CLL     =$1B64      SRA     =$1B6F      SLA     =$1BD7
   MTS     =$1C45      MTC     =$1C77      MRD     =$1C7A      MRR     =$1C85
   MIW     =$1C88      MIR     =$1C90      MOW     =$1C93      MOR     =$1C9D
   MPF     =$1CA0      MIB     =$1CEA      mtwrite =$1CF6      mtrw    =$1CFB
   srAS    =$1D40      srA     =$1D42      srAM    =$1D44      srT     =$1D4D
   srAMR   =$1D4F   ?  srR     =$1D52      srT2    =$1D5D      slT     =$1D71
   slA     =$1D7B      exchAR  =$1D86      splitsL =$1D94      clear   =$1DA8
   midNN   =$1DB5      bcdcor  =$1DD3      IOstate =$1DDD      rdroff  =$1DDD
?  pchoff  =$1DE3   ?  mtoff   =$1DE9      mtlane  =$1DEF      IOstend =$1DF1
   fnxoff  =$1DF1      fnxfn   =$1DF9      b220asc =$1E01      BCDLadrl=$1E8B
   BCDLadrh=$1F25      BCDHadrl=$1FBF      BCDHadrh=$2009      putpdcmd=$2053
?  pdoscmd =$2065      pdosxeq =$206A      simend  =$2081      fnametbl=$2082
   MEM     =$20D0      fnend   =$214A      memb    =$7530      noAD    =$8000
   operr   =$8C34   MD resi    =$8000   MD seti    =$8000      DOSCMD  =$BE03
?  PRINTERR=$BE0C      BSSTATE =$BE42      KBD     =$C000      ALTCHAR =$C00F
   KBSTROBE=$C010      SPKR    =$C030      PRBL2   =$F94A   ?  TABV    =$FB5B
   BASCALC =$FBC1      BEEP    =$FBDD      HOME    =$FC58      CROUT   =$FD8E
```

```
COUT    =$FDED
```