```
**********************************************************
*                                                        *
*                     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.      *
*                                                        *
**********************************************************
```

```

         put   B220HISTORY
**********************************************************
*                                                        *
*                       History                          *
*                                                        *
* 03/29/16 - Ran first B220 op--HLT!  BCD address to MEM *
*            address is OK.                              *
*                                                        *
* 03/31/16 - Began implementing B220 front panel display *
*            in 40-column text mode.                     *
*                                                        *
* 04/02/16 - Front panel complete, adding keyboard cntl. *
*                                                        *
* 04/05/16 - Keyboard control complete, adding opcodes.  *
*                                                        *
* 04/11/16 - Refined error handling. Added B220CODE file *
*            loading. Implemented partial field STA/R/B. *
*                                                        *
* 04/12/16 - Added conditional branches, STx, LDR, LDB,  *
*            LSA, CLx, CLL, SRx, IBB, DBB.               *
*            Revised manual (keyboard) control.          *
*                                                        *
* 04/13/16 - Added non-BCD digit checking for addresses. *
*            Improved macros for B220 code assembly.     *
*            Split source into small 'put' files.        *
*                                                        *
* 04/15/16 - Added SLx and tested all shifts.            *
*                                                        *
* 04/18/16 - Added ADD and SUB and variants.             *
*                                                        *
* 04/19/16 - Added ADL, tested ADD, ADA, SUB, SUA, ADL.  *
*                                                        *
* 04/22/16 - Added simple MUL and a faster, byte-shifting*
*            version (currently FMU).                    *
*                                                        *
* 04/26/16 - Added EXT and RND. Added special cases for  *
*            SRT 10 and  SLT 10.                         *
*                                                        *
* 04/27/16 - Added simple version of DIV.                *
*                                                        *
* 04/29/16 - Added CFA, CFR.                             *
*                                                        *
* 05/02/16 - Added BFA, BFR. Made 'compare' subroutine.  *
*                                                        *
* 05/04/16 - Added RTF, DFL, and DLB.  Split B220EXEC.   *
*                                                        *
* 05/09/16 - Added help redisplay. Paginated EXEC1 & 2.  *
*                                                        *
* 05/12/16 - Moved HLT execution to 'fetch'. Looks good! *
*                                                        *
* 05/15/16 - Fixed bug in 'compare'.  Added simple SPO.  *
*                                                        *
* 05/16/16 - Added Z reset command, revised help.        *
*                                                        *
* 05/18/16 - Added PWR command; first disk command.      *
*                                                        *
* 06/02/16 - Added PRD, PRB commands, removed B220CODE   *
*            pre-load hack.                              *
*                                                        *
* 06/07/16 - Moved FP ops to B220EXEC2. Changed Quit to  *
*            go to full text window and reconnect ProDOS.*
*                                                        *
* 06/19/16 - Fixed STR/STB partial field bug.            *
*                                                        *
* 06/24/16 - Changed PWR to truncate preexisting file.   *
*                                                        *
```

===== Page 3 =====

```
* 07/01/16 - Added FAD, FSU.                             *
*                                                        *
* 07/21/16 - Added FMU.                                  *
*                                                        *
* 07/25/16 - Many small JMP --> Bxx space optimizations. *
*            RTF now moves upward! Generalized 'clear'.  *
*                                                        *
* 07/28/16 - Added FDV. Organized shift subroutines.     *
*                                                        *
* 08/22/16 - Modified 'b220asc' table for ) and %.       *
*                                                        *
* 08/27/16 - Fixed LBC bug--hi byte was high by one.     *
*            Fixed SPO: +, form feed, and 'ignore'.      *
*                                                        *
* 09/01/16 - Implemented B220 "tab" in SPO.              *
*                                                        *
* 09/02/16 - Fixed RTF: rB now incremented when NN = 00. *
*                                                        *
* 09/03/16 - Fixed BCH. Was branching on equal.          *
*                                                        *
* 09/05/16 - Fixed IFL, DFL, DLB: if s odd, zeroed s+1.  *
*                                                        *
* 09/09/16 - Added SOR/SOH op and subset of Mag Tape ops.*
*                                                        *
* 09/10/16 - Split PTUNITn into PTRDRn and PTPCHn.       *
*                                                        *
* 09/11/16 - Combined paper tape and mag tape I/O.       *
*                                                        *
* 09/16/16 - Added MRD B-modification.                   *
*                                                        *
* 09/20/16 - Added MPE as NOP.                           *
*                                                        *
* 09/21/16 - Added MLS for SNAP 1E.                      *
*                                                        *
* 09/23/16 - Added IOM (Interrogate Overflow Mode).      *
*                                                        *
* 09/24/16 - Fixed IFL bug: No Ov if hi field posn even. *
*                                                        *
* 11/12/16 - Several small cleanups. ** RELEASED v1.0 ** *
*                                                        *
* 01/16/17 - Moved MEM to top in prep for IOCFG addition.*
*                                                        *
* 01/17/17 - Added I/O configuration editor.             *
*            Restricted PTRDR and PTPCH units to 0 and 1.*
*                                                        *
* 01/25/17 - Integrated I/O Config Editor into B220SIM.  *
*            Fixed MPB bug.                              *
*                                                        *
* 02/01/17 - Added "v1.1" and I/O Config help line.      *
*            ** RELEASED v1.1 **                         *
*                                                        *
* 04/27/17 - Added 'skipincP' to skip P reg increment if *
*            PRB sign 6/7 instruction executed.          *
*                                                        *
* 05/01/17 - Char code matched to CCONV: 04 = ), 10 = (, *
*            27 = $, 32 = ?, 34 = '                      *
*                                                        *
* 06/27/17 - Fixed bug in 'divide', now RTS on overflow. *
*                                                        *
* 08/09/20 - Fixed align & normalization bugs in FAD/FSU.*
*            Fixed post-normalization bug in FDV.        *
*            Kluged KAD as a HLT for rA modification.    *
*            Added "Quit to BASIC" to help lines.        *
*            Cleaned up SUB code.                        *
*                                                        *
* 08/11/20 - Fixed sign logic bugs in CAD/CAA/CSU/CSA.   *
*                                                        *
```

```
               ===== Page 4 =====

* 08/12/20 - Fixed rotate bugs in SLA/SLT/SLS.           *
*                                                        *
* 08/13/20 - Preserve rR sign in FDV.                    *
*            Always clear rR in RND.                     *
*            Force rA sign to 0 or 1 in ADL.             *
*            Post-normalize in FAD.                      *
*            Fix FAD result on exponent overflow.        *
*                                                        *
* 08/14/20 - Fix FMU overflow exit state.                *
*                                                        *
* 08/15/20 - Clear rA sign before ovflow check in DIV.   *
*            Clear rA sign if ovflow in FDV.             *
*                                                        *
* 08/16/20 - Force normal zero result in FAD/FSU.        *
*                                                        *
* 08/23/20 - Rewrote SRx to save code!                   *
*                                                        *
* 08/24/20 - Detect EXP Ovflo before mant srT2 in FDV.   *
*            Clear rA EXP on exponent overflow in FDV,   *
*             except when it occurs in ':shrT2'.         *
*            Carry out of mantissa in FAD is not zero.   *
*                                                        *
* 09/01/20 - Changed 'B220msg' to v1.2.                  *
*            Made B220HISTORY a separate PUT file.       *
*            ** RELEASED v1.2 **                         *
*                                                        *
**********************************************************
```

```
         use   B220DEFS
* 6502 equates

BCSop    equ   $B0        ; BCS opcode
BNEop    equ   $D0        ; BNE opcode
CLCop    equ   $18        ; CLC opcode
SECop    equ   $38        ; SEC opcode
NOPop    equ   $EA        ; NOP opcode
ADCZop   equ   $65        ; ADC zp opcode
BITZop   equ   $24        ; BIT zp opcode
CMPIop   equ   $C9        ; CMP # opcode
SBCZop   equ   $E5        ; SBC zp opcode
ADCYop   equ   $79        ; ADC aaaa,y opcode
SBCYop   equ   $F9        ; SBC aaaa,y opcode

* Apple equates

WNDTOP   equ   $22        ; Top line of text window
CH       equ   $24        ; COUT horizontal cursor
BASL     equ   $28        ; Screen base address
IN       equ   $200       ; Keyboard input buffer
KBD      equ   $C000      ; Keyboard port
ALTCHAR  equ   $C00F      ; Store to enable alt charset
KBSTROBE equ   $C010      ; Keyboard strobe reset
SPKR     equ   $C030      ; Toggle speaker

* Apple entry points

DOSCON   equ   $3D0       ; ProDOS reconnect vector
DOSCMD   equ   $BE03      ; BASIC.SYSTEM PDOS command
PRINTERR equ   $BE0C      ; Print ProDOS error msg
BSSTATE  equ   $BE42      ; BASIC.SYSTEM state var
PRBL2    equ   $F94A      ; Print (X) blanks
TABV     equ   $FB5B      ; Vertical tab to (A)
BASCALC  equ   $FBC1      ; Set BASL to line (A)
BEEP     equ   $FBDD      ; Beep
HOME     equ   $FC58      ; Clear screen
CROUT    equ   $FD8E      ; Output a CR
COUT     equ   $FDED      ; Output char in A

* Simulation parameters

memb     equ   5000*6     ; 5000 6-byte B220 words
MEM      equ   $9600-memb ; Simulated B220 memory
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
```

===== Page 7 =====

```
**********************************************************
*                                                        *
*                  Macro Definitions                     *
*                                                        *
**********************************************************

seti     mac              ; Set indicator
         lda   #$FF
         sta   ]1         ; Set non-zero.
         eom

resi     mac              ; Reset indicator
         lda   #0
         sta   ]1         ; Zero indicator.
         eom

         org   $800
         dsk   /ap/merlin/work/b220/b220sim
```

```
                ********************************************************** 
                *                                                        *
                *                    Entry Point                         *
                *                                                        *
                **********************************************************

0800: 4C 57 0C  B220SIM  jmp   init       ; Start simulator.
0803: 4C 91 0C  RESTART  jmp   restart    ; Restart warm.
```

```
                59             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 33 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 22 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 33 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
```

```
                **********************************************************
                *                                                        *
                *         Shift Register left 1 digit (4 bits)           *
                *                                                        *
                * Y = addr of Hi (left) byte of reg, X = byte length - 1 *
                * X and Y are unchanged on exit.                         *
                * High digit of sign byte of rA, rR, and rC is cleared.  *
                *                                                        *
                **********************************************************

0930: 8C 38 09  shleft1  sty   :shift+1   ; Save register address
0933: 8A                 txa              ;  and byte length - 1.
0934: A0 04              ldy   #4         ; Digit = 4 bits.
0936: 18        :nxshift clc              ; Shift in zeroes.
0937: 36 00     :shift   rol   0*0,x      ; Shift 1 bit
0939: CA                 dex              ;  for all bytes.
093A: 10 FB              bpl   :shift
093C: AA                 tax              ; Restore X
093D: 88                 dey
093E: D0 F6              bne   :nxshift   ; Shift 4 times.
0940: AC 38 09           ldy   :shift+1   ; Restore Y = reg address.
0943: C0 96              cpy   #rP        ; rP has no sign byte,
0945: F0 0C              beq   :rts       ;  so skip it.
0947: C0 94              cpy   #rB        ; rB has no sign byte,
0949: F0 08              beq   :rts       ;  so skip it.
094B: B9 00 00           lda   0,y        ; Clear high digit
094E: 29 0F              and   #$0F       ;  of sign byte.
0950: 99 00 00           sta   0,y
0953: 60        :rts     rts

                **********************************************************
                *                                                        *
                *                   Get Digit or CR                      *
                *                                                        *
                * On exit: If C = 0, A = digit value                     *
                *          If C = 1, CR received                         *
                *          X and Y unchanged.                            *
                *                                                        *
                **********************************************************

0954: 20 DD FB  beepget  jsr   BEEP       ; Signal error key
0957: AD 00 C0  getdig   lda   KBD        ; Get digit or <Enter>
095A: 10 FB              bpl   getdig
095C: 8D 10 C0           sta   KBSTROBE   ; Clear strobe
095F: C9 8D              cmp   #$8D       ; <Enter>?
0961: F0 0B              beq   :done      ; Yes, exit.
0963: C9 B0              cmp   #"0"       ; -No, less than "0"?
0965: 90 ED              bcc   beepget    ; -Yes, get another.
0967: C9 BA              cmp   #"9"+1     ; -No, greater than "9"?
0969: B0 E9              bcs   beepget    ; -Yes, get another.
096B: 29 0F              and   #$0F       ; -No, isolate digit
096D: 18                 clc              ; C = 0 for digit
096E: 60        :done    rts              ; C = 1 for CR.
```

```
               **********************************************************
               *                                                        *
               *            Edit B220SIM I/O Configuration             *
               *                                                        *
               **********************************************************

               cursor   equ   $57        ; Mousetext checkerboard
               uparrow  equ   $8B        ; Up arrow
               dnarrow  equ   $8A        ; Down arrow
               ltarrow  equ   $88        ; Left arrow
               escape   equ   $9B        ; ESCAPE key
               delete   equ   $FF        ; DELETE key
               iocfgtt  equ   11         ; HTAB for screen title
               rtmargin equ   4          ; Right margin
               fnamecol equ   rtmargin+8 ; File name column

               fnx      equ   linev      ; File name index (0..7)
               selected equ   linev+1    ; Selected index (0..7)
               selsave  equ   line1      ; Temp Y storage
               savex    equ   line1+1    ; Temp X storage
               selch    equ   line2      ; Selected fname cursor
               line     equ   line2+1    ; Line number (0..23)
               changed  equ   line4      ; File name changed flg
               selBASL  equ   line8      ; Selected line base (DA.DB)

               iocfgstr equ   *          ; I/O Config Screen string
096F: C9 AF CF          asc   "I/O Configuration",0D
0981: 0D                db    $0D
0982: A0 D5 EE          asc   " Unit    File pathname",0D
0999: AD AD AD          asc   "------  ------------------------",0D
09BA: D0 D4 D2          asc   "PTRDR0",01
09C1: D0 D4 D2          asc   "PTRDR1",01
09C8: D0 D4 D0          asc   "PTPCH0",01
09CF: D0 D4 D0          asc   "PTPCH1",01
09D6: 0D                db    $0D
09D7: CD D4 D5          asc   "MTU0L0",01
09DE: CD D4 D5          asc   "MTU0L1",01
09E5: CD D4 D5          asc   "MTU1L0",01
09EC: CD D4 D5          asc   "MTU1L1",01
09F3: 0D 0D 0D          db    $0D,$0D,$0D,$0D,$0D
09F8: A0 A0 A0          asc   "    ESC to return to B220SIM"
0A14: 00                db    00         ; End of screen

0A15: A2 00    ediocfg  ldx   #0         ; Edit I/O Configuration
0A17: 86 22             stx   WNDTOP     ; Set full screen.
0A19: 86 D4             stx   selected   ; Select first file name.
0A1B: 20 58 FC          jsr   HOME       ; Clear screen
0A1E: A2 00    disiocfg ldx   #0         ; iocfgstr index = 0
0A20: 86 D3             stx   fnx        ; fname index = 0
0A22: 86 D8             stx   line       ; Line = 0
0A24: 8A                txa
0A25: 20 C1 FB          jsr   BASCALC    ; Set BASL for line 0
0A28: A0 0B             ldy   #iocfgtt   ; HTAB to title
0A2A: BD 6F 09 :nxch    lda   iocfgstr,x ; Next disp string char
0A2D: 10 06             bpl   :command   ; -Command char if +
0A2F: 91 28             sta   (BASL),y   ; -Display if not cmd.
0A31: C8                iny              ; Advance CH
0A32: E8       :advance inx              ; Advance str index
0A33: D0 F5             bne   :nxch      ; (always)

0A35: F0 48    :command beq   :editfn    ; Screen complete, edit.
0A37: C9 0D             cmp   #$0D       ; CR?
0A39: D0 0B             bne   :fname     ; -No, insert file name.
0A3B: E6 D8    :nxtline inc   line       ; -Yes, next line.
0A3D: A5 D8             lda   line       ; Compute new line's
0A3F: 20 C1 FB          jsr   BASCALC    ;  base addr (BASL)
0A42: A0 04             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 21 1E >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 21 1E >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 19 1E >353           ldx   fnxoff,y   ;  for new file.
0ACD: 9D 05 1E >354           sta   rdroff,x
0AD0: 9D 06 1E >355           sta   rdroff+1,x
0AD3: 9D 07 1E >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
```

```
                              ===== Page 17 =====

                60            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 B3 1E >20            lda   BCDLadrl,x ; -No, compute 'instptr'
0B63: 79 E7 1F >21            adc   BCDHadrl,y
0B66: 85 C8    >22            sta   instptr    ; Low byte of instr address
0B68: BD 4D 1F >23            lda   BCDLadrh,x
0B6B: 79 31 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 33 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 64 10 >80  :skip     lda   optabl,y   ; Get execute address.
0BDB: 8D 0B 0C >81            sta   :go+1
0BDE: B9 BE 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 B3 1E >91            lda   BCDLadrl,x ; -No, compute 'memptr'
0BF5: 79 E7 1F >92            adc   BCDHadrl,y
0BF8: 85 CA    >93            sta   memptr     ; Low byte of memory address
0BFA: BD 4D 1F >94            lda   BCDLadrh,x
0BFD: 79 31 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
```

```
               * B220 error routines

0C34: A9 CF    OPerr    lda   #"O"       ; OPcode error
0C36: D0 14             bne   ]err       ; (always)

0C38: A9 D6    OFLerr   lda   #"V"       ; Overflow error
0C3A: D0 10             bne   ]err       ; (always)

0C3C: A9 C6    FIELDerr lda   #"F"       ; Field error
0C3E: D0 0C             bne   ]err       ; (always)

0C40: A9 C1    ADDRerr  lda   #"A"       ; Address error
0C42: D0 08             bne   ]err       ; (always)

0C44: 85 00    IOerr    sta   0          ; Save I/O err code
0C46: A9 C9             lda   #"I"       ; I/O error
0C48: D0 02             bne   ]err

0C4A: A9 D8    UNDIGerr lda   #"X"       ; Non-BCD digit error
0C4C: 8D 67 05 ]err     sta   ERRlab     ; Show on screen.
0C4F: 85 C1             sta   ERR        ; Set error indicator,
0C51: 20 DD FB          jsr   BEEP       ;  sound beep,
0C54: 4C 0D 08          jmp   ]stop      ;   and stop...
```

```
               ***********************************************************
               *                                                         *
               *                     Initialize B220                     *
               *                                                         *
               ***********************************************************

0C57: A0 C8    init     ldy   #fnend-fnametbl ; Move fnames to $300.
0C59: B9 AA 20 :fnloop  lda   fnametbl,y
0C5C: 99 00 03          sta   fnames,y
0C5F: 88                dey
0C60: C0 FF             cpy   #$FF       ; Loop until
0C62: D0 F5             bne   :fnloop    ;  Y underflows.
0C64: A9 D0             lda   #<MEM      ; Initialize B220 memory to 0
0C66: 85 CA             sta   memptr
0C68: A9 20             lda   #>MEM
0C6A: 85 CB             sta   memptr+1
0C6C: A0 00             ldy   #0
0C6E: 98       :loop    tya
0C6F: 91 CA    :pagloop sta   (memptr),y
0C71: C8                iny
0C72: D0 FB             bne   :pagloop
0C74: E6 CB             inc   memptr+1
0C76: A5 CB             lda   memptr+1
0C78: C9 96             cmp   #>$9600
0C7A: 90 F2             bcc   :loop
0C7C: A2 36    reset    ldx   #B220end-B220strt-1 ; Clear B220 state
0C7E: A9 00             lda   #0
0C80: 95 90    :regloop sta   B220strt,x
0C82: CA                dex
0C83: 10 FB             bpl   :regloop
0C85: A2 13             ldx   #IOstend-IOstate-1 ; Rewind paper
0C87: 9D 05 1E :offlp   sta   IOstate,x  ; tapes and mag tapes.
0C8A: CA                dex
0C8B: 10 FA             bpl   :offlp
                        seti  OvHlt      ; Set Ovflow Halt mode.
0C8D: A9 FF             lda   #$FF
0C8F: 85 C7             sta   OvHlt      ; Set non-zero.
                        eom
0C91: 20 04 0D restart  jsr   disppanl   ; Init screen for B220
0C94: 20 33 0F          jsr   display    ;  panel & display state.
0C97: 4C 54 0B          jmp   newP       ; Start simulation.
```

```
         put   B220PANEL
**********************************************************
*                                                        *
*           B220 front panel display routines            *
*                                                        *
**********************************************************

off      equ   " "        ; blank (neon off)
on       equ   "*"        ; asterisk (neon on)

AR8      equ   $580       ; Line 4
AR4      equ   $600       ; Line 5
AR2      equ   $680       ; Line 6
AR1      equ   $700       ; Line 7
ARv      equ   $428       ; Line 9
BPC8     equ   $5A8       ; Line 12
BPC4     equ   $628       ; Line 13
BPC2     equ   $6A8       ; Line 14
BPC1     equ   $728       ; Line 15
BPCv     equ   $450       ; Line 17
STATlin  equ   $550       ; Line 19

B220col  equ   13-1       ; Leftmost title column
Acol     equ   6-1        ; Leftmost digit column of A
Rcol     equ   24-1       ; Leftmost digit column of R
Bcol     equ   6-1        ; Leftmost digit column of B
Pcol     equ   14-1       ; Leftmost digit column of P
Ccol     equ   22-1       ; Leftmost digit column of C
SW1col   equ   7-1        ; SW 1 column
RUNcol   equ   18-1       ; RUN column
ERRcol   equ   22-1       ; ERR column
COMPcol  equ   26-1       ; COMP column
OFLcol   equ   32-1       ; OFL column
RPTcol   equ   35-1       ; RPT column

* Register label addresses

Alab     equ   AR8+3
Rlab     equ   AR8+21
Blab     equ   BPC8+3
Plab     equ   BPC8+11
Clab     equ   BPC8+19
SWlab    equ   STATlin+3
ERRlab   equ   STATlin+ERRcol+2 ; Error type character
```

===== Page 22 =====

```
               >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
```

```
                              ===== Page 24 =====

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 v1.2"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 Config: I, Quit to BASIC: Q",00
               >175
0F22: 20 58 FC >176  disphelp jsr   HOME       ; Display help lines.
0F25: A0 00    >177           ldy   #0         ; (window is last 4 lines)
0F27: B9 93 0E >178  :helplp  lda   Help1,y
0F2A: F0 06    >179           beq   :done
0F2C: 20 ED FD >180           jsr   COUT
0F2F: C8       >181           iny
0F30: D0 F5    >182           bne   :helplp    ; (always)
               >183
0F32: 60       >184  :done    rts
```

```
               >186  **********************************************************
               >187  *                                                        *
               >188  *                   Display B220 State                   *
               >189  *                                                        *
               >190  **********************************************************
               >191
0F33: 20 45 0F >192  display  jsr   dispA      ; Display A
0F36: 20 4C 0F >193           jsr   dispR      ; Display R
0F39: 20 53 0F >194           jsr   dispB      ; Display B
0F3C: 20 5A 0F >195           jsr   dispP      ; Display P
0F3F: 20 61 0F >196           jsr   dispC      ; Display C
0F42: 4C 68 0F >197           jmp   dispSTAT   ; Disp Status & return.
               >198
0F45: A9 9A    >199  dispA    lda   #<Aattr    ; Register A attributes
0F47: A0 0C    >200           ldy   #>Aattr
0F49: 4C F5 0F >201           jmp   dispreg    ; Display the register.
               >202
0F4C: A9 B2    >203  dispR    lda   #<Rattr    ; Register R attributes
0F4E: A0 0C    >204           ldy   #>Rattr
0F50: 4C F5 0F >205           jmp   dispreg    ; Display the register.
               >206
0F53: A9 CA    >207  dispB    lda   #<Battr    ; Register B attributes
0F55: A0 0C    >208           ldy   #>Battr
0F57: 4C F5 0F >209           jmp   dispreg    ; Display the register.
               >210
0F5A: A9 DA    >211  dispP    lda   #<Pattr    ; Register P attributes
0F5C: A0 0C    >212           ldy   #>Pattr
0F5E: 4C F5 0F >213           jmp   dispreg    ; Display the register.
               >214
0F61: A9 EA    >215  dispC    lda   #<Cattr    ; Register C attributes
0F63: A0 0C    >216           ldy   #>Cattr
0F65: 4C F5 0F >217           jmp   dispreg    ; Display the register.
               >218
0F68: A9 50    >219  dispSTAT lda   #<STATlin  ; Set ptr to STATlin
0F6A: 85 CC    >220           sta   ptr
0F6C: A9 05    >221           lda   #>STATlin
0F6E: 85 CD    >222           sta   ptr+1
0F70: A2 00    >223           ldx   #0
0F72: A0 06    >224           ldy   #SW1col    ; Start at switch 1
0F74: B5 B6    >225  :swloop  lda   CSW,x      ; Is it on?
0F76: 20 CC 0F >226           jsr   INDshow    ; Display it's state
0F79: E8       >227           inx              ; Next switch
0F7A: E0 0A    >228           cpx   #10        ; Until done...
0F7C: 90 F6    >229           bcc   :swloop
0F7E: A0 11    >230           ldy   #RUNcol
0F80: A5 C0    >231           lda   RUN
0F82: 20 CC 0F >232           jsr   INDshow
0F85: A0 15    >233           ldy   #ERRcol
0F87: A5 C1    >234           lda   ERR
0F89: 20 CC 0F >235           jsr   INDshow
0F8C: A0 19    >236           ldy   #COMPcol
0F8E: A5 C2    >237           lda   COMP       ; <0, 0, >0: < = >
0F90: 30 07    >238           bmi   :lt
0F92: F0 0A    >239           beq   :eq
0F94: A2 0C    >240           ldx   #:gtstr-:ltstr ; Point to > string
0F96: 4C A0 0F >241           jmp   :show
               >242
0F99: A2 00    >243  :lt      ldx   #:ltstr-:ltstr ; Point to < string
0F9B: 4C A0 0F >244           jmp   :show
               >245
0F9E: A2 06    >246  :eq      ldx   #:eqstr-:ltstr ; Point to = string
0FA0: BD BA 0F >247  :show    lda   :ltstr,x
0FA3: F0 06    >248           beq   :next
0FA5: 91 CC    >249           sta   (ptr),y
0FA7: C8       >250           iny
0FA8: E8       >251           inx
0FA9: D0 F5    >252           bne   :show      ; (always)
```

```
               >253
0FAB: A0 1F    >254  :next    ldy   #OFLcol
0FAD: A5 C3    >255           lda   Ov         ; Overflow indicator
0FAF: 20 CC 0F >256           jsr   INDshow
0FB2: A0 22    >257           ldy   #RPTcol
0FB4: A5 C4    >258           lda   Rp         ; Repeat indicator
0FB6: 20 CC 0F >259           jsr   INDshow
0FB9: 60       >260           rts
               >261
0FBA: 3C       >262  :ltstr   asc   '<' ; Inverse
0FBB: A0 BD A0 >263           asc   " = >",00
0FC0: BC A0    >264  :eqstr   asc   "< "
0FC2: 3D       >265           asc   '=' ; Inverse
0FC3: A0 BE 00 >266           asc   " >",00
0FC6: BC A0 BD >267  :gtstr   asc   "< = "
0FCA: 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
0FCC: 18       >280  INDshow  clc              ; >0 ==> inv, 0 ==> norm
0FCD: 69 FF    >281           adc   #$FF       ; Set C if >0, reset if 0
0FCF: B1 CC    >282  :loop    lda   (ptr),y    ; Get indicator char
0FD1: 29 20    >283           and   #$20       ; Is it Upper Case?
0FD3: D0 06    >284           bne   :notuc     ; -No, leave it alone.
0FD5: B1 CC    >285           lda   (ptr),y    ; -Yes, turn off $40 bit
0FD7: 29 BF    >286           and   #$BF       ;   to avoid mousetext.
0FD9: D0 02    >287           bne   :switch    ; (always)
               >288
0FDB: B1 CC    >289  :notuc   lda   (ptr),y    ; Recover original char
0FDD: 90 04    >290  :switch bcc   :norm      ; Set to normal
0FDF: 29 7F    >291           and   #$7F       ; Set to inverse
0FE1: B0 02    >292           bcs   :store     ; (always)
               >293
0FE3: 09 80    >294  :norm    ora   #$80       ; Set to normal
0FE5: 91 CC    >295  :store   sta   (ptr),y
0FE7: C8       >296           iny              ; Advance to next char
0FE8: B1 CC    >297           lda   (ptr),y    ;  and examine it.
0FEA: 09 80    >298           ora   #$80       ; Force normal
0FEC: 49 A0    >299           eor   #"         " ; Space?
0FEE: F0 04    >300           beq   :done      ; -Yes, done.
0FF0: 29 E0    >301           and   #$E0       ; -No, digit?
0FF2: D0 DB    >302           bne   :loop      ; -No, keep going.
0FF4: 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
0FF5: 85 CC       >313  dispreg  sta   ptr         ; Set register attribute ptr
0FF7: 84 CD       >314           sty   ptr+1
0FF9: A0 00       >315           ldy   #0
0FFB: B1 CC       >316  :cpyattr lda   (ptr),y     ; Copy reg attributes to page 0
0FFD: 99 D3 00    >317           sta   linev,y
1000: C8          >318           iny
1001: C0 0A       >319           cpy   #10
1003: 90 F6       >320           bcc   :cpyattr
1005: B1 CC       >321           lda   (ptr),y     ; Addr of low byte of register
1007: 8D 1A 10    >322           sta   :reg+1
100A: C8          >323           iny
100B: B1 CC       >324           lda   (ptr),y
100D: A8          >325           tay               ; Set Y = rightmost column
100E: 18          >326           clc
100F: A5 CC       >327           lda   ptr         ; Advance ptr to digit mask
1011: 69 0C       >328           adc   #12
1013: 85 CC       >329           sta   ptr
1015: 90 02       >330           bcc   :reg
1017: E6 CD       >331           inc   ptr+1
1019: A5 00       >332  :reg     lda   0*0         ; Load register byte
101B: CE 1A 10    >333           dec   :reg+1      ;  and move to next highest.
101E: 85 D0       >334           sta   t1          ; Save current reg byte
1020: 20 33 10    >335           jsr   dispdig     ; Display lo digit of reg byte
1023: 88          >336           dey               ; Move left one column.
1024: 30 0C       >337           bmi   :done       ; Quit if done...
1026: 20 33 10    >338           jsr   dispdig     ; Display hi digit of reg byte
1029: 88          >339  :skip    dey               ; Move left.
102A: 30 06       >340           bmi   :done       ; -Display complete.
102C: B1 CC       >341           lda   (ptr),y     ; Check mask
102E: F0 F9       >342           beq   :skip       ; -Skip this screen column
1030: D0 E7       >343           bne   :reg        ; -Keep going...
                  >344
1032: 60          >345  :done    rts
                  >346
```

```
                 ***********************************************************
                 *                                                         *
                 *           Display one digit of B220 register            *
                 *                                                         *
                 ***********************************************************

1033: A5 D0      dispdig  lda   t1          ; Get (shifted) reg byte.
1035: 29 0F               and   #$0F        ; Mask low digit,
1037: 09 B0               ora   #$B0        ;  make ASCII digit,
1039: 91 D3               sta   (linev),y   ;   and store it on screen.
103B: 46 D0               lsr   t1          ; 1-bit to Carry
103D: A9 A0               lda   #off
103F: 90 02               bcc   :st1
1041: A9 AA               lda   #on
1043: 91 D5      :st1     sta   (line1),y   ; Store 1-bit state to screen
1045: 46 D0               lsr   t1          ; 2-bit to Carry
1047: A9 A0               lda   #off
1049: 90 02               bcc   :st2
104B: A9 AA               lda   #on
104D: 91 D7      :st2     sta   (line2),y   ; Store 2-bit state to screen
104F: 46 D0               lsr   t1          ; 4-bit to Carry
1051: A9 A0               lda   #off
1053: 90 02               bcc   :st4
1055: A9 AA               lda   #on
1057: 91 D9      :st4     sta   (line4),y   ; Store 4-bit state to screen
1059: 46 D0               lsr   t1          ; 8-bit to Carry
105B: A9 A0               lda   #off
105D: 90 02               bcc   :st8
105F: A9 AA               lda   #on
1061: 91 DB      :st8     sta   (line8),y   ; Store 8-bit state to screen
1063: 60                  rts
```

```
                62            put   B220EXEC1
               >1    * OPcode execute phase dispatch table
               >2
               >3    optabl   equ   *           ; Low byte of execute routines
1064: 18       >4             db    <HLT        ; S ---- 00 ---- HaLT
1065: 18       >5             db    <NOP        ; S ---- 01 ---- No OP
1066: 34       >6             db    <OPerr      ;        02
1067: 1B       >7             db    <PRD        ; S unnv 03 ADDR Pap tape RD
1068: 21       >8             db    <PRB        ; S u--v 04 ADDR Pap tape Rd, Br
1069: B8       >9             db    <PRI        ; S unnv 05 ADDR Pap tape Rd, Inv
106A: BB       >10            db    <PWR        ; S unn- 06 ADDR Pap tape WR
106B: E8       >11            db    <PWI        ; S u--- 07 ADDR Pap tape Wr, Int
106C: 0D       >12            db    <KAD        ; S ---- 08 ---- Keyboard ADd
106D: EB       >13            db    <SPO        ; S dnnv 09 ADDR Sup Print Out
106E: 34 34 34 >14            db    <OPerr,<OPerr,<OPerr,<OPerr,<OPerr,<OPerr
1074: 84       >15            db    <CAD        ; S ---v 10 ADDR Clear ADd (Abs)
1075: 6F       >16            db    <CSU        ; S ---v 11 ADDR Clear SUb (Abs)
1076: A4       >17            db    <ADD        ; S ---v 12 ADDR ADD (Abs)
1077: 36       >18            db    <SUB        ; S ---v 13 ADDR SUBtract (Abs)
1078: 4C       >19            db    <MUL        ; S ---- 14 ADDR MULtiply
1079: D3       >20            db    <DIV        ; S ---- 15 ADDR DIVide
107A: 4E       >21            db    <RND        ; S ---- 16 ---- RouND
107B: 70       >22            db    <EXT        ; S ---- 17 ADDR EXTract
107C: 98       >23            db    <CFA        ; S sLfv 18 ADDR Comp Fld A (R)
107D: 14       >24            db    <ADL        ; S ---- 19 ADDR ADd to Location
107E: 34 34 34 >25            db    <OPerr,<OPerr,<OPerr,<OPerr,<OPerr,<OPerr
1084: F4       >26            db    <IBB        ; S nnnn 20 ADDR Increase B, Br
1085: 07       >27            db    <DBB        ; S nnnn 21 ADDR Decrease B, Br
1086: 56       >28            db    <FAD        ; S n--v 22 ADDR Float ADd (Abs)
1087: 63       >29            db    <FSU        ; S n--v 23 ADDR Float SUb (Abs)
1088: 78       >30            db    <FMU        ; S ---- 24 ADDR Float MUltiply
1089: 09       >31            db    <FDV        ; S ---- 25 ADDR Float DiVide
108A: 8C       >32            db    <IFL        ; S sLnn 26 ADDR Inc Fld Loc
108B: D2       >33            db    <DFL        ; S sLnn 27 ADDR Dec Fld Loc
108C: E2       >34            db    <DLB        ; S sLnn 28 ADDR Dec fld loc,Ld B
108D: 8E       >35            db    <RTF        ; S -nn- 29 ADDR Record TransFer
108E: 34 34 34 >36            db    <OPerr,<OPerr,<OPerr,<OPerr,<OPerr,<OPerr
1094: 5D       >37            db    <BUN        ; S ---- 30 ADDR Branch UNcond
1095: 1A       >38            db    <BOF        ; S ---- 31 ADDR Branch OverFlow
1096: 27       >39            db    <BRP        ; S ---- 32 ADDR Branch RePeat
1097: 2D       >40            db    <BSA        ; S ---n 33 ADDR Branch Sign A
1098: 37       >41            db    <BCH        ; S ---v 34 ADDR Br Comp Hi (Lo)
1099: 4B       >42            db    <BCE        ; S ---v 35 ADDR Br Comp Eq (Un)
109A: 74       >43            db    <BFA        ; S sLnn 36 ADDR Branch Field A
109B: 70       >44            db    <BFR        ; S sLnn 37 ADDR Branch Field R
109C: C3       >45            db    <BCS        ; S u--- 38 ADDR Br Control Sw
109D: D0       >46            db    <SOR        ; S ---V 39 ---- Set Ov Remember
109E: 34 34 34 >47            db    <OPerr,<OPerr,<OPerr,<OPerr,<OPerr,<OPerr
10A4: E4       >48            db    <STA        ; S sLfv 40 ADDR STore A (R/B)
10A5: 4B       >49            db    <LDR        ; S ---- 41 ADDR LoaD R
10A6: 57       >50            db    <LDB        ; S ---v 42 ADDR LoaD B (Comp)
10A7: 7D       >51            db    <LSA        ; S ---n 43 ---- Load Sign A
10A8: 86       >52            db    <STP        ; S ---- 44 ADDR STore P
10A9: 9B       >53            db    <CLA        ; S ---v 45 ---- CLr A/R/AR/B/AB/T
10AA: BC       >54            db    <CLL        ; S ---- 46 ADDR CLear Location
10AB: 34       >55            db    <OPerr      ;        47
10AC: C7       >56            db    <SRA        ; S ---v 48 --nn Shft Rt A (AR/AS)
10AD: FC       >57            db    <SLA        ; S ---v 49 --nn Shft Lt A (AR/AS)
10AE: 34 34 34 >58            db    <OPerr,<OPerr,<OPerr,<OPerr,<OPerr,<OPerr
10B4: 6D       >59            db    <MTS        ; S uhhv 50 addr Mag Tape Search
10B5: 9F       >60            db    <MTC        ; S uhhK 51 addr Mag Tape sCan
10B6: A2       >61            db    <MRD        ; S un-v 52 addr Mag tape ReaD
10B7: AD       >62            db    <MRR        ; S un-v 53 addr Mt Read Record
10B8: B0       >63            db    <MIW        ; S unkk 54 addr Mt Init Write
```

```
10B9: B8       >64            db    <MIR       ; S un-- 55 addr Mt Init wr Rec
10BA: BB       >65            db    <MOW       ; S unkk 56 addr Mt OverWrite
10BB: C5       >66            db    <MOR       ; S un-- 57 addr Mt Overwr Rec
10BC: C8       >67            db    <MPF       ; S un-v 58 ---- Mt Pos Fwd
10BD: 12       >68            db    <MIB       ; S u--v 59 addr Mt Interr Branch
```

```
                              ===== Page 31 =====

               >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
10BE: 91       >74            db    >HLT+noAD  ; S ---- 00 ---- HaLT
10BF: 91       >75            db    >NOP+noAD  ; S ---- 01 ---- No OP
10C0: 8C       >76            db    >operr     ;        02
10C1: 11       >77            db    >PRD       ; S unnv 03 ADDR Pap tape RD
10C2: 11       >78            db    >PRB       ; S u--v 04 ADDR Pap tape Rd, Br
10C3: 11       >79            db    >PRI       ; S unnv 05 ADDR Pap tape Rd, Inv
10C4: 11       >80            db    >PWR       ; S unn- 06 ADDR Pap tape WR
10C5: 12       >81            db    >PWI       ; S u--- 07 ADDR Pap tape Wr, Int
10C6: 88       >82            db    >KAD+noAD  ; S ---- 08 ---- Keyboard ADd
10C7: 12       >83            db    >SPO       ; S dnnv 09 ADDR Sup Print Out
10C8: 8C 8C 8C >84            db    >operr,>operr,>operr,>operr,>operr,>operr
10CE: 13       >85            db    >CAD       ; S ---v 10 ADDR Clear ADd (Abs)
10CF: 13       >86            db    >CSU       ; S ---v 11 ADDR Clear SUbtr (Abs)
10D0: 13       >87            db    >ADD       ; S ---v 12 ADDR ADD (Abs)
10D1: 14       >88            db    >SUB       ; S ---v 13 ADDR SUBtract (Abs)
10D2: 14       >89            db    >MUL       ; S ---- 14 ADDR MULtiply
10D3: 14       >90            db    >DIV       ; S ---- 15 ADDR DIVide
10D4: 95       >91            db    >RND+noAD  ; S ---- 16 ---- RouND
10D5: 15       >92            db    >EXT       ; S ---- 17 ADDR EXTract
10D6: 15       >93            db    >CFA       ; S sLfv 18 ADDR Comp Fld A (R)
10D7: 14       >94            db    >ADL       ; S ---- 19 ADDR ADd to Location
10D8: 8C 8C 8C >95            db    >operr,>operr,>operr,>operr,>operr,>operr
10DE: 19       >96            db    >IBB       ; S nnnn 20 ADDR Increase B, Br
10DF: 1A       >97            db    >DBB       ; S nnnn 21 ADDR Decrease B, Br
10E0: 16       >98            db    >FAD       ; S n--v 22 ADDR Float ADd (Abs)
10E1: 17       >99            db    >FSU       ; S n--v 23 ADDR Float SUb (Abs)
10E2: 17       >100           db    >FMU       ; S ---- 24 ADDR Float MUltiply
10E3: 18       >101           db    >FDV       ; S ---- 25 ADDR Float DiVide
10E4: 18       >102           db    >IFL       ; S sLnn 26 ADDR Inc Fld Loc
10E5: 18       >103           db    >DFL       ; S sLnn 27 ADDR Dec Fld Loc
10E6: 18       >104           db    >DLB       ; S sLnn 28 ADDR Dec fld loc,Ld B
10E7: 19       >105           db    >RTF       ; S -nn- 29 ADDR Record TransFer
10E8: 8C 8C 8C >106           db    >operr,>operr,>operr,>operr,>operr,>operr
10EE: 1A       >107           db    >BUN       ; S ---- 30 ADDR Branch UNcond
10EF: 1A       >108           db    >BOF       ; S ---- 31 ADDR Branch OverFlow
10F0: 1A       >109           db    >BRP       ; S ---- 32 ADDR Branch RePeat
10F1: 1A       >110           db    >BSA       ; S ---n 33 ADDR Branch Sign A
10F2: 1A       >111           db    >BCH       ; S ---v 34 ADDR Br Comp Hi (Lo)
10F3: 1A       >112           db    >BCE       ; S ---v 35 ADDR Br Comp Eq (Un)
10F4: 1A       >113           db    >BFA       ; S sLnn 36 ADDR Branch Field A
10F5: 1A       >114           db    >BFR       ; S sLnn 37 ADDR Branch Field R
10F6: 1A       >115           db    >BCS       ; S u--- 38 ADDR Br Control Sw
10F7: 1A       >116           db    >SOR       ; S ---v 39 ---- Set Ov Remember
10F8: 8C 8C 8C >117           db    >operr,>operr,>operr,>operr,>operr,>operr
10FE: 1A       >118           db    >STA       ; S sLfv 40 ADDR STore A (R/B)
10FF: 1B       >119           db    >LDR       ; S ---- 41 ADDR LoaD R
1100: 1B       >120           db    >LDB       ; S ---v 42 ADDR LoaD B (Comp)
1101: 9B       >121           db    >LSA+noAD  ; S ---n 43 ---- Load Sign A
1102: 1B       >122           db    >STP       ; S ---- 44 ADDR STore P
1103: 9B       >123           db    >CLA+noAD  ; S ---v 45 ---- CLr A/R/AR/B/AB/T
1104: 1B       >124           db    >CLL       ; S ---- 46 ADDR CLear Location
1105: 8C       >125           db    >operr     ;        47
1106: 9B       >126           db    >SRA+noAD  ; S ---v 48 --nn Shft Rt A (AR/AS)
1107: 9B       >127           db    >SLA+noAD  ; S ---v 49 --nn Shft Lt A (AR/AS)
1108: 8C 8C 8C >128           db    >operr,>operr,>operr,>operr,>operr,>operr
110E: 1C       >129           db    >MTS       ; S uhhv 50 addr Mag Tape Search
110F: 1C       >130           db    >MTC       ; S uhhk 51 addr Mag Tape sCan
1110: 1C       >131           db    >MRD       ; S un-v 52 addr Mag tape ReaD
1111: 1C       >132           db    >MRR       ; S un-v 53 addr Mt Read Record
```

===== Page 32 =====

```
1112: 1C        >133            db    >MIW       ; S unkk 54 addr Mt Init Write
1113: 1C        >134            db    >MIR       ; S un-- 55 addr Mt Init wr Rec
1114: 1C        >135            db    >MOW       ; S unkk 56 addr Mt OverWrite
1115: 1C        >136            db    >MOR       ; S un-- 57 addr Mt Overwr Rec
1116: 9C        >137            db    >MPF+noAD  ; S un-v 58 ---- Mt Pos Fwd
1117: 1D        >138            db    >MIB       ; S u--v 59 addr Mt Interr Branch
```

===== Page 33 =====

```
                >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
1118: 4C 72 0B >151  NOP      jmp   fetch       ; Do nothing.
                >152
111B: 20 35 11 >153  PRD      jsr   ]prd        ; Paper tape ReaD
111E: 4C 72 0B >154           jmp   fetch
                >155
1121: A5 99    >156  PRB      lda   rC+sL       ; Paper tape Read & Branch
1123: 29 F0    >157           and   #$F0        ; Fake NN = 00 (100 words)
1125: 85 99    >158           sta   rC+sL
1127: A5 9A    >159           lda   rC+VV
1129: 29 0F    >160           and   #$0F
112B: 09 01    >161           ora   #$01        ;  and xeq sign 6/7.
112D: 85 9A    >162           sta   rC+VV
112F: 20 35 11 >163  :read    jsr   ]prd        ; Read "tape" until
1132: 4C 2F 11 >164           jmp   :read       ;  sign 6/7 xeq.
                >165
1135: 20 C6 11 >166  ]prd     jsr   ptread      ; Read disk into MEM
1138: A5 9A    >167           lda   rC+VV       ; Examine variant digit
113A: 29 08    >168           and   #$08        ; 8-bit on?
113C: 85 D3    >169           sta   linev       ; Set B-mod mask.
113E: A5 9A    >170           lda   rC+VV       ; Variant again...
1140: A0 00    >171           ldy   #0
1142: 29 01    >172           and   #$01        ; Execute 6/7 sign?
1144: F0 02    >173           beq   :noxeq      ; -No, ignore 6/7 sign.
1146: A0 06    >174           ldy   #6          ; -Yes, set xeq mask.
1148: 84 D4    >175  :noxeq   sty   linev+1
114A: A6 D0    >176  :scanlp  ldx   t1          ; Index to unit offset.
114C: 18       >177           clc               ; Advance unit offset.
114D: BD 07 1E >178           lda   rdroff+2,x ; Lo byte
1150: 69 06    >179           adc   #6
1152: 9D 07 1E >180           sta   rdroff+2,x
1155: 90 08    >181           bcc   :scan       ; No carry.
1157: FE 06 1E >182           inc   rdroff+1,x ; Carry into mid byte.
115A: D0 03    >183           bne   :scan       ; No carry.
115C: FE 05 1E >184           inc   rdroff,x    ; Carry into hi byte.
115F: A0 00    >185  :scan    ldy   #0          ; Scan sign digits
1161: B1 CA    >186           lda   (memptr),y ;  for 8/9 or 6/7.
1163: 25 D3    >187           and   linev       ; Variant 8-bit
1165: F0 0C    >188           beq   :noBmod     ; No B modification
1167: B1 CA    >189           lda   (memptr),y ; B modify ADDR.
1169: 29 01    >190           and   #$01        ; Turn off 8-bit
116B: 91 CA    >191           sta   (memptr),y
116D: 20 96 11 >192           jsr   Bmodmem     ; B-modify address.
1170: 4C 7B 11 >193           jmp   :cont
                >194
1173: B1 CA    >195  :noBmod  lda   (memptr),y ; Re-fetch sign digit
1175: 25 D4    >196           and   linev+1     ; Apply xeq mask (0/6)
1177: C9 06    >197           cmp   #6          ; Sign = 6 or 7?
1179: F0 08    >198           beq   :xeq        ; -Yes, execute it.
117B: 20 AC 11 >199  :cont    jsr   incmem      ; Advance memptr.
117E: C6 D1    >200           dec   NN          ; More words?
1180: D0 C8    >201           bne   :scanlp     ; -Yes, continue scan.
1182: 60       >202           rts               ; -No, return.
                >203
1183: A2 00    >204  :xeq     ldx   #0          ; Execute input word.
1185: B1 CA    >205  :xeqlp   lda   (memptr),y
1187: 95 98    >206           sta   rC,x
```

```
1189: C8       >207           iny
118A: E8       >208           inx
118B: E0 06    >209           cpx   #6
118D: D0 F6    >210           bne   :xeqlp
118F: 86 C6    >211           stx   skipincP   ; Don't inc P reg.
1191: 68       >212           pla              ; No return.
1192: 68       >213           pla
1193: 4C 93 0B >214           jmp   execute    ; Execute instruction.
               >215
1196: C8       >216  Bmodmem  iny              ; Advance to
1197: C8       >217           iny              ;  ADDR field.
1198: C8       >218           iny
1199: C8       >219           iny
119A: C8       >220           iny
119B: F8       >221           sed              ; / Decimal mode.
119C: 18       >222           clc
119D: B1 CA    >223           lda   (memptr),y
119F: 65 95    >224           adc   rB+1
11A1: 91 CA    >225           sta   (memptr),y
11A3: 88       >226           dey
11A4: B1 CA    >227           lda   (memptr),y
11A6: 65 94    >228           adc   rB
11A8: 91 CA    >229           sta   (memptr),y
11AA: D8       >230           cld              ; \ Binary mode.
11AB: 60       >231           rts
               >232
11AC: 18       >233  incmem   clc              ; Advance memptr
11AD: A5 CA    >234           lda   memptr     ;  to next word.
11AF: 69 06    >235           adc   #6
11B1: 85 CA    >236           sta   memptr
11B3: 90 02    >237           bcc   :nocarry
11B5: E6 CB    >238           inc   memptr+1   ; Propagate carry.
11B7: 60       >239  :nocarry rts
               >240
11B8: 4C 34 0C >241  PRI      jmp   OPerr      ; Unimplemented
```

```
              >243  zeroff   equ   line1       ; Zero offset flag
              >244  cmdfnx   equ   line2+1     ; File name index
              >245
11BB: 84 D5   >246  PWR      sty   zeroff      ; New file if offset=0.
11BD: 20 CE 11 >247          jsr   ptwrite     ; Paper tape WRite
11C0: 20 CF 12 >248          jsr   incoff      ; Increment unit offset
11C3: 4C 72 0B >249          jmp   fetch
              >250
11C6: A9 00   >251  ptread   lda   #0          ; PTRDR device class
11C8: 20 DB 11 >252          jsr   setread     ; Start read command
11CB: 4C D3 11 >253          jmp   ptrdwrt     ;  and do the I/O.
              >254
11CE: A9 02   >255  ptwrite  lda   #2          ; PTPCH device class
11D0: 20 EA 11 >256          jsr   setwrite    ; Start write command.
11D3: 20 DD 1D >257  ptrdwrt jsr   midNN       ; Get word count
11D6: 85 D1   >258           sta   NN          ;  in binary.
11D8: 4C FD 11 >259          jmp   doio        ; Do the I/O.
              >260
11DB: 85 D0   >261  setread  sta   t1          ; Set device class (0/2/4)
11DD: A0 03   >262           ldy   #3          ; Set I/O cmd to read file.
11DF: B9 55 12 >263  :loadlp lda   load,y
11E2: 99 5E 12 >264          sta   Bxxxx+1,y
11E5: 88      >265           dey
11E6: 10 F7   >266           bpl   :loadlp
11E8: 30 0D   >267           bmi   getfnx      ; (always)
              >268
11EA: 85 D0   >269  setwrite sta   t1          ; Set device class (0/2/4)
11EC: A0 03   >270           ldy   #3          ; Set I/O cmd to write file.
11EE: B9 59 12 >271  :savelp lda   save,y
11F1: 99 5E 12 >272          sta   Bxxxx+1,y
11F4: 88      >273           dey
11F5: 10 F7   >274           bpl   :savelp
11F7: 20 74 12 >275  getfnx  jsr   getfnxt1    ; Y = fnx, t1 ==> offset
11FA: 84 D8   >276           sty   cmdfnx      ; Save fnx.
11FC: 60      >277           rts
              >278
11FD: A2 00   >279  doio     ldx   #0          ; New ProDOS command.
11FF: A9 5D   >280           lda   #<Bxxxx     ; Start with command.
1201: A0 12   >281           ldy   #>Bxxxx
1203: 20 7B 20 >282          jsr   putpdcmd
1206: A4 D8   >283           ldy   cmdfnx      ; Y = file name index.
1208: B9 21 1E >284          lda   fnxfn,y
120B: A0 03   >285           ldy   #>fnames
120D: 20 7B 20 >286          jsr   putpdcmd    ; Add file name.
1210: A9 64   >287           lda   #<Aparm
1212: A0 12   >288           ldy   #>Aparm
1214: 20 7B 20 >289          jsr   putpdcmd    ; Add ",A$".
1217: A5 CB   >290           lda   memptr+1
1219: A4 CA   >291           ldy   memptr
121B: 20 9B 12 >292          jsr   puthx       ; Add hex address...
121E: A9 68   >293           lda   #<Lparm
1220: A0 12   >294           ldy   #>Lparm
1222: 20 7B 20 >295          jsr   putpdcmd    ; Add ",L$"
1225: A5 D1   >296           lda   NN          ; Binary word count
1227: 0A      >297           asl               ; NN * 2
1228: 65 D1   >298           adc   NN          ; NN * 3
122A: 85 CE   >299           sta   inptr
122C: A9 00   >300           lda   #0
122E: 69 00   >301           adc   #0          ; Hi byte of NN * 3
1230: 26 CE   >302           rol   inptr       ; Lo byte of NN * 6
1232: 2A      >303           rol
1233: 85 CF   >304           sta   inptr+1     ; Hi byte of NN * 6
1235: A4 CE   >305           ldy   inptr
1237: 20 9B 12 >306          jsr   puthx       ; Add hex length "xxxx"
123A: 86 D6   >307           stx   savex       ; Save X before "B" param
123C: A9 6C   >308           lda   #<Bparm
123E: A0 12   >309           ldy   #>Bparm
```

```
1240: 20 7B 20 >310           jsr   putpdcmd   ; Add ",B$"
1243: 20 B7 12 >311           jsr   putoff     ; Add hex offset "xxxxxx"
1246: A5 D5    >312           lda   zeroff     ; Create file on write?
1248: D0 02    >313           bne   :useB      ; -No, use B$offset.
124A: A6 D6    >314           ldx   savex      ; -Yes, no B param.
124C: 20 92 20 >315  :useB    jsr   pdosxeq    ; Execute ProDOS command.
124F: 90 03    >316           bcc   :ok        ; No error.
1251: 4C 44 0C >317           jmp   IOerr      ; I/O error.
1254: 60       >318  :ok      rts
               >319
1255: CC CF C1 >320  load     asc   "LOAD"
1259: D3 C1 D6 >321  save     asc   "SAVE"
125D: C2 F8 F8 >322  Bxxxx    asc   "Bxxxx ",00
1264: AC C1 A4 >323  Aparm    asc   ",A$",00
1268: AC CC A4 >324  Lparm    asc   ",L$",00
126C: AC C2 A4 >325  Bparm    asc   ",B$",00
               >326
               >327  * Get file name index (fnx) and offset displacement (t1)
               >328  *  Entry: t1 = fnx base (0:RDR, 2:PCH, 4:MTape)
               >329  *  Exit: A, t1 = Displacement to unit offset (0..15)
               >330  *        Y = file name index (0..7)
               >331  *        X unchanged.
               >332
1270: A9 04    >333  getMTt1  lda   #4         ; Mag tape fnx base
1272: 85 D0    >334           sta   t1
1274: A5 99    >335  getfnxt1 lda   rC+sL      ; Get unit #
1276: 29 E0    >336           and   #$E0       ; Unit = 0 or 1?
1278: D0 1E    >337           bne   :ioerr     ; -No, I/O error.
127A: A5 99    >338           lda   rC+sL      ; -Yes, isolate
127C: 29 10    >339           and   #$10       ;   unit #.
127E: F0 02    >340           beq   :zero      ; Unit 0.
1280: A9 01    >341           lda   #1         ; Unit 1.
1282: 18       >342  :zero    clc              ; Add fnx base: 0 (PTRDR),
1283: 65 D0    >343           adc   t1         ;  2 (PTPCH), 4 (MT unit).
1285: A8       >344           tay
1286: C9 04    >345           cmp   #4         ; Mag tape? (4 or 5)
1288: 90 08    >346           bcc   :fnx       ; -No, A = file name index.
128A: C9 05    >347           cmp   #5         ; -Yes, if MT unit = 1,
128C: 69 00    >348           adc   #0         ;   add 1.
128E: 79 13 1E >349           adc   mtlane-4,y ; Add lane 0/1.
1291: A8       >350           tay
1292: B9 19 1E >351  :fnx     lda   fnxoff,y   ; Disp to unit offset
1295: 85 D0    >352           sta   t1         ;  in t1.
1297: 60       >353           rts
               >354
1298: 4C 44 0C >355  :ioerr   jmp   IOerr      ; I/O error relay.
               >356
129B: 20 9F 12 >357  puthx    jsr   putbyte    ; Put first byte in hex
129E: 98       >358           tya              ;  and fall into putbyte.
129F: 48       >359  putbyte  pha              ; Save byte
12A0: 4A       >360           lsr
12A1: 4A       >361           lsr
12A2: 4A       >362           lsr
12A3: 4A       >363           lsr
12A4: 20 A8 12 >364           jsr   :stdig     ; Put hi hex digit
12A7: 68       >365           pla              ;  and then lo dig.
12A8: 29 0F    >366  :stdig   and   #$0F       ; Isolate digit
12AA: 09 B0    >367           ora   #"0"       ; Or in zone
12AC: C9 BA    >368           cmp   #$BA       ; >9?
12AE: 90 02    >369           bcc   :store     ; -No, store it.
12B0: 69 06    >370           adc   #6         ; -Yes, cvt to A..F
12B2: 9D 00 02 >371  :store   sta   IN,x       ; Add char to IN buffer.
12B5: E8       >372           inx
12B6: 60       >373           rts
               >374
12B7: A4 D0    >375  putoff   ldy   t1         ; Index to unit offset.
12B9: A9 03    >376           lda   #3         ; 3-byte binary offset
```

```
12BB: 85 CC    >377            sta   ptr         ; 3-byte offset.
12BD: B9 05 1E >378  :offlp    lda   rdroff,y
12C0: 48       >379            pha
12C1: 05 D5    >380            ora   zeroff      ; Update zero
12C3: 85 D5    >381            sta   zeroff      ;  offset flag.
12C5: 68       >382            pla
12C6: 20 9F 12 >383            jsr   putbyte
12C9: C8       >384            iny               ; Inc byte index
12CA: C6 CC    >385            dec   ptr         ; More bytes?
12CC: D0 EF    >386            bne   :offlp      ; -Yes, go again.
12CE: 60       >387            rts
               >388
12CF: A6 D0    >389  incoff    ldx   t1          ; Unit offset index.
12D1: 18       >390            clc
12D2: BD 07 1E >391            lda   rdroff+2,x ; Lo byte
12D5: 65 CE    >392            adc   inptr       ; Add length * 6
12D7: 9D 07 1E >393            sta   rdroff+2,x
12DA: BD 06 1E >394            lda   rdroff+1,x ; Mid byte
12DD: 65 CF    >395            adc   inptr+1
12DF: 9D 06 1E >396            sta   rdroff+1,x
12E2: 90 03    >397            bcc   :rts        ; Carry out?
12E4: FE 05 1E >398            inc   rdroff,x    ; -Yes, inc hi byte.
12E7: 60       >399  :rts      rts               ; -No, return.
               >400
12E8: 4C 34 0C >401  PWI       jmp   OPerr       ; Unimplemented
               >402
               >403  KAD       equ   ]stop       ; Kluge to allow rA mod.
```

```
12EB: 20 DD 1D >405  SPO      jsr   midNN      ; Get count (NN) in A
12EE: 85 D1    >406           sta   NN         ; NN = binary word count.
12F0: A0 00    >407  :nxword  ldy   #0
12F2: B1 CA    >408           lda   (memptr),y ; Get sign
12F4: C9 02    >409           cmp   #2         ; Alphanumeric?
12F6: D0 3A    >410           bne   :num       ; -No, numeric.
12F8: C8       >411  :nxchar  iny              ; -Yes, print alpha.
12F9: B1 CA    >412           lda   (memptr),y ; Get next char
12FB: C9 26    >413           cmp   #$26       ; "Tab" code?
12FD: F0 11    >414           beq   :tab       ; -Yes, do tab.
12FF: C9 02    >415           cmp   #$02       ; -No, "Ignore" code?
1301: F0 07    >416           beq   :ignore    ; -Yes, skip it.
1303: AA       >417           tax              ; -No, translate B220
1304: BD 29 1E >418           lda   b220asc,x  ;       char to ASCII.
1307: 20 ED FD >419           jsr   COUT       ;        and print it.
130A: C0 05    >420  :ignore  cpy   #5         ; Word complete?
130C: D0 EA    >421           bne   :nxchar    ; -No, keep going.
130E: F0 4E    >422           beq   :done      ; -Yes, word done (always)
               >423
1310: A2 00    >424  :tab     ldx   #0
1312: A5 24    >425           lda   CH
1314: DD 6A 13 >426  :nxtab   cmp   tabs,x     ; Find first tab
1317: 90 07    >427           bcc   :gottab    ;  greater than CH.
1319: E8       >428           inx
131A: E0 05    >429           cpx   #5
131C: D0 F6    >430           bne   :nxtab
131E: F0 EA    >431           beq   :ignore    ; (always) Skip if past tabs.
               >432
1320: 84 D0    >433  :gottab  sty   t1         ; Save Y
1322: BC 6A 13 >434           ldy   tabs,x     ; Get target tab position.
1325: A9 A0    >435  :prtblnk lda   #"         "
1327: 20 ED FD >436           jsr   COUT       ; Print blanks until at
132A: C4 24    >437           cpy   CH         ;  target tab position.
132C: D0 F7    >438           bne   :prtblnk
132E: A4 D0    >439           ldy   t1         ; Restore Y
1330: D0 D8    >440           bne   :ignore    ;  and continue. (always)
               >441
1332: A2 A0    >442  :num     ldx   #"         " ; Print blank if sign 0
1334: C9 00    >443           cmp   #0
1336: F0 09    >444           beq   :prtsign
1338: A2 AD    >445           ldx   #"-"       ; Print - if sign 1
133A: C9 01    >446           cmp   #1
133C: F0 03    >447           beq   :prtsign
133E: 09 B0    >448           ora   #"0"       ; Else print sign digit.
1340: AA       >449           tax
1341: 8A       >450  :prtsign txa
1342: 20 ED FD >451           jsr   COUT
1345: C8       >452  :nxbyte  iny              ; Print rest of number.
1346: B1 CA    >453           lda   (memptr),y
1348: 48       >454           pha
1349: 4A       >455           lsr
134A: 4A       >456           lsr
134B: 4A       >457           lsr
134C: 4A       >458           lsr              ; Hi digit it A
134D: 09 B0    >459           ora   #"0"       ; OR in zone
134F: 20 ED FD >460           jsr   COUT       ;  and print digit.
1352: 68       >461           pla              ; Recover low digit
1353: 29 0F    >462           and   #$0F       ; Isolate it
1355: 09 B0    >463           ora   #"0"       ;  add zone
1357: 20 ED FD >464           jsr   COUT       ;   and print it.
135A: C0 05    >465           cpy   #5         ; End of word?
135C: D0 E7    >466           bne   :nxbyte    ; -No, continue.
135E: C6 D1    >467  :done    dec   NN         ; -Yes, more words?
1360: F0 05    >468           beq   :quit      ; -No, all done.
1362: 20 AC 11 >469           jsr   incmem     ; -Yes, increment memptr.
1365: D0 89    >470           bne   :nxword    ; (always)
               >471
```

```
1367: 4C 72 0B >472  :quit    jmp   fetch
               >473
136A: 09 11 19 >474  tabs     db    9,17,25,33,41 ; SPO tab table
               >475
               >476
136F: A5 9A    >477  CSU      lda   rC+VV      ; CSU/CSA
1371: 29 0F    >478           and   #$0F       ; Isolate variant digit.
1373: C9 01    >479           cmp   #$01       ; CSA?
1375: D0 06    >480           bne   :csu       ; -No, CSU.
1377: A5 AA    >481           lda   rD+S       ; -Yes, CSA.
1379: 09 01    >482           ora   #$01       ;   Force sign negative.
137B: D0 11    >483           bne   loadrA     ; (always)
               >484
137D: A5 AA    >485  :csu     lda   rD+S       ; CSU
137F: 49 01    >486           eor   #$01       ; Flip the 1-bit
1381: 4C 8E 13 >487           jmp   loadrA     ;  and complete the load.
               >488
               >489
1384: A5 9A    >490  CAD      lda   rC+VV      ; CAD/CAA
1386: 29 0F    >491           and   #$0F       ; Isolate variant digit.
1388: C9 01    >492           cmp   #$01       ; CAA?
138A: F0 11    >493           beq   CAA        ; -Yes.
138C: A5 AA    >494           lda   rD+S       ; -No, CAD.  Sign unchanged.
138E: 85 9E    >495  loadrA   sta   rA+S       ; Set rA sign.
1390: A0 05    >496           ldy   #5
1392: B1 CA    >497  :cpyloop lda   (memptr),y
1394: 99 9E 00 >498           sta   rA,y
1397: 88       >499           dey
1398: D0 F8    >500           bne   :cpyloop
139A: 4C 72 0B >501           jmp   fetch
               >502
139D: A5 AA    >503  CAA      lda   rD+S       ; CAA
139F: 29 FE    >504           and   #$FE       ; Force sign positive
13A1: 4C 8E 13 >505           jmp   loadrA     ;  and complete the load.
```

```
13A4: A5 9A    >507  ADD      lda   rC+VV       ; ADD, ADA
13A6: 29 0F    >508           and   #$0F
13A8: C9 01    >509           cmp   #1          ; ADA?
13AA: D0 04    >510           bne   :add        ; -No, ADD.
13AC: A9 00    >511           lda   #0          ; -Yes, force MEM sign +
13AE: 85 AA    >512           sta   rD+S
13B0: 20 B6 13 >513  :add     jsr   ]add        ; Do the add.
13B3: 4C 72 0B >514           jmp   fetch
               >515
13B6: A5 9E    >516  ]add     lda   rA+S
13B8: 29 01    >517           and   #$01
13BA: 85 9E    >518           sta   rA+S        ; Force sign 0 (+) or 1 (-)
13BC: 45 AA    >519           eor   rD+S        ; Signs same or different?
13BE: 29 01    >520           and   #$01
13C0: D0 18    >521           bne   :subtr      ; -Different, subtract.
13C2: A0 05    >522           ldy   #5          ; -Same, add.
13C4: F8       >523           sed               ; / Decimal mode.
13C5: 18       >524           clc
13C6: B9 9E 00 >525  :addloop lda   rA,y        ; Do the addition...
13C9: 71 CA    >526           adc   (memptr),y
13CB: 99 9E 00 >527           sta   rA,y
13CE: 88       >528           dey
13CF: D0 F5    >529           bne   :addloop
13D1: D8       >530           cld               ; \ Back to binary.
13D2: 90 3F    >531           bcc   :done       ; Done.
               >532           seti  Ov          ; Signal Overflow
13D4: A9 FF    >532           lda   #$FF
13D6: 85 C3    >532           sta   Ov          ; Set non-zero.
               >532           eom
13D8: D0 39    >533           bne   :done       ; (always)
               >534
13DA: A0 01    >535  :subtr   ldy   #1          ; Compare magnitudes.
13DC: B9 9E 00 >536  :comloop lda   rA,y
13DF: D1 CA    >537           cmp   (memptr),y
13E1: F0 04    >538           beq   :cont       ; Equal, keep comparing.
13E3: B0 07    >539           bcs   :Abig       ; rA is bigger
13E5: 90 16    >540           bcc   :Asmall     ; rA is smaller
               >541
13E7: C8       >542  :cont    iny
13E8: C0 06    >543           cpy   #6
13EA: D0 F0    >544           bne   :comloop    ; If =, fall into :Abig.
13EC: A0 05    >545  :Abig    ldy   #5          ; Subtract MEM from rA.
13EE: F8       >546           sed               ; / Decimal mode.
13EF: B9 9E 00 >547  :subloop lda   rA,y
13F2: F1 CA    >548           sbc   (memptr),y
13F4: 99 9E 00 >549           sta   rA,y
13F7: 88       >550           dey
13F8: D0 F5    >551           bne   :subloop
13FA: D8       >552           cld               ; \ Back to binary.
13FB: F0 16    >553           beq   :done       ; (always)
               >554
13FD: A5 AA    >555  :Asmall  lda   rD+S        ; MEM - rA ==> rA
13FF: 29 01    >556           and   #$01        ; rA sign = MEM sign.
1401: 85 9E    >557           sta   rA+S
1403: A0 05    >558           ldy   #5
1405: F8       >559           sed               ; / Decimal mode.
1406: 38       >560           sec
1407: B1 CA    >561  :sloop   lda   (memptr),y
1409: F9 9E 00 >562           sbc   rA,y
140C: 99 9E 00 >563           sta   rA,y
140F: 88       >564           dey
1410: D0 F5    >565           bne   :sloop
1412: D8       >566           cld               ; \ Back to binary.
1413: 60       >567  :done    rts
```

```
1414: A5 9E    >569  ADL      lda   rA+S        ; Force rA sign
1416: 29 01    >570           and   #$01        ;  to 0 or 1.
1418: 85 9E    >571           sta   rA+S
141A: A2 FA    >572           ldx   #-6         ; MEM + rA ==> MEM
141C: B5 A4    >573  :pushlp  lda   rA+6,x      ; Push rA
141E: 48       >574           pha
141F: E8       >575           inx
1420: D0 FA    >576           bne   :pushlp
1422: 20 B6 13 >577           jsr   ]add        ; rA + MEM ==> rA
1425: A0 05    >578           ldy   #5          ; rA ==> MEM
1427: B9 9E 00 >579  :mvloop  lda   rA,y
142A: 91 CA    >580           sta   (memptr),y
142C: 68       >581           pla               ;  and pop rA.
142D: 99 9E 00 >582           sta   rA,y
1430: 88       >583           dey
1431: 10 F4    >584           bpl   :mvloop
1433: 4C 72 0B >585           jmp   fetch
               >586
1436: A5 9A    >587  SUB      lda   rC+VV       ; SUB, SUA
1438: 29 0F    >588           and   #$0F
143A: C9 01    >589           cmp   #1          ; SUA?
143C: F0 06    >590           beq   :setsign    ; -Yes, force operand neg.
143E: A5 AA    >591  :sub     lda   rD+S        ; -No, SUB.
1440: 29 01    >592           and   #$01        ; Invert
1442: 49 01    >593           eor   #$01        ;  operand
1444: 85 AA    >594  :setsign sta   rD+S        ;   sign
1446: 20 B6 13 >595           jsr   ]add        ;    and add.
1449: 4C 72 0B >596           jmp   fetch
```

```
144C: 20 52 14 >598  MUL      jsr   multiply   ; Multiply
144F: 4C 72 0B >599           jmp   fetch
               >600
1452: 45 9E    >601  multiply eor   rA+S       ; Multiply subroutine
1454: 29 01    >602           and   #$01
1456: 48       >603           pha              ; Save result sign
1457: A2 00    >604           ldx   #0
1459: A0 05    >605           ldy   #5
145B: B1 CA    >606  :init    lda   (memptr),y ; rD = multiplicand
145D: 99 AA 00 >607           sta   rD,y
1460: 99 B0 00 >608           sta   rD10,y     ; rD10 = multiplicand
1463: B9 9E 00 >609           lda   rA,y       ; rR = multiplier
1466: 99 A4 00 >610           sta   rR,y
1469: 96 9E    >611           stx   rA,y       ; rA = 0 (including sign)
146B: 88       >612           dey
146C: 10 ED    >613           bpl   :init
146E: A5 C3    >614           lda   Ov         ; FMU overflow pending?
1470: F0 02    >615           beq   :cont      ; -No, continue.
1472: 68       >616           pla              ; -Yes, discard result sign
1473: 60       >617           rts              ;   and return.
               >618
1474: 86 AA    >619  :cont    stx   rD+S       ; Clear rD sign
1476: 86 B0    >620           stx   rD10+S     ;  and rD10 sign.
1478: A0 04    >621           ldy   #4         ; 4 bits/digit.
147A: 18       >622  :shloop  clc              ; Shift in zeros.
147B: 26 B5    >623           rol   rD10+5     ; Multiply rD10 by 10.
147D: 26 B4    >624           rol   rD10+4
147F: 26 B3    >625           rol   rD10+3
1481: 26 B2    >626           rol   rD10+2
1483: 26 B1    >627           rol   rD10+1
1485: 26 B0    >628           rol   rD10
1487: 88       >629           dey
1488: D0 F0    >630           bne   :shloop
148A: A9 05    >631           lda   #5         ; Set multiplier byte
148C: 85 D0    >632           sta   t1         ;  count = 5.
148E: F8       >633           sed              ; / Decimal mode.
148F: A5 A9    >634  :ckadd1  lda   rR+5
1491: 29 0F    >635           and   #$0F       ; Low digit of multiplier
1493: F0 10    >636           beq   :ckadd10   ; Skip add1 if zero.
1495: A8       >637           tay              ; Y = add1 count.
1496: A2 05    >638  :add1    ldx   #5
1498: 18       >639           clc              ; rA = rA + rD
1499: B5 9E    >640  :add1lp  lda   rA,x
149B: 75 AA    >641           adc   rD,x
149D: 95 9E    >642           sta   rA,x
149F: CA       >643           dex
14A0: 10 F7    >644           bpl   :add1lp
14A2: 88       >645           dey              ; More adds?
14A3: D0 F1    >646           bne   :add1      ; -Yes.
14A5: A5 A9    >647  :ckadd10 lda   rR+5       ; Low multiplier byte
14A7: 29 F0    >648           and   #$F0       ; High digit of byte
14A9: F0 14    >649           beq   :shift     ; Skip add10 if zero.
14AB: 4A       >650           lsr
14AC: 4A       >651           lsr
14AD: 4A       >652           lsr
14AE: 4A       >653           lsr
14AF: A8       >654           tay              ; Y = add10 count.
14B0: A2 05    >655  :add10   ldx   #5
14B2: 18       >656           clc              ; rA = rA + rD10
14B3: B5 9E    >657  :add10lp lda   rA,x
14B5: 75 B0    >658           adc   rD10,x
14B7: 95 9E    >659           sta   rA,x
14B9: CA       >660           dex
14BA: 10 F7    >661           bpl   :add10lp
14BC: 88       >662           dey              ; More adds?
14BD: D0 F1    >663           bne   :add10     ; -Yes.
14BF: 20 85 1D >664  :shift   jsr   srT2       ; -No, shift |rA| & |rR|
14C2: A5 9E    >665            lda   rA+S       ;  right 2 digits
14C4: 85 9F    >666            sta   rA+1       ;   including rA sign.
14C6: 86 9E    >667            stx   rA+S       ; Clear rA sign.
14C8: C6 D0    >668            dec   t1         ; Keep going if more
14CA: D0 C3    >669            bne   :ckadd1    ;  multiplier digits.
14CC: D8       >670            cld              ; \ Back to binary.
14CD: 68       >671            pla              ; Recover product sign
14CE: 85 9E    >672            sta   rA+S       ;  and set rA & rR signs.
14D0: 85 A4    >673            sta   rR+S
14D2: 60       >674            rts
```

```
14D3: 20 D9 14 >676  DIV      jsr   divide     ; DIVide
14D6: 4C 72 0B >677           jmp   fetch
               >678
14D9: 45 9E    >679  divide   eor   rA+S
14DB: 29 01    >680           and   #$01
14DD: 48       >681           pha              ; Sign of quotient
14DE: A5 9E    >682           lda   rA+S
14E0: 85 A4    >683           sta   rR+S       ; Sign of remainder
14E2: C8       >684           iny              ; Y = 1: skip signs.
14E3: B9 9E 00 >685  :comp    lda   rA,y       ; Compare rA magnitude
14E6: D1 CA    >686           cmp   (memptr),y ;  with divisor magnitude.
14E8: 90 0D    >687           bcc   :divide    ; rA < MEM, so divide.
14EA: D0 05    >688           bne   :oflow     ; rA > MEM, overflow.
14EC: C8       >689           iny
14ED: C0 06    >690           cpy   #6
14EF: D0 F2    >691           bne   :comp
               >692  :oflow   seti  Ov         ; Signal overflow
14F1: A9 FF    >692           lda   #$FF
14F3: 85 C3    >692           sta   Ov         ; Set non-zero.
               >692           eom
14F5: 68       >693           pla              ; Drop result sign
14F6: 60       >694           rts              ;  and return.
               >695
14F7: A0 0A    >696  :divide  ldy   #10        ; Quotient digit count = 10.
14F9: 84 D0    >697           sty   t1
14FB: A0 05    >698           ldy   #5
14FD: B1 CA    >699  :div2rD  lda   (memptr),y ; Move divisor to rD
14FF: 99 AA 00 >700           sta   rD,y
1502: 88       >701           dey
1503: D0 F8    >702           bne   :div2rD
1505: 84 9E    >703           sty   rA+S       ; Clear sign of rA
1507: 84 AA    >704           sty   rD+S       ;  and rD.
1509: F8       >705           sed              ; / Decimal mode.
150A: A0 04    >706  :shift   ldy   #4         ; 4 bits/digit.
150C: 18       >707  :shiftlp clc              ; Shift AR left 1 digit
150D: 20 99 1D >708           jsr   slT        ;  shifting in zeros.
1510: 26 9E    >709           rol   rA+S       ;   (include sign in A)
1512: 88       >710           dey
1513: D0 F7    >711           bne   :shiftlp
1515: A2 00    >712           ldx   #0
1517: B5 9E    >713  :complp  lda   rA,x       ; Compare A with divisor
1519: D5 AA    >714           cmp   rD,x
151B: 90 25    >715           bcc   :zero      ; Speed up quotient zeros.
151D: D0 05    >716           bne   :sub       ; A > divisor
151F: E8       >717           inx
1520: E0 06    >718           cpx   #6
1522: D0 F3    >719           bne   :complp
1524: A2 05    >720  :sub     ldx   #5         ; A(ext) = A(ext) - D(ext).
1526: 38       >721           sec
1527: B5 9E    >722  :sublp   lda   rA,x
1529: F5 AA    >723           sbc   rD,x
152B: 95 9E    >724           sta   rA,x
152D: CA       >725           dex
152E: 10 F7    >726           bpl   :sublp
1530: 90 04    >727           bcc   :restore   ; Restore if underflow
1532: E6 A9    >728           inc   rR+5       ; Increment quotient digit.
1534: D0 EE    >729           bne   :sub       ; (always)
               >730
1536: A2 05    >731  :restore ldx   #5         ; Add divisor back to A.
1538: 18       >732           clc
1539: B5 9E    >733  :restlp  lda   rA,x
153B: 75 AA    >734           adc   rD,x
153D: 95 9E    >735           sta   rA,x
153F: CA       >736           dex
1540: 10 F7    >737           bpl   :restlp
1542: C6 D0    >738  :zero    dec   t1         ; Quotient complete?
1544: D0 C4    >739           bne   :shift     ; -No, keep dividing.
```

```
1546: 20 AE 1D >740           jsr   exchAR     ; -Yes, exchange A and R
1549: D8       >741           cld              ; \ Back to binary.
154A: 68       >742           pla
154B: 85 9E    >743           sta   rA+S       ; Set quotient sign.
154D: 60       >744           rts
```

```
154E: A5 A5    >746  RND      lda   rR+1       ; Hi digit of rR
1550: C9 50    >747           cmp   #$50       ; C=1 if hi digit >= 5.
1552: A2 A4    >748           ldx   #rR        ; Clear rR.
1554: 20 D0 1D >749           jsr   clear      ; (Doesn't disturb C)
1557: 90 14    >750           bcc   :done      ; Done if hi digit < 5.
1559: F8       >751           sed              ; / Decimal mode.
155A: 38       >752           sec              ; Add 1 to rA.
155B: A2 05    >753           ldx   #5
155D: B5 9E    >754  :rndloop lda   rA,x
155F: 69 00    >755           adc   #0
1561: 95 9E    >756           sta   rA,x
1563: CA       >757           dex
1564: D0 F7    >758           bne   :rndloop
1566: D8       >759           cld              ; \ Back to binary.
1567: 90 04    >760           bcc   :done
               >761           seti  Ov         ; Signal Overflow.
1569: A9 FF    >761           lda   #$FF
156B: 85 C3    >761           sta   Ov         ; Set non-zero.
               >761           eom
156D: 4C 72 0B >762  :done    jmp   fetch
               >763
1570: A0 05    >764  EXT      ldy   #5         ; Extract digits from rA
1572: B1 CA    >765  :extlp   lda   (memptr),y ;  where MEM digits are odd.
1574: 29 11    >766           and   #$11       ; Isolate odd bits
1576: AA       >767           tax              ; $00, $01, $10, $11.
1577: BD 86 15 >768           lda   :exttbl,x  ; $00, $0F, $F0, $FF.
157A: 39 9E 00 >769           and   rA,y       ; Mask rA digits
157D: 99 9E 00 >770           sta   rA,y
1580: 88       >771           dey
1581: 10 EF    >772           bpl   :extlp
1583: 4C 72 0B >773           jmp   fetch
               >774
1586: 00 0F    >775  :exttbl  db    $00,$0F    ; Indices $00, $01 used
1588: 03 02 01 >776  signtbl  db    3,2,1,0,7,6,5,4,8,9 ; CFx sign order
1592: 00 00 00 >777           db    0,0,0,0    ; (filler)
1596: F0 FF    >778           db    $F0,$FF    ; Indices $10, $11 used.
               >779
1598: A5 9A    >780  CFA      lda   rC+VV      ; CFA, CFR
159A: A2 A4    >781           ldx   #rR
159C: 29 01    >782           and   #$01       ; CFR?
159E: D0 02    >783           bne   :cfr       ; -Yes.
15A0: A2 9E    >784           ldx   #rA        ; No, CFA.
15A2: A5 9A    >785  :cfr     lda   rC+VV      ; Reload variant
15A4: 29 10    >786           and   #$10       ; Partial field bit
15A6: A8       >787           tay              ;  to Y.
15A7: A9 D0    >788           lda   #BNEop     ; Do signed compare.
15A9: 20 B8 15 >789           jsr   compare
15AC: 85 C2    >790           sta   COMP       ; Set COMPare indicator
15AE: A5 C1    >791           lda   ERR        ; Error detected?
15B0: D0 03    >792           bne   :err       ; -Yes, report it.
15B2: 4C 72 0B >793           jmp   fetch
               >794
15B5: 4C 4C 0C >795  :err     jmp   ]err
```

```
               **********************************************************
               *                                                        *
               * Compare register with (memptr), whole or partial field.*
               *                                                        *
               * Entry: X = Register addr, (memptr) = comparand addr    *
               *        Y = Whole (0) or partial (not 0)                *
               *        A = BNE (signed comp) or BCS (unsigned comp)    *
               *                                                        *
               *  Exit: A = COMP indicator state (<0, 0, >0)            *
               *                                                        *
               **********************************************************

15B8: 8D E2 15 compare  sta   :magonly   ; Signed/unsigned (BNE, BCS)
15BB: B5 00             lda   0,x        ; Save register sign
15BD: 8D E5 15          sta   :cmpsign+1 ;  for compare.
15C0: 8E 14 16          stx   :comp1+1   ; And save register
15C3: 8E 3F 16          stx   :comp2+1   ;  address for loads.
15C6: 8E 4A 16          stx   :byte+1
15C9: 84 CC             sty   ptr        ; Save whole/partial.
15CB: C0 00             cpy   #0         ; Whole/partial (0, not 0)
15CD: D0 06             bne   :partial   ; -Yes.
15CF: A9 00             lda   #0         ; -No, fake 0:0 field
15D1: A2 0B             ldx   #11        ;   and compare signs.
15D3: D0 0F             bne   :cmpsign   ; (always)

15D5: 20 BC 1D :partial jsr   splitsL    ; Split sL: A = s and X = L.
15D8: 18                clc              ; A = low digit, 1..10
15D9: 69 01             adc   #1         ;     low dig + 1, 2..11
15DB: 38                sec
15DC: 86 D0             stx   t1         ; Digit length
15DE: E5 D0             sbc   t1         ; A = hi digit #
15E0: 90 18             bcc   :flderr    ; <0 ==> Field error.
15E2: D0 1F    :magonly bne   :comp      ; >0 ==> Comp magnitudes.
15E4: A0 00    :cmpsign ldy   #0*0       ; =0 ==> Compare signs.
15E6: C4 AA             cpy   rD+S       ; Reg sign = MEM sign?
15E8: F0 15             beq   :nosign    ; -Yes, comp magnitudes.
15EA: B9 88 15          lda   signtbl,y  ; -No, translate reg sign
15ED: A4 AA             ldy   rD+S       ; MEM sign
15EF: BE 88 15          ldx   signtbl,y  ;  translated.
15F2: 86 D0             stx   t1
15F4: C5 D0             cmp   t1         ; Compare signs.
15F6: E6 CC             inc   ptr        ; Force no flip.
15F8: D0 26             bne   :neql      ; (always) Sign determines.

15FA: A5 C6    :flderr  lda   "F"        ; Signal Field error.
15FC: 85 C1             sta   ERR
15FE: 60                rts

15FF: 18       :nosign  clc              ; Exclude sign from field
1600: 69 01             adc   #1         ; Field start + 1
1602: CA                dex              ; Field length - 1
1603: 18       :comp    clc
1604: 69 01             adc   #1
1606: 4A                lsr              ; A = hi byte for compare
1607: A8                tay              ; Y = hi byte index
1608: B0 2E             bcs   :lodigit   ; C ==> lo digit of hi byte.
160A: CA       :hidigit dex              ; Next digit, too?
160B: D0 3C             bne   :byte      ; -Yes, comp whole byte.
160D: B1 CA             lda   (memptr),y ; MEM byte
160F: 29 F0             and   #$F0       ; -No, final digit.
1611: 85 D0             sta   t1
1613: B9 00 00 :comp1   lda   0*0,y      ; Reg byte
1616: 29 F0             and   #$F0       ; Hi digit
1618: C5 D0    :final   cmp   t1         ; Compare final digit.
161A: D0 04    :done    bne   :neql      ; =?
161C: A9 00             lda   #0         ; -Yes, A = 0.
161E: F0 06             beq   :fin       ; (always)
```

```

1620: A9 01    :neql    lda   #1
1622: B0 02             bcs   :fin        ; >
1624: A9 FF             lda   #-1         ; <
1626: A4 CC    :fin     ldy   ptr         ; Recover whole/partial
1628: D0 0D             bne   :noflip     ; Partial ==> no flip
162A: A6 AA             ldx   rD+S        ; Original sign
162C: F0 09             beq   :noflip     ; + if 0.
162E: E0 04             cpx   #4          ; Collate as + or -?
1630: B0 05             bcs   :noflip     ; + if >= 4.
1632: AA                tax               ; - if 1, 2, or 3.
1633: F0 02             beq   :noflip     ; Comp =, no flip.
1635: 49 80             eor   #$80        ; Exchange > and <.
1637: 60       :noflip  rts

1638: B1 CA    :lodigit lda   (memptr),y ; MEM byte
163A: 29 0F             and   #$0F        ; Lo digit
163C: 85 D0             sta   t1          ; Save for compare.
163E: B9 00 00 :comp2   lda   0*0,y       ; Reg byte
1641: 29 0F             and   #$0F        ; Lo digit
1643: C5 D0             cmp   t1          ; Compare digits.
1645: D0 D3             bne   :done       ; Done if unequal.
1647: F0 07             beq   :nxbyte     ; Else continue (always)

1649: B9 00 00 :byte    lda   0*0,y       ; Reg byte
164C: D1 CA             cmp   (memptr),y ; Compare w MEM.
164E: D0 CA             bne   :done       ; Done if unequal.
1650: C8       :nxbyte  iny               ; Advance byte index and
1651: CA                dex               ;  decrement digit count
1652: D0 B6             bne   :hidigit    ; Continue if digits left,
1654: F0 C4             beq   :done       ;  else done. (always)
```

```
                63            put   B220EXEC2
1656: 29 01    >1    FAD      and   #$01       ; Standardize sign of
1658: 85 AA    >2             sta   rD+S       ;  MEM operand (0/1).
165A: A5 9A    >3             lda   rC+VV      ; FAD or FAA?
165C: 29 0F    >4             and   #$0F
165E: 49 01    >5             eor   #$01
1660: D0 02    >6             bne   ]fad       ; -FAD, continue.
1662: 85 AA    >7             sta   rD+S       ; -FAA, force +.
1664: A5 99    >8    ]fad     lda   rC+sL      ; Get normalization limit.
1666: 4A       >9             lsr
1667: 4A       >10            lsr
1668: 4A       >11            lsr
1669: 4A       >12            lsr
166A: D0 02    >13            bne   :nonzero
166C: A9 0A    >14            lda   #10
166E: 85 D1    >15   :nonzero sta   NN         ; Save binary norm limit.
1670: A5 9E    >16            lda   rA+S       ; Standardize rA sign (0/1)
1672: 29 01    >17            and   #$01
1674: 85 9E    >18            sta   rA+S
1676: A0 05    >19            ldy   #5         ; Copy MEM operand to rD.
1678: B1 CA    >20   :mem2rD  lda   (memptr),y
167A: 99 AA 00 >21            sta   rD,y
167D: 88       >22            dey
167E: D0 F8    >23            bne   :mem2rD    ; (rD sign already set)
1680: 84 D0    >24            sty   t1         ; Init t1 = 0
1682: A2 01    >25            ldx   #EXP       ; Compare rA & rD magnitudes
1684: B5 9E    >26   :complp  lda   rA,x
1686: D5 AA    >27            cmp   rD,x
1688: 90 3B    >28            bcc   :Alt       ; rA < rD.
168A: D0 05    >29            bne   :Age       ; rA > rD.
168C: E8       >30            inx              ; rA = rD so far...
168D: E0 06    >31            cpx   #6
168F: D0 F3    >32            bne   :complp
1691: F8       >33   :Age     sed              ; / Decimal mode.
1692: A5 9F    >34            lda   rA+EXP     ; rA >= rD. C = 1.
1694: E5 AB    >35            sbc   rD+EXP     ; Operand misalignment
1696: F0 3D    >36            beq   :doarith   ; Misalignment = 0, go.
1698: C9 08    >37            cmp   #8         ; Is misalignment > 7?
169A: B0 7E    >38            bcs   :done      ; -Yes, rA unchanged.
169C: 4A       >39            lsr              ; -No, div by 2, C = odd.
169D: 90 0E    >40            bcc   :bytesh    ; Even, so shift bytes.
169F: A2 04    >41            ldx   #4         ; Odd.  4 bits / digit.
16A1: 18       >42   :digsh   clc              ; Shift rD right 1 digit.
16A2: 66 AC    >43            ror   rD+MANT
16A4: 66 AD    >44            ror   rD+MANT+1
16A6: 66 AE    >45            ror   rD+MANT+2
16A8: 66 AF    >46            ror   rD+MANT+3
16AA: CA       >47            dex
16AB: D0 F4    >48            bne   :digsh
16AD: A8       >49   :bytesh  tay              ; Byte shift count
16AE: F0 25    >50            beq   :doarith   ; -Ready to go.
16B0: A5 AE    >51   :bytenxt lda   rD+MANT+2  ; -Shift right 2 digits
16B2: 85 AF    >52            sta   rD+MANT+3
16B4: A5 AD    >53            lda   rD+MANT+1
16B6: 85 AE    >54            sta   rD+MANT+2
16B8: A5 AC    >55            lda   rD+MANT
16BA: 85 AD    >56            sta   rD+MANT+1
16BC: A9 00    >57            lda   #0
16BE: 85 AC    >58            sta   rD+MANT
16C0: 88       >59            dey
16C1: D0 ED    >60            bne   :bytenxt
16C3: F0 10    >61            beq   :doarith   ; (always)
               >62
16C5: A2 05    >63   :Alt     ldx   #5         ; Exchange rA and rD
16C7: B5 9E    >64   :exchAD  lda   rA,x       ;  so |rA| > |rD|.
16C9: B4 AA    >65            ldy   rD,x
16CB: 94 9E    >66            sty   rA,x
```

```
16CD: 95 AA    >67            sta   rD,x
16CF: CA       >68            dex
16D0: 10 F5    >69            bpl   :exchAD
16D2: 38       >70            sec              ; Now |rA| >= |rD|.
16D3: B0 BC    >71            bcs   :Age       ; (always)
               >72
16D5: A5 9E    >73   :doarith lda   rA+S       ; Compare signs.
16D7: C5 AA    >74            cmp   rD+S
16D9: D0 43    >75            bne   :subtr     ; -Different, subtract.
16DB: A2 03    >76            ldx   #3         ; -Same, add.
16DD: 18       >77            clc
16DE: B5 A0    >78   :add     lda   rA+MANT,x  ; rA mantissa =
16E0: 75 AC    >79            adc   rD+MANT,x  ;  rA mantissa +
16E2: 95 A0    >80            sta   rA+MANT,x  ;   rD mantissa.
16E4: 05 D0    >81            ora   t1         ; Summarize zero
16E6: 85 D0    >82            sta   t1         ;  mantissa.
16E8: CA       >83            dex
16E9: 10 F3    >84            bpl   :add
16EB: B0 06    >85            bcs   :carry     ; Carry out of mantissa.
16ED: A5 D0    >86            lda   t1         ; Result mantissa = 0?
16EF: F0 41    >87            beq   :clrexp    ; -Yes, Result = 0.
16F1: D0 43    >88            bne   :norm      ; -No, normalize. (always)
               >89
16F3: A5 9F    >90   :carry   lda   rA+EXP     ; -Carry into EXP field.
16F5: C9 99    >91            cmp   #$99       ; Is EXP = 99 (max)?
16F7: D0 0A    >92            bne   :adj       ; -No, shift right.
16F9: A9 01    >93            lda   #$01       ; -Yes, force EXP
16FB: 85 9F    >94            sta   rA+EXP     ;   to 01 (unshifted sum)
16FD: A9 00    >95            lda   #0         ;    and force rA sign
16FF: 85 9E    >96            sta   rA+S       ;     to 0.
1701: F0 13    >97            beq   :ovflo     ;      and overflow. (always)
               >98
1703: 38       >99   :adj     sec              ; Restore the carry out.
1704: A2 04    >100           ldx   #4         ; 4 bits / digit.
1706: 20 6C 1D >101  :srloop  jsr   srAM       ; -Shift mant 1 dig right.
1709: 18       >102           clc              ; Shift in zeroes.
170A: CA       >103           dex
170B: D0 F9    >104           bne   :srloop
170D: 18       >105           clc
170E: A5 9F    >106           lda   rA+EXP     ; Increment rA exponent.
1710: 69 01    >107           adc   #1
1712: 85 9F    >108           sta   rA+EXP
1714: 90 04    >109           bcc   :done      ; -No overflow.
               >110  :ovflo   seti  Ov         ; -Signal exponent overflow.
1716: A9 FF    >110           lda   #$FF
1718: 85 C3    >110           sta   Ov         ; Set non-zero.
               >110           eom
171A: D8       >111  :done    cld              ; \ Back to binary.
171B: 4C 72 0B >112           jmp   fetch
               >113
171E: A2 03    >114  :subtr   ldx   #3         ; Subtract.
1720: 38       >115           sec
1721: B5 A0    >116  :sub     lda   rA+MANT,x  ; rA mantissa =
1723: F5 AC    >117           sbc   rD+MANT,x  ;  rA mantissa -
1725: 95 A0    >118           sta   rA+MANT,x  ;   rD mantissa.
1727: 05 D0    >119           ora   t1         ; Summarize zero
1729: 85 D0    >120           sta   t1         ;  mantissa.
172B: CA       >121           dex
172C: 10 F3    >122           bpl   :sub
172E: A5 D0    >123           lda   t1         ; Result mantissa = 0?
1730: D0 04    >124           bne   :norm      ; -No, normalize.
1732: 85 9F    >125  :clrexp  sta   rA+EXP     ; -Yes, exponent = 0.
1734: F0 E4    >126           beq   :done      ; (always)
               >127
1736: A5 A0    >128  :norm    lda   rA+MANT    ; Normalize result.
1738: 29 F0    >129           and   #$F0       ; Hi digit = 0?
173A: D0 DE    >130           bne   :done      ; -No, all done.
```

```
173C: A2 04    >131          ldx   #4          ; -Yes, shift left 1 dig.
173E: 18       >132  :diglp  clc               ; Shift in zeroes.
173F: 26 A3    >133          rol   rA+MANT+3
1741: 26 A2    >134          rol   rA+MANT+2
1743: 26 A1    >135          rol   rA+MANT+1
1745: 26 A0    >136          rol   rA+MANT
1747: CA       >137          dex
1748: D0 F4    >138          bne   :diglp
174A: C6 D1    >139          dec   NN          ; Norm limit exceeded?
174C: 10 04    >140          bpl   :ok         ; -No, continue.
               >141          resi  RUN         ; -Limit exceeded, halt.
174E: A9 00    >141          lda   #0
1750: 85 C0    >141          sta   RUN         ; Zero indicator.
               >141          eom
1752: 38       >142  :ok     sec
1753: A5 9F    >143          lda   rA+EXP      ; Decrement rA exponent
1755: E9 01    >144          sbc   #1
1757: 85 9F    >145          sta   rA+EXP
1759: B0 DB    >146          bcs   :norm
175B: A2 9E    >147          ldx   #rA         ; Exponent underflow,
175D: 20 D0 1D >148          jsr   clear       ;  clear rA.
1760: 4C 1A 17 >149          jmp   :done
               >150
1763: 29 01    >151  FSU     and   #$01        ; Standardize sign of
1765: 85 AA    >152          sta   rD+S        ;  MEM operand (0/1).
1767: A5 9A    >153          lda   rC+VV       ; FSU or FSA?
1769: 29 0F    >154          and   #$0F
176B: C9 01    >155          cmp   #1
176D: F0 04    >156          beq   :setneg     ; -FSA, set operand -.
176F: A5 AA    >157          lda   rD+S        ; -FSU.
1771: 49 01    >158          eor   #$01        ;   Complement sign
1773: 85 AA    >159  :setneg sta   rD+S        ;    of operand,
1775: 4C 64 16 >160          jmp   ]fad        ;     and do FAD.
```

```
1778: 18       >162  FMU      clc              ; Floating MUltiply
1779: C8       >163           iny              ; Y = 1 (exponent field)
177A: F8       >164           sed              ; / Decimal mode.
177B: B1 CA    >165           lda   (memptr),y ; Operand exponent
177D: 85 CC    >166           sta   ptr        ; Save for restoration.
177F: 65 9F    >167           adc   rA+EXP     ;  + rA exponent
1781: 90 0A    >168           bcc   :notov     ; No overflow.
1783: C9 50    >169           cmp   #$50       ; Sum < 150?
1785: 90 0A    >170           bcc   :ok        ; -Yes, no overflow.
               >171           seti  Ov         ; -No, signal overflow
1787: A9 FF    >171           lda   #$FF
1789: 85 C3    >171           sta   Ov         ; Set non-zero.
               >171           eom
178B: B0 09    >172           bcs   :cont      ;   and continue a bit.
               >173
178D: C9 50    >174  :notov   cmp   #$50       ; Sum < 50?
178F: 90 67    >175           bcc   :unflow    ; -Yes, underflow.
1791: 38       >176  :ok      sec              ; -No, subtract extra
1792: E9 50    >177           sbc   #$50       ;   excess 50 and
1794: 85 D1    >178           sta   NN         ;    save result exponent.
1796: A9 00    >179  :cont    lda   #0         ; Clear operand and
1798: 91 CA    >180           sta   (memptr),y ;  rA exponents.
179A: 85 9F    >181           sta   rA+EXP
179C: A5 A0    >182           lda   rA+MANT    ; Is rA unnormalized?
179E: 29 F0    >183           and   #$F0
17A0: F0 56    >184           beq   :unflow    ; -Yes, underflow.
17A2: C8       >185           iny              ; Y = 2 (mantissa)
17A3: B1 CA    >186           lda   (memptr),y ; Is memory operand
17A5: 29 F0    >187           and   #$F0       ;  unnormalized?
17A7: F0 4F    >188           beq   :unflow    ; -Yes, underflow.
17A9: A5 AA    >189           lda   rD+S       ; Recover operand sign.
17AB: 20 52 14 >190           jsr   multiply   ; Do the multiply.
17AE: A5 C3    >191           lda   Ov         ; Overflow pending?
17B0: D0 3F    >192           bne   :ovflow    ; -Yes, quit.
17B2: A2 02    >193           ldx   #2         ; -No, shift rA & rR
17B4: B5 9F    >194  :shloop  lda   rA+1,x     ;  left one byte.
17B6: 95 9E    >195           sta   rA,x
17B8: E8       >196           inx
17B9: E0 06    >197           cpx   #6         ; Skip rR sign byte.
17BB: D0 05    >198           bne   :notsign
17BD: A5 A5    >199           lda   rR+1
17BF: 85 A3    >200           sta   rA+5
17C1: E8       >201           inx
17C2: E0 0B    >202  :notsign cpx   #11        ; Done?
17C4: D0 EE    >203           bne   :shloop    ; -No, continue.
17C6: A9 00    >204           lda   #0         ; -Yes, clear
17C8: 85 A9    >205           sta   rR+5       ;   low byte of rR.
17CA: A5 A0    >206           lda   rA+MANT    ; Is rA normalized?
17CC: 29 F0    >207           and   #$F0
17CE: D0 13    >208           bne   :normal    ; -Yes.
17D0: A0 04    >209           ldy   #4         ; -No, shift rA & rR
17D2: 18       >210  :shdig   clc              ;   left one digit.
17D3: 20 99 1D >211           jsr   slT
17D6: 88       >212           dey
17D7: D0 F9    >213           bne   :shdig
17D9: A5 D1    >214           lda   NN         ; Recover result exp
17DB: F0 1B    >215           beq   :unflow    ; Underflow if 0.
17DD: F8       >216           sed              ; / Decimal mode.
17DE: 38       >217           sec
17DF: E9 01    >218           sbc   #1         ; Compensate for shift.
17E1: 85 D1    >219           sta   NN
17E3: A5 D1    >220  :normal  lda   NN
17E5: 85 9F    >221           sta   rA+EXP     ; Set result exponent.
17E7: D8       >222  :done    cld              ; \ Binary mode.
17E8: A0 01    >223           ldy   #1         ; Restore memory
17EA: A5 CC    >224           lda   ptr        ;  operand's exponent.
17EC: 91 CA    >225           sta   (memptr),y
```

```
17EE: 4C 72 0B >226           jmp   fetch
               >227
17F1: A9 00    >228  :ovflow  lda   #0
17F3: 85 A4    >229           sta   rR+S       ; Clear rR sign
17F5: 4C E7 17 >230           jmp   :done      ;  and clean up.
               >231
17F8: 20 FE 17 >232  :unflow  jsr   clearAR    ; Clear rA and rR
17FB: 4C E7 17 >233           jmp   :done      ;  and clean up.
               >234
17FE: A2 9E    >235  clearAR  ldx   #rA        ; Clear rA.
1800: 20 D0 1D >236           jsr   clear
1803: A2 A4    >237           ldx   #rR        ; Clear rR.
1805: 20 D0 1D >238           jsr   clear
1808: 60       >239           rts
```

```
1809: C8       >241  FDV      iny              ; Floating DiVide (Y==>EXP)
180A: B1 CA    >242           lda   (memptr),y ; Save MEM exponent
180C: 85 CC    >243           sta   ptr        ;  for restoration
180E: A9 00    >244           lda   #0         ;   and clear it for
1810: 91 CA    >245           sta   (memptr),y ;    for divide.
1812: C8       >246           iny              ; Y ==> MEM mantissa
1813: B1 CA    >247           lda   (memptr),y ; Hi byte of mant
1815: 29 F0    >248           and   #$F0       ; Divisor normalized?
1817: F0 5D    >249           beq   :denorm    ; -No, overflow.
1819: A5 A0    >250           lda   rA+MANT    ; Hi byte of rA mant
181B: 29 F0    >251           and   #$F0       ; Dividend normalized?
181D: F0 67    >252           beq   :unflo     ; -No, underflow.
181F: F8       >253           sed              ; /Decimal mode.
1820: 38       >254           sec
1821: A5 9F    >255           lda   rA+EXP     ; Dividend exponent
1823: E5 CC    >256           sbc   ptr        ;  - divisor exponent.
1825: B0 07    >257           bcs   :chkov     ; *dend >= *isor, ck ovflo.
1827: 38       >258           sec              ; *dend <  *isor, ck unflo.
1828: E9 50    >259           sbc   #$50       ; Restore excess-50
182A: 90 5A    >260           bcc   :unflo     ; Exponent underflow.
182C: B0 05    >261           bcs   :ok        ; (always)
               >262
182E: 18       >263  :chkov   clc
182F: 69 50    >264           adc   #$50       ; Restore excess-50
1831: B0 3F    >265           bcs   :ovflo     ; Exponent overflow.
1833: 85 D1    >266  :ok      sta   NN         ; Save result exponent.
1835: A9 00    >267           lda   #0         ; Clear rA exponent
1837: 85 9F    >268           sta   rA+EXP     ;  for divide.
1839: A0 04    >269           ldy   #4         ; 4 bits/digit.
183B: 18       >270  :shrt    clc              ; Shift in zeros.
183C: 20 77 1D >271           jsr   srAMR      ; Shift rA mant & rR
183F: 88       >272           dey              ;  right one digit.
1840: D0 F9    >273           bne   :shrt
1842: A5 A4    >274           lda   rR+S       ; Save original rR sign
1844: 48       >275           pha
1845: A5 AA    >276           lda   rD+S       ; Y=0, A=MEM sign
1847: 20 D9 14 >277           jsr   divide     ; Divide clears decimal mode.
184A: 68       >278           pla              ; Restore original rR sign
184B: 85 A4    >279           sta   rR+S
184D: A5 9F    >280           lda   rA+1       ; Hi byte of quotient.
184F: 29 F0    >281           and   #$F0       ; Is hi digit = 0?
1851: D0 0C    >282           bne   :shrT2     ; -No, shift right 2 digs.
1853: A0 04    >283           ldy   #4         ; -Yes, shift right 1 dig.
1855: 18       >284  :shloop  clc              ; Shift in zeros.
1856: 20 75 1D >285           jsr   srT        ; Shift |rA| & |rR|
1859: 88       >286           dey              ;  right one digit.
185A: D0 F9    >287           bne   :shloop
185C: 18       >288           clc              ; Indicate no overflow.
185D: F0 0D    >289           beq   :setexp    ; (always)
               >290
185F: F8       >291  :shrT2   sed              ; / Decimal mode.
1860: 18       >292           clc
1861: A5 D1    >293           lda   NN
1863: 69 01    >294           adc   #1         ; EXP = EXP + 1
1865: 85 D1    >295           sta   NN
1867: B0 0D    >296           bcs   :denorm    ; Exponent overflow
1869: 20 85 1D >297           jsr   srT2       ; Make room for exponent
186C: A5 D1    >298  :setexp  lda   NN         ; Set quotient exponent.
186E: 85 9F    >299           sta   rA+EXP
1870: 90 0A    >300           bcc   :done      ; (always)
               >301
1872: A9 00    >302  :ovflo   lda   #0         ; On exponent overflow
1874: 85 9F    >303           sta   rA+EXP     ;  clear result exponent.
1876: 85 9E    >304  :denorm  sta   rA+S       ; Clear rA sign and
               >305           seti  Ov         ;  set Overflow indicator.
1878: A9 FF    >305           lda   #$FF
187A: 85 C3    >305           sta   Ov         ; Set non-zero.
```

===== Page 55 =====

```
               >305           eom
187C: A5 CC    >306  :done    lda   ptr         ; Recover MEM exponent
187E: A0 01    >307           ldy   #1          ;  and put it back into
1880: 91 CA    >308           sta   (memptr),y ;   divisor in memory.
1882: D8       >309           cld               ; \ Binary mode.
1883: 4C 72 0B >310           jmp   fetch
               >311
1886: 20 FE 17 >312  :unflo   jsr   clearAR     ; Clear rA and rR
1889: 4C 7C 18 >313           jmp   :done       ;  and finish up.
```

```
188C: A9 18    >315  IFL      lda   #CLCop     ; Patch ]dfl for IFL
188E: 8D 2B 19 >316           sta   ]clc
1891: A9 65    >317           lda   #ADCZop
1893: 8D 3A 19 >318           sta   ]adc
1896: A9 C9    >319           lda   #CMPIop
1898: 8D 3C 19 >320           sta   ]cmp
189B: A9 EA    >321           lda   #NOPop
189D: 8D 64 19 >322           sta   ]nop
18A0: A9 79    >323           lda   #ADCYop
18A2: 8D 67 19 >324           sta   ]sub
18A5: A9 C3    >325           lda   #Ov
18A7: 8D 86 19 >326           sta   ]Ov+3
18AA: 20 F7 18 >327           jsr   ]dfl       ; Do the IFL.
18AD: A9 C4    >328           lda   #Rp        ; Patch ]dfl back.
18AF: 8D 86 19 >329           sta   ]Ov+3
18B2: A9 F9    >330           lda   #SBCYop
18B4: 8D 67 19 >331           sta   ]sub
18B7: A9 38    >332           lda   #SECop
18B9: 8D 64 19 >333           sta   ]nop
18BC: A9 24    >334           lda   #BITZop
18BE: 8D 3C 19 >335           sta   ]cmp
18C1: A9 E5    >336           lda   #SBCZop
18C3: 8D 3A 19 >337           sta   ]adc
18C6: A9 EA    >338           lda   #NOPop
18C8: 8D 2B 19 >339           sta   ]clc
18CB: A5 C1    >340           lda   ERR        ; Error detected?
18CD: D0 10    >341           bne   ]errpt     ; -Yes, report it.
18CF: 4C 72 0B >342  ]fetch4  jmp   fetch
               >343
               >344  DFL      resi  Rp         ; Reset Repeat indicator.
18D2: A9 00    >344           lda   #0
18D4: 85 C4    >344           sta   Rp         ; Zero indicator.
               >344           eom
18D6: 20 F7 18 >345           jsr   ]dfl       ; Decrease FieLd
18D9: A5 C1    >346           lda   ERR        ; Error detected?
18DB: D0 02    >347           bne   ]errpt     ; -Yes, report it.
18DD: F0 F0    >348           beq   ]fetch4    ; (always)
               >349
18DF: 4C 4C 0C >350  ]errpt   jmp   ]err
               >351
               >352  DLB      resi  Rp         ; Reset Repeat indicator.
18E2: A9 00    >352           lda   #0
18E4: 85 C4    >352           sta   Rp         ; Zero indicator.
               >352           eom
18E6: 20 F7 18 >353           jsr   ]dfl       ; Decrease Field
18E9: A5 AD    >354           lda   rD+3       ; Load rB from rD 8:4.
18EB: 85 94    >355           sta   rB
18ED: A5 AE    >356           lda   rD+4
18EF: 85 95    >357           sta   rB+1
18F1: A5 C1    >358           lda   ERR        ; Error detected?
18F3: D0 EA    >359           bne   ]errpt     ; -Yes, report it.
18F5: F0 D8    >360           beq   ]fetch4    ; (always)
```

===== Page 57 =====

```
18F7: A2 AA    >362  ]dfl     ldx   #rD         ; Clear rD.
18F9: 20 D0 1D >363           jsr   clear
18FC: A2 B0    >364           ldx   #rD10       ; Clear rD10.
18FE: 20 D0 1D >365           jsr   clear
1901: 20 BC 1D >366           jsr   splitsL     ; A = s, X = L
1904: 18       >367           clc
1905: 69 01    >368           adc   #1          ; A = s + 1
1907: 4A       >369           lsr               ; A = (s+1)/2, C = even dig
1908: 08       >370           php               ; Push Carry status.
1909: A8       >371           tay               ; Y = low byte index
190A: A5 9A    >372           lda   rC+VV       ; NN
190C: 99 B0 00 >373           sta   rD10,y      ; rD10 = subtrahend
190F: B0 16    >374           bcs   :subtr      ; Even dig first, no shift.
1911: 86 D0    >375           stx   t1          ; Save X
1913: 98       >376           tya               ; Move Y to X.
1914: AA       >377           tax
1915: 16 B0    >378           asl   rD10,x      ; Odd dig first, shift
1917: 36 AF    >379           rol   rD10-1,x    ;  1 digit left.
1919: 16 B0    >380           asl   rD10,x
191B: 36 AF    >381           rol   rD10-1,x
191D: 16 B0    >382           asl   rD10,x
191F: 36 AF    >383           rol   rD10-1,x
1921: 16 B0    >384           asl   rD10,x
1923: 36 AF    >385           rol   rD10-1,x
1925: A6 D0    >386           ldx   t1          ; Restore X.
1927: 28       >387  :subtr   plp               ; Pop C.
1928: F8       >388           sed               ; / Decimal mode.
1929: 90 39    >389           bcc   ]nop        ; Not C = odd dig first.
192B: EA       >390  ]clc     nop               ; <Patch to CLC for IFL>
192C: CA       >391  :evendig dex               ; Both even and odd digs?
192D: D0 36    >392           bne   :byte       ; -Yes, subtr whole byte.
192F: B9 B0 00 >393           lda   rD10,y      ; -No, subtr final digit.
1932: 29 0F    >394           and   #$0F        ; Isolate even digit
1934: 85 D0    >395           sta   t1          ;  and save for subtract.
1936: B1 CA    >396           lda   (memptr),y  ; MEM byte
1938: 29 0F    >397           and   #$0F        ; Isolate even digit
193A: E5 D0    >398  ]adc     sbc   t1          ;  & subtr. <ADC for IFL>
193C: 24 10    >399  ]cmp     bit   $10         ; CMP# if IFL (to set C)
193E: 29 0F    >400           and   #$0F        ; Mask result
1940: 85 D0    >401           sta   t1          ;  and save it.
1942: B1 CA    >402           lda   (memptr),y  ; Recover MEM byte,
1944: 29 F0    >403           and   #$F0        ;  mask out even digit,
1946: 05 D0    >404           ora   t1          ;   OR in difference,
1948: 91 CA    >405           sta   (memptr),y  ;    and put it back.
194A: A4 AE    >406           ldy   rD+4        ; Save high 4 digits of
194C: 84 AF    >407           sty   rD+5        ;  difference in rD 8:4.
194E: A4 AD    >408           ldy   rD+3
1950: 84 AE    >409           sty   rD+4
1952: 85 AD    >410           sta   rD+3
1954: 08       >411           php               ; Push Carry status.
1955: A2 04    >412           ldx   #4          ; 4 bits/digit
1957: 26 AF    >413  :shlp    rol   rD+5        ; Shift rD left 1 digit
1959: 26 AE    >414           rol   rD+4        ;  to line up with rB.
195B: 26 AD    >415           rol   rD+3
195D: CA       >416           dex
195E: D0 F7    >417           bne   :shlp
1960: 28       >418           plp               ; Pop Carry status.
1961: 4C 80 19 >419           jmp   :done
               >420
1964: 38       >421  ]nop     sec               ; <Patch to NOP for IFL>
1965: B1 CA    >422  :byte    lda   (memptr),y  ; MEM byte
1967: F9 B0 00 >423  ]sub     sbc   rD10,y      ;  minus subtrahend
196A: 91 CA    >424           sta   (memptr),y  ;   back to MEM.
196C: 84 D0    >425           sty   t1          ; Save Y
196E: A4 AE    >426           ldy   rD+4        ; Save 4 hi digits of
1970: 84 AF    >427           sty   rD+5        ;  difference in rD 8:4.
1972: A4 AD    >428           ldy   rD+3
```

```
                              ===== Page 58 =====

1974: 84 AE    >429             sty   rD+4
1976: 85 AD    >430             sta   rD+3
1978: A4 D0    >431             ldy   t1         ; Restore Y
197A: 88       >432             dey
197B: 30 0B    >433             bmi   :flderr    ; Field error.
197D: CA       >434             dex              ; More digits?
197E: D0 AC    >435             bne   :evendig   ; -Yes, keep subtracting.
1980: D8       >436  :done      cld              ; \ -No. Back to binary.
1981: 90 04    >437             bcc   :noRpt     ; Underflow ==> no Rpt
               >438  ]Ov        seti  Rp         ; Set Rpt <Ov for IFL>
1983: A9 FF    >438             lda   #$FF
1985: 85 C4    >438             sta   Rp         ; Set non-zero.
               >438             eom
1987: 60       >439  :noRpt     rts
               >440
1988: A9 C6    >441  :flderr    lda   #"F"       ; Signal Field error
198A: 85 C1    >442             sta   ERR
198C: D8       >443             cld              ; Clear decimal mode.
198D: 60       >444             rts
```

```
198E: 84 CF    >446  RTF      sty   inptr+1    ; 'inptr+1' = 0
1990: 84 D0    >447           sty   t1         ; 't1' = 0
1992: 20 DD 1D >448           jsr   midNN      ; Extract NN (word count)
1995: 85 CE    >449           sta   inptr      ; Save binary NN (1..100)
1997: A6 95    >450           ldx   rB+1       ; Convert rB to MEM
1999: E0 9A    >451           cpx   #$99+1     ;  address in 'ptr'.
199B: B0 51    >452           bcs   :underr    ; Undigit error.
199D: A4 94    >453           ldy   rB
199F: C0 4A    >454           cpy   #$49+1
19A1: B0 4E    >455           bcs   :addrerr   ; Address error.
19A3: BD B3 1E >456           lda   BCDLadrl,x
19A6: 79 E7 1F >457           adc   BCDHadrl,y
19A9: 85 CC    >458           sta   ptr
19AB: BD 4D 1F >459           lda   BCDLadrh,x
19AE: 79 31 20 >460           adc   BCDHadrh,y
19B1: B0 3B    >461           bcs   :underr    ; Carry out ==> undigit.
19B3: 85 CD    >462           sta   ptr+1      ; 'ptr' = dest MEM addr.
19B5: A5 CE    >463           lda   inptr      ; Binary NN
19B7: 0A       >464           asl              ; NN * 2 (2..200)
19B8: 65 CE    >465           adc   inptr      ; NN * 3 (3..300)
19BA: 26 CF    >466           rol   inptr+1    ; Capture high bit.
19BC: 0A       >467           asl
19BD: 26 CF    >468           rol   inptr+1    ; NN * 6 (6..600)
19BF: AA       >469           tax              ; Byte count lo
19C0: A0 00    >470           ldy   #0
19C2: B1 CA    >471  :movelp  lda   (memptr),y ; Move bytes upward.
19C4: 91 CC    >472           sta   (ptr),y
19C6: CA       >473           dex              ; Dec byte count lo
19C7: F0 09    >474           beq   :ckhi      ; If 0, chk hi byte.
19C9: C8       >475  :cont    iny
19CA: D0 F6    >476           bne   :movelp
19CC: E6 CB    >477           inc   memptr+1   ; Advance ptr pages
19CE: E6 CD    >478           inc   ptr+1
19D0: D0 F0    >479           bne   :movelp    ; (always)
               >480
19D2: C6 CF    >481  :ckhi    dec   inptr+1    ; Dec byte count hi
19D4: 10 F3    >482           bpl   :cont      ; Continue if >= 0.
19D6: A5 D1    >483           lda   NN         ; NN = 00 (100)?
19D8: D0 02    >484           bne   :lt100     ; -No, less than 100.
19DA: E6 D0    >485           inc   t1         ; -Yes, set 100.
19DC: F8       >486  :lt100   sed              ; / Decimal mode.
19DD: 18       >487           clc
19DE: A5 95    >488           lda   rB+1       ; rB = rB + NN
19E0: 65 D1    >489           adc   NN
19E2: 85 95    >490           sta   rB+1
19E4: A5 94    >491           lda   rB
19E6: 65 D0    >492           adc   t1         ; 1 if NN = 0, else 0.
19E8: 85 94    >493           sta   rB
19EA: D8       >494           cld              ; \ Back to binary.
19EB: 4C 72 0B >495           jmp   fetch
               >496
19EE: 4C 4A 0C >497  :underr  jmp   UNDIGerr   ; Relay jump.
19F1: 4C 40 0C >498  :addrerr jmp   ADDRerr    ; Relay jump.
```

===== Page 60 =====

```
19F4: F8       >500  IBB      sed              ; / Decimal mode.
19F5: 18       >501           clc
19F6: A5 95    >502           lda   rB+1       ; rB = rB + rC(4:4)
19F8: 65 9A    >503           adc   rC+VV
19FA: 85 95    >504           sta   rB+1
19FC: A5 94    >505           lda   rB
19FE: 65 99    >506           adc   rC+sL
1A00: 85 94    >507           sta   rB
1A02: D8       >508           cld              ; \ Back to binary.
1A03: 90 58    >509           bcc   BUN        ; No overflow ==> branch
1A05: B0 66    >510           bcs   ]fetch3    ; Overflow ==> continue
               >511
1A07: F8       >512  DBB      sed              ; / Decimal mode.
1A08: 38       >513           sec
1A09: A5 95    >514           lda   rB+1       ; rB = rB - rC(4:4)
1A0B: E5 9A    >515           sbc   rC+VV
1A0D: 85 95    >516           sta   rB+1
1A0F: A5 94    >517           lda   rB
1A11: E5 99    >518           sbc   rC+sL
1A13: 85 94    >519           sta   rB
1A15: D8       >520           cld              ; \ Back to binary.
1A16: B0 45    >521           bcs   BUN        ; No underflow ==> branch
1A18: 90 53    >522           bcc   ]fetch3    ; Underflow. (always)
```

===== Page 61 =====

```
1A1A: A5 C3    >524  BOF      lda   Ov          ; Overflow indicator set?
1A1C: D0 02    >525           bne   :ovflo      ; -Yes, clear it and branch.
1A1E: F0 4D    >526           beq   ]fetch3     ; (always)
               >527
               >528  :ovflo   resi  Ov          ; Reset Overflow indicator
1A20: A9 00    >528           lda   #0
1A22: 85 C3    >528           sta   Ov          ; Zero indicator.
               >528           eom
1A24: 4C 5D 1A >529           jmp   BUN         ;  and take the branch.
               >530
1A27: A5 C4    >531  BRP      lda   Rp          ; Repeat indicator set?
1A29: D0 32    >532           bne   BUN         ; -Yes, branch.
1A2B: F0 40    >533           beq   ]fetch3     ; (always)
               >534
1A2D: A5 9A    >535  BSA      lda   rC+VV       ; Get comparand digit
1A2F: 29 0F    >536           and   #$0F
1A31: C5 9E    >537           cmp   rA+S        ; Equal to rA sign?
1A33: F0 28    >538           beq   BUN         ; -Yes, take branch.
1A35: D0 36    >539           bne   ]fetch3     ; (always)
               >540
1A37: A5 9A    >541  BCH      lda   rC+VV       ; BCH or BCL?
1A39: 29 01    >542           and   #$01
1A3B: F0 06    >543           beq   :bch        ; -BCH.
1A3D: A5 C2    >544           lda   COMP        ; -BCL.
1A3F: 30 1C    >545           bmi   BUN         ; Branch if Lo
1A41: 10 2A    >546           bpl   ]fetch3     ; (always)
               >547
1A43: A5 C2    >548  :bch     lda   COMP
1A45: F0 26    >549           beq   ]fetch3     ; Equal.
1A47: 10 14    >550           bpl   BUN         ; Branch if Hi
1A49: 30 22    >551           bmi   ]fetch3     ; (always)
               >552
1A4B: A5 9A    >553  BCE      lda   rC+VV       ; BCE or BCU?
1A4D: 29 01    >554           and   #$01
1A4F: F0 06    >555           beq   :bce        ; BCE.
1A51: A5 C2    >556           lda   COMP
1A53: D0 08    >557           bne   BUN         ; Branch if unequal.
1A55: F0 16    >558           beq   ]fetch3     ; (always)
               >559
1A57: A5 C2    >560  :bce     lda   COMP
1A59: F0 02    >561           beq   BUN         ; Branch if equal.
1A5B: D0 10    >562           bne   ]fetch3     ; (always)
               >563
1A5D: A5 9C    >564  BUN      lda   rC+ADDR     ; Set new P reg
1A5F: 85 96    >565           sta   rP
1A61: A5 9D    >566           lda   rC+ADDR+1
1A63: 85 97    >567           sta   rP+1
1A65: A5 CA    >568           lda   memptr      ;  and instptr.
1A67: 85 C8    >569           sta   instptr
1A69: A5 CB    >570           lda   memptr+1
1A6B: 85 C9    >571           sta   instptr+1
1A6D: 4C 72 0B >572  ]fetch3  jmp   fetch
```

```
1A70: A2 A4    >574  BFR      ldx   #rR         ; X points to rR
1A72: D0 02    >575           bne   ]bfr
               >576
1A74: A2 9E    >577  BFA      ldx   #rA         ; X points to rA
1A76: A4 9A    >578  ]bfr     ldy   rC+VV       ; Y = 2-digit comparand
1A78: A5 99    >579           lda   rC+sL
1A7A: 29 10    >580           and   #$10        ; s even or odd?
1A7C: F0 0E    >581           beq   :even       ; -Even, no digit swap.
1A7E: 98       >582           tya               ; -Odd, swap digits.
1A7F: C9 80    >583           cmp   #$80        ; Hi bit to C
1A81: 2A       >584           rol               ;  and rotate 1 bit.
1A82: C9 80    >585           cmp   #$80        ; Hi bit to C
1A84: 2A       >586           rol               ;  and rotate 1 bit.
1A85: C9 80    >587           cmp   #$80        ; Hi bit to C
1A87: 2A       >588           rol               ;  and rotate 1 bit.
1A88: C9 80    >589           cmp   #$80        ; Hi bit to C
1A8A: 2A       >590           rol               ;  and rotate 1 bit.
1A8B: A8       >591           tay
1A8C: 84 B5    >592  :even    sty   rD10+5      ; Expand comparand
1A8E: 84 B4    >593           sty   rD10+4      ;  to full width in rD10.
1A90: 84 B3    >594           sty   rD10+3
1A92: 84 B2    >595           sty   rD10+2
1A94: 84 B1    >596           sty   rD10+1
1A96: 98       >597           tya
1A97: 29 0F    >598           and   #$0F        ; Mask off hi sign digit.
1A99: 85 B0    >599           sta   rD10
1A9B: A5 CB    >600           lda   memptr+1    ; Push 'memptr' on stack.
1A9D: 48       >601           pha
1A9E: A5 CA    >602           lda   memptr
1AA0: 48       >603           pha
1AA1: A9 B0    >604           lda   #rD10       ; Point 'memptr' at rD10
1AA3: 85 CA    >605           sta   memptr
1AA5: A9 00    >606           lda   #0
1AA7: 85 CB    >607           sta   memptr+1
               >608
1AA9: A0 01    >609           ldy   #1          ; Partial field compare
1AAB: A9 B0    >610           lda   #BCSop      ; Unsigned compare
1AAD: 20 B8 15 >611           jsr   compare
1AB0: AA       >612           tax               ; Save A
1AB1: 68       >613           pla               ; Pop 'memptr'
1AB2: 85 CA    >614           sta   memptr
1AB4: 68       >615           pla
1AB5: 85 CB    >616           sta   memptr+1
1AB7: A5 C1    >617           lda   ERR         ; Error detected?
1AB9: D0 05    >618           bne   :err        ; -Yes, report it.
1ABB: 8A       >619           txa               ; Recover COMP flags
1ABC: F0 9F    >620           beq   BUN         ; -Branch if equal.
1ABE: D0 6E    >621           bne   ]fetch2     ; -Else NOP. (always)
               >622
1AC0: 4C 4C 0C >623  :err     jmp   ]err
               >624
1AC3: A5 99    >625  BCS      lda   rC+sL       ; Get switch #
1AC5: 4A       >626           lsr
1AC6: 4A       >627           lsr
1AC7: 4A       >628           lsr
1AC8: 4A       >629           lsr
1AC9: AA       >630           tax
1ACA: B5 B6    >631           lda   CSW,x       ; Get switch state
1ACC: D0 8F    >632           bne   BUN         ; -True, take branch.
1ACE: F0 5E    >633           beq   ]fetch2     ; -False, no branch.
```

```
1AD0: A5 9A    >635  SOR      lda   rC+VV      ; SOR / SOH / IOM?
1AD2: 29 0F    >636           and   #$0F
1AD4: C9 02    >637           cmp   #2         ; IOM?
1AD6: F0 05    >638           beq   :iom       ; -Yes.
1AD8: 85 C7    >639           sta   OvHlt      ; -No, set Ovflo mode.
1ADA: 4C 72 0B >640  :fetch   jmp   fetch
               >641
1ADD: A5 C7    >642  :iom     lda   OvHlt
1ADF: F0 F9    >643           beq   :fetch     ; No branch if SOR mode.
1AE1: 4C 5D 1A >644           jmp   BUN        ; Branch if SOH mode.
               >645
1AE4: A5 9A    >646  STA      lda   rC+VV      ; STA, STR, STB?
1AE6: 29 0F    >647           and   #$0F       ; Isolate reg variant.
1AE8: A2 A4    >648           ldx   #rR
1AEA: C9 01    >649           cmp   #1         ; STR?
1AEC: F0 08    >650           beq   :store     ; -Yes.
1AEE: A2 90    >651           ldx   #rBx
1AF0: C9 02    >652           cmp   #2         ; STB?
1AF2: F0 02    >653           beq   :store     ; -Yes.
1AF4: A2 9E    >654           ldx   #rA        ; STA
1AF6: A5 9A    >655  :store   lda   rC+VV      ; Partial field :store?
1AF8: 29 10    >656           and   #$10
1AFA: D0 0F    >657           bne   :stfield   ; -Yes, do it.
1AFC: 8E 02 1B >658           stx   :stloop+1  ; -No, full word store.
1AFF: A0 05    >659           ldy   #5
1B01: B9 00 00 >660  :stloop  lda   0*0,y      ; Store the register.
1B04: 91 CA    >661           sta   (memptr),y
1B06: 88       >662           dey
1B07: 10 F8    >663           bpl   :stloop
1B09: 30 23    >664           bmi   ]fetch2    ; (always)
               >665
1B0B: 8E 1C 1B >666  :stfield stx   :evendig+1 ; Save register
1B0E: 8E 32 1B >667           stx   :odddig+1  ;  address...
1B11: 20 BC 1D >668           jsr   splitsL    ; Split sL: A = s and X = L
1B14: 18       >669           clc
1B15: 69 01    >670           adc   #1         ; A = s + 1
1B17: 4A       >671           lsr              ; A = (s+1)/2, C = even dig
1B18: A8       >672           tay              ; Y = byte offset
1B19: 90 16    >673           bcc   :odddig    ; -Start digit is odd.
1B1B: B9 00 00 >674  :evendig lda   0*0,y      ; -Start digit is even.
1B1E: CA       >675           dex              ; Both even & odd digits?
1B1F: D0 1D    >676           bne   :byte      ; -Yes, move full byte.
1B21: E8       >677           inx              ; -No, restore dig counter.
1B22: 29 0F    >678           and   #$0F       ; Isolate even digit
1B24: 85 D0    >679           sta   t1         ;  and save it.
1B26: B1 CA    >680           lda   (memptr),y ; Get MEM byte,
1B28: 29 F0    >681           and   #$F0       ;  clear target digit,
1B2A: 05 D0    >682           ora   t1         ;   OR in new digit,
1B2C: 91 CA    >683           sta   (memptr),y ;    and put it back.
1B2E: 4C 72 0B >684  ]fetch2  jmp   fetch      ; All done.
               >685
1B31: B9 00 00 >686  :odddig  lda   0*0,y      ; Start digit is odd.
1B34: 29 F0    >687           and   #$F0       ; Isolate reg digit
1B36: 85 D0    >688           sta   t1         ;  and save it.
1B38: B1 CA    >689           lda   (memptr),y ; Get MEM byte,
1B3A: 29 0F    >690           and   #$0F       ;  clear target digit,
1B3C: 05 D0    >691           ora   t1         ;   OR in new digit,
1B3E: 91 CA    >692  :byte    sta   (memptr),y ;    and put it back.
1B40: 88       >693           dey              ; Move byte index.
1B41: 30 05    >694           bmi   :flderr    ; -Err if field too long.
1B43: CA       >695           dex              ; More digits?
1B44: D0 D5    >696           bne   :evendig   ; -Yes, continue.
1B46: F0 E6    >697           beq   ]fetch2    ; -No, finished. (always)
               >698
1B48: 4C 3C 0C >699  :flderr  jmp   FIELDerr   ; Report field error.
```

```
                             ===== Page 64 =====

1B4B: A0 05    >701  LDR      ldy   #5          ; MEM(ADDR) ==> rR
1B4D: B1 CA    >702  :ldr     lda   (memptr),y
1B4F: 99 A4 00 >703           sta   rR,y
1B52: 88       >704           dey
1B53: 10 F8    >705           bpl   :ldr
1B55: 30 41    >706           bmi   ]fetch1     ; (always)
               >707
1B57: A5 9A    >708  LDB      lda   rC+VV       ; LDB, LBC
1B59: A0 05    >709           ldy   #5
1B5B: 29 01    >710           and   #$01
1B5D: D0 0C    >711           bne   :lbc        ; Load rB Complement
1B5F: B1 CA    >712  :ldb     lda   (memptr),y
1B61: 85 95    >713           sta   rB+1
1B63: 88       >714           dey
1B64: B1 CA    >715           lda   (memptr),y
1B66: 85 94    >716           sta   rB
1B68: 4C 72 0B >717           jmp   fetch       ; -Yes, done.
               >718
1B6B: F8       >719  :lbc     sed               ; / Decimal mode
1B6C: 38       >720           sec               ;  for 10's complement.
1B6D: A9 00    >721  :ldbc    lda   #0
1B6F: F1 CA    >722           sbc   (memptr),y
1B71: 85 95    >723           sta   rB+1
1B73: 88       >724           dey
1B74: A9 00    >725           lda   #0
1B76: F1 CA    >726           sbc   (memptr),y
1B78: 85 94    >727           sta   rB
1B7A: D8       >728           cld               ; \ -Yes, back to binary.
1B7B: 90 1B    >729           bcc   ]fetch1     ; (always)
               >730
1B7D: A5 9A    >731  LSA      lda   rC+VV       ; Load Sign A
1B7F: 29 0F    >732           and   #$0F        ; Isolate new sign digit
1B81: 85 9E    >733           sta   rA+S        ;  and put into rA.
1B83: 4C 72 0B >734           jmp   fetch
```

===== Page 65 =====

```
1B86: A0 05    >736  STP      ldy   #5          ; rP + 1 ==> MEM(0:4)
1B88: F8       >737           sed               ; / Decimal mode
1B89: 18       >738           clc
1B8A: A5 97    >739           lda   rP+1
1B8C: 69 01    >740           adc   #1
1B8E: 91 CA    >741           sta   (memptr),y
1B90: 88       >742           dey
1B91: A5 96    >743           lda   rP
1B93: 69 00    >744           adc   #0
1B95: 91 CA    >745           sta   (memptr),y
1B97: D8       >746           cld               ; \ Back to binary
1B98: 4C 72 0B >747  ]fetch1  jmp   fetch       ; -Yes, done.
               >748
1B9B: A5 9A    >749  CLA      lda   rC+VV       ; CLA/R/B
1B9D: 4A       >750           lsr               ; 1-bit to C
1B9E: 85 D0    >751           sta   t1          ; Save mask
1BA0: 90 05    >752           bcc   :notA       ; rA not included.
1BA2: A2 9E    >753           ldx   #rA
1BA4: 20 D0 1D >754           jsr   clear       ; Clear rA.
1BA7: 46 D0    >755  :notA    lsr   t1          ; 2-bit to C
1BA9: 90 05    >756           bcc   :notR       ; rR not included.
1BAB: A2 A4    >757           ldx   #rR
1BAD: 20 D0 1D >758           jsr   clear       ; Clear rR.
1BB0: 46 D0    >759  :notR    lsr   t1          ; 4-bit to C.
1BB2: 90 05    >760           bcc   :fetch      ; rB not included.
1BB4: A2 90    >761           ldx   #rBx
1BB6: 20 D0 1D >762           jsr   clear       ; Clear rB.
1BB9: 4C 72 0B >763  :fetch   jmp   fetch
               >764
1BBC: A9 00    >765  CLL      lda   #0          ; CLear Location
1BBE: A0 05    >766           ldy   #5
1BC0: 91 CA    >767  :cllloop sta   (memptr),y
1BC2: 88       >768           dey
1BC3: 10 FB    >769           bpl   :cllloop
1BC5: 30 D1    >770           bmi   ]fetch1     ; (always)
```

===== Page 66 =====

```
1BC7: A5 9D    >772  SRA      lda   rC+ADDR+1  ; SRA, SRT, SRS nn
1BC9: 29 1F    >773           and   #$1F       ; Isolate count 0..19
1BCB: C9 10    >774           cmp   #$10       ; Greater than 9?
1BCD: 90 02    >775           bcc   :nocor     ; -No, don't correct.
1BCF: E9 06    >776           sbc   #6         ; -Yes, cnvrt to binary.
1BD1: 0A       >777  :nocor   asl              ; Multiply digit shift
1BD2: 0A       >778           asl              ;  count by 4 (bits/digit).
1BD3: A8       >779           tay              ; Y = bit shift count.
1BD4: A5 9A    >780           lda   rC+VV      ; SRA, SRT, SRS
1BD6: 29 0F    >781           and   #$0F
1BD8: C9 01    >782           cmp   #1         ; SRT?
1BDA: D0 08    >783           bne   :notsrt    ; -No.
1BDC: A6 9E    >784           ldx   rA+S       ; -Yes, SRT. Set rR sign
1BDE: 86 A4    >785           stx   rR+S       ;   to rA sign, then
1BE0: A2 75    >786           ldx   #<srT      ;    shift both A and R.
1BE2: D0 08    >787           bne   :setsh     ; Go shift. (always)
               >788
1BE4: A2 68    >789  :notsrt  ldx   #<srAS
1BE6: C9 02    >790           cmp   #2         ; SRS?
1BE8: F0 02    >791           beq   :setsh     ; -Yes, shift right A & Sign
1BEA: A2 6A    >792           ldx   #<srA      ; SRA
1BEC: 8E F4 1B >793  :setsh   stx   :shiftr+1  ; Set shift subroutine.
1BEF: 98       >794           tya              ; Is shift count = 0?
1BF0: F0 07    >795           beq   :fetch     ; -Yes, done.
1BF2: 18       >796  :nxbit   clc              ; Shift in zeros.
1BF3: 20 6A 1D >797  :shiftr  jsr   srA        ; (or srT or srAS)
1BF6: 88       >798           dey              ; Count exhausted?
1BF7: D0 F9    >799           bne   :nxbit     ; -No, keep shifting.
1BF9: 4C 72 0B >800  :fetch   jmp   fetch      ; -Yes, done.
```

===== Page 67 =====

```
1BFC: A5 9D    >802  SLA      lda   rC+ADDR+1  ; SLA, SLT, SLS nn
1BFE: 29 1F    >803           and   #$1F       ; Isolate count 0..19
1C00: C9 10    >804           cmp   #$10       ; Greater than 9?
1C02: 90 02    >805           bcc   :nocor     ; -No, don't correct.
1C04: E9 06    >806           sbc   #6         ; -Yes, cnvrt to binary.
1C06: AA       >807  :nocor   tax              ; X = shift count.
1C07: A5 9A    >808           lda   rC+VV      ; SLA, SLT, SLS?
1C09: 29 0F    >809           and   #$0F
1C0B: C9 01    >810           cmp   #1         ; SLT?
1C0D: F0 19    >811           beq   :slt       ; -Yes, shift left AR
1C0F: E0 00    >812           cpx   #0         ; -No, check count.
1C11: F0 12    >813           beq   :fetch     ; Done if count = 0.
1C13: C9 02    >814           cmp   #2         ; SLS?
1C15: F0 3C    >815           beq   :sls       ; -Yes, shift left A + Sign
1C17: A0 04    >816  :sla     ldy   #4         ; SLA. Shift 4 bits/digit.
1C19: A5 9F    >817  :nxbita  lda   rA+1       ; To rotate rA,
1C1B: 2A       >818           rol              ;  preset C to high bit.
1C1C: 20 A3 1D >819           jsr   slA        ; Rotate A left 1 bit.
1C1F: 88       >820           dey              ; More bits?
1C20: D0 F7    >821           bne   :nxbita    ; -Yes.
1C22: CA       >822           dex              ; More digits?
1C23: D0 F2    >823           bne   :sla       ; -Yes.
1C25: 4C 72 0B >824  :fetch   jmp   fetch
               >825
1C28: A5 A4    >826  :slt     lda   rR+S       ; Copy rR Sign
1C2A: 85 9E    >827           sta   rA+S       ;  to rA Sign.
1C2C: 8A       >828           txa              ; Is count = 0?
1C2D: F0 F6    >829           beq   :fetch     ; -Yes, done.
1C2F: E0 0A    >830           cpx   #10        ; -No, count >= 10?
1C31: 90 10    >831           bcc   :nxdig     ; -No, do general case.
1C33: 86 D0    >832           stx   t1         ; -Yes, special case SLT >= 10.
1C35: 20 AE 1D >833           jsr   exchAR     ; Exchange A and R magnitudes
1C38: A5 D0    >834           lda   t1         ; Recover count.
1C3A: 38       >835           sec
1C3B: E9 0A    >836           sbc   #10        ; Is count = 10?
1C3D: F0 E6    >837           beq   :fetch     ; -Yes, done.
1C3F: AA       >838           tax              ; -No, keep shifting.
1C40: A5 9F    >839           lda   rA+1       ; Hi magnitude digit.
1C42: 2A       >840           rol              ; High bit to C
1C43: A0 04    >841  :nxdig   ldy   #4         ; 4 bits/digit
1C45: A5 9F    >842  :nxbitt  lda   rA+1       ; To rotate rA, rR
1C47: 2A       >843           rol              ;  preset C to high bit.
1C48: 20 99 1D >844           jsr   slT        ; Rotate AR left 1 bit.
1C4B: 88       >845           dey              ; More bits?
1C4C: D0 F7    >846           bne   :nxbitt    ; -Yes.
1C4E: CA       >847           dex              ; More digits?
1C4F: D0 F2    >848           bne   :nxdig     ; -Yes.
1C51: F0 D2    >849           beq   :fetch     ; (always)
               >850
1C53: A0 04    >851  :sls     ldy   #4         ; SLS. 4 bits/digit
1C55: A5 9E    >852  :nxbit   lda   rA+S       ; Use sign digit
1C57: 29 0F    >853           and   #$0F       ;  and mask it.
1C59: C9 08    >854           cmp   #8         ; Hi bit of sign to C
1C5B: 20 A3 1D >855           jsr   slA        ; Rotate A left 1 bit
1C5E: A5 9E    >856           lda   rA+S       ;  then rotate sign.
1C60: 2A       >857           rol
1C61: 29 0F    >858           and   #$0F       ; Mask again
1C63: 85 9E    >859           sta   rA+S       ;  and put it back.
1C65: 88       >860           dey              ; More bits?
1C66: D0 ED    >861           bne   :nxbit     ; -Yes.
1C68: CA       >862           dex              ; More digits?
1C69: D0 E8    >863           bne   :sls       ; -Yes.
1C6B: F0 B8    >864           beq   :fetch     ; (always)
```

```
               >866  * Mag Tape Op Codes
               >867
               >868  blkcnt   equ   line2      ; Temp block count.
               >869
1C6D: 20 70 12 >870  MTS      jsr   getMTt1    ; Set t1 to offset disp.
1C70: A5 9A    >871           lda   rC+VV      ; MTS...MRW
1C72: 29 0F    >872           and   #$0F       ; Isolate variant
1C74: C9 04    >873           cmp   #4         ; MLS?
1C76: F0 14    >874           beq   :mls       ; -Yes, lane select.
1C78: C9 08    >875           cmp   #8         ; -No, MRW?
1C7A: F0 03    >876           beq   :mrw       ; -Yes, rewind.
1C7C: 4C 34 0C >877           jmp   OPerr      ; -No, unimplemented.
               >878
1C7F: A6 D0    >879  :mrw     ldx   t1
1C81: A9 00    >880           lda   #0         ; Set unit position
1C83: 9D 05 1E >881           sta   rdroff,x   ;  to zero.
1C86: 9D 06 1E >882           sta   rdroff+1,x
1C89: 9D 07 1E >883           sta   rdroff+2,x
1C8C: A5 9A    >884  :mls     lda   rC+VV
1C8E: 29 10    >885           and   #$10       ; Isolate lane low bit.
1C90: F0 02    >886           beq   :lane0
1C92: A9 01    >887           lda   #1         ; Lane 1
1C94: AA       >888  :lane0   tax              ; Save lane #
1C95: 98       >889           tya              ; fnx
1C96: 4A       >890           lsr              ; A = 2 (unit 0) or 3 (unit 1)
1C97: A8       >891           tay
1C98: 8A       >892           txa              ; Recover lane #.
1C99: 99 15 1E >893           sta   mtlane-2,y ; Select lane.
1C9C: 4C 72 0B >894           jmp   fetch
               >895
1C9F: 4C 34 0C >896  MTC      jmp   OPerr      ; Not implemented
               >897
1CA2: A9 04    >898  MRD      lda   #4         ; Mag tape device class
1CA4: 20 DB 11 >899           jsr   setread    ; Mag tape ReaD
1CA7: 20 23 1D >900           jsr   mtrw       ; Do the I/O.
1CAA: 4C 72 0B >901           jmp   fetch
               >902
1CAD: 4C 34 0C >903  MRR      jmp   OPerr      ; Not implemented
               >904
1CB0: 84 D5    >905  MIW      sty   zeroff     ; New file if offset=0.
1CB2: 20 1E 1D >906           jsr   mtwrite    ; Go write...
1CB5: 4C 72 0B >907           jmp   fetch
               >908
1CB8: 4C 34 0C >909  MIR      jmp   OPerr      ; Not implemented
               >910
1CBB: A9 FF    >911  MOW      lda   #$FF       ; Rewrite file if
1CBD: 85 D5    >912           sta   zeroff     ;  offset = 0.
1CBF: 20 1E 1D >913           jsr   mtwrite    ; Go write...
1CC2: 4C 72 0B >914           jmp   fetch
               >915
1CC5: 4C 34 0C >916  MOR      jmp   OPerr      ; Not implemented
               >917
1CC8: 20 BC 1D >918  MPF      jsr   splitsL
1CCB: 86 D7    >919           stx   blkcnt     ; # of blocks
1CCD: 20 70 12 >920           jsr   getMTt1    ; Set t1 = unit offset disp
1CD0: A9 58    >921           lda   #<600      ; Set offset increment
1CD2: 85 CE    >922           sta   inptr      ;  in bytes.
1CD4: A9 02    >923           lda   #>600
1CD6: 85 CF    >924           sta   inptr+1
1CD8: A5 9A    >925           lda   rC+VV      ; Check subop.
1CDA: 29 0F    >926           and   #$0F
1CDC: F0 0B    >927           beq   :mpf       ; Mag tape Position Fwd
1CDE: C9 01    >928           cmp   #1
1CE0: F0 11    >929           beq   :mpb       ; Mag tape Position Back
1CE2: C9 02    >930           cmp   #2
1CE4: F0 0A    >931           beq   :mpe       ; Mag tape Position End
1CE6: 4C 34 0C >932           jmp   OPerr      ; Unimplemented.
```

```
               >933
1CE9: 20 CF 12 >934  :mpf     jsr   incoff      ; Inc offset 100 words.
1CEC: C6 D7    >935           dec   blkcnt      ; More?
1CEE: D0 F9    >936           bne   :mpf        ; -Yes, repeat.
1CF0: 4C 72 0B >937  :mpe     jmp   fetch       ; -No, done.
               >938
1CF3: A6 D0    >939  :mpb     ldx   t1          ; Index to unit offset.
1CF5: 38       >940           sec
1CF6: BD 07 1E >941           lda   rdroff+2,x ; Decrement offset
1CF9: E5 CE    >942           sbc   inptr       ;  by 100 words.
1CFB: 9D 07 1E >943           sta   rdroff+2,x
1CFE: BD 06 1E >944           lda   rdroff+1,x
1D01: E5 CF    >945           sbc   inptr+1
1D03: 9D 06 1E >946           sta   rdroff+1,x
1D06: B0 03    >947           bcs   :nobor
1D08: DE 05 1E >948           dec   rdroff,x
1D0B: C6 D7    >949  :nobor   dec   blkcnt      ; More blocks?
1D0D: D0 E4    >950           bne   :mpb        ; -Yes, repeat.
1D0F: 4C 72 0B >951           jmp   fetch       ; -No, done.
               >952
1D12: A5 9A    >953  MIB      lda   rC+VV       ; MIB / MIE
1D14: 29 01    >954           and   #$01        ; Isolate variant.
1D16: D0 03    >955           bne   :mie
1D18: 4C 5D 1A >956           jmp   BUN         ; MIB always branches.
               >957
1D1B: 4C 72 0B >958  :mie     jmp   fetch       ; MIE never branches.
               >959
1D1E: A9 04    >960  mtwrite  lda   #4          ; Mag tape device class
1D20: 20 EA 11 >961           jsr   setwrite    ; Set to write file.
1D23: 20 BC 1D >962  mtrw     jsr   splitsL
1D26: 86 D7    >963           stx   blkcnt      ; # of blocks
1D28: A9 64    >964  :nxblock lda   #100        ; Set block length
1D2A: 85 D1    >965           sta   NN          ;  to 100 words.
1D2C: 20 FD 11 >966           jsr   doio        ; Do the I/O operation.
1D2F: 20 CF 12 >967           jsr   incoff      ; Increment unit offset.
1D32: A5 9B    >968           lda   rC+OP       ; Check opcode.
1D34: C9 52    >969           cmp   #$52        ; MRD?
1D36: D0 1E    >970           bne   :noBmod     ; -No, skip B-mod scan.
1D38: A5 9A    >971           lda   rC+VV       ; -Yes, does variant
1D3A: 29 08    >972           and   #$08        ;       specify B-mod?
1D3C: F0 18    >973           beq   :noBmod     ; -No, skip it.
1D3E: A0 00    >974  :Bmod    ldy   #0          ; Scan mem for B-mod.
1D40: B1 CA    >975           lda   (memptr),y ; Get sign digit.
1D42: C9 08    >976           cmp   #$08        ; Is sign 8 or 9?
1D44: 90 07    >977           bcc   :next       ; -No, skip word.
1D46: 29 01    >978           and   #$01        ; -Yes, reset 8 bit.
1D48: 91 CA    >979           sta   (memptr),y
1D4A: 20 96 11 >980           jsr   Bmodmem     ; Modify ADDR field.
1D4D: 20 AC 11 >981  :next    jsr   incmem      ; Advance memptr.
1D50: C6 D1    >982           dec   NN          ; More words?
1D52: D0 EA    >983           bne   :Bmod       ; -Yes, continue.
1D54: F0 0D    >984           beq   :more       ; -No, done. (always)
               >985
1D56: 18       >986  :noBmod  clc
1D57: A5 CA    >987           lda   memptr      ; Increment memptr
1D59: 69 58    >988           adc   #<600       ;  by 6 * 100.
1D5B: 85 CA    >989           sta   memptr
1D5D: A5 CB    >990           lda   memptr+1
1D5F: 69 02    >991           adc   #>600
1D61: 85 CB    >992           sta   memptr+1
1D63: C6 D7    >993  :more    dec   blkcnt      ; More blocks?
1D65: D0 C1    >994           bne   :nxblock    ; -Yes, do them.
1D67: 60       >995           rts               ; -No, done.
```

```
               >997  ***********************************************************
               >998  *                                                         *
               >999  *            Utility Shifting Subroutines                 *
               >1000 *                                                         *
               >1001 ***********************************************************
               >1002
               >1003 ]keep    equ   */256      ; Keep here to 'kend' on one page.
               >1004
1D68: 66 9E    >1005 srAS     ror   rA         ; rA & sign right 1 bit
1D6A: 66 9F    >1006 srA      ror   rA+1       ; Sign not included
1D6C: 66 A0    >1007 srAM     ror   rA+2       ; FP mantissa
1D6E: 66 A1    >1008          ror   rA+3
1D70: 66 A2    >1009          ror   rA+4
1D72: 66 A3    >1010          ror   rA+5
1D74: 60       >1011          rts
               >1012
1D75: 66 9F    >1013 srT      ror   rA+1       ; |rA| & |rR| right 1 bit
1D77: 20 6C 1D >1014 srAMR    jsr   srAM       ; Shift rA Mantissa & |rR|
1D7A: 66 A5    >1015 srR      ror   rR+1       ; Shift |rR|
1D7C: 66 A6    >1016          ror   rR+2
1D7E: 66 A7    >1017          ror   rR+3
1D80: 66 A8    >1018          ror   rR+4
1D82: 66 A9    >1019          ror   rR+5
1D84: 60       >1020          rts
               >1021
1D85: A2 0A    >1022 srT2     ldx   #10        ; |rA| & |rR| right
1D87: B5 9E    >1023 :shloop  lda   rA,x       ;  2 digits (1 byte).
1D89: E0 05    >1024          cpx   #5         ; About to store in rR+S?
1D8B: D0 04    >1025          bne   :cont      ; -No, continue.
1D8D: 85 A5    >1026          sta   rR+1       ; -Yes, skip rR sign.
1D8F: F0 02    >1027          beq   :next      ;   and on to next byte.
1D91: 95 9F    >1028 :cont    sta   rA+1,x
1D93: CA       >1029 :next    dex
1D94: D0 F1    >1030          bne   :shloop    ; Exclude rA sign.
1D96: 86 9F    >1031          stx   rA+1       ; Shift in zeros.
1D98: 60       >1032          rts
               >1033
1D99: 26 A9    >1034 slT      rol   rR+5       ; Rotate |rR| & |rA| left
1D9B: 26 A8    >1035          rol   rR+4       ;  one bit.
1D9D: 26 A7    >1036          rol   rR+3
1D9F: 26 A6    >1037          rol   rR+2
1DA1: 26 A5    >1038          rol   rR+1       ; Fall into slA.
               >1039
1DA3: 26 A3    >1040 slA      rol   rA+5       ; Rotate |rA| left 1 bit
1DA5: 26 A2    >1041          rol   rA+4
1DA7: 26 A1    >1042          rol   rA+3
1DA9: 26 A0    >1043          rol   rA+2
1DAB: 26 9F    >1044          rol   rA+1
1DAD: 60       >1045          rts
               >1046
1DAE: A2 05    >1047 exchAR   ldx   #5         ; Exchange |rA| and |rR|
1DB0: B5 9E    >1048 :exch    lda   rA,x       ; (equivalent to SLT 10)
1DB2: B4 A4    >1049          ldy   rR,x
1DB4: 95 A4    >1050          sta   rR,x
1DB6: 94 9E    >1051          sty   rA,x
1DB8: CA       >1052          dex
1DB9: D0 F5    >1053          bne   :exch
1DBB: 60       >1054          rts
               >1055
               >1056 ]kend    equ   */256      ; Warn if page crossing
               >1057          err   ]kend-]keep ; between ]keep and ]kend.
```

```
                              ===== Page 71 =====

               >1059 *********************************************************
               >1060 *                                                       *
               >1061 *         Split sL field into A = s and X = L           *
               >1062 *                                                       *
               >1063 *********************************************************
               >1064
1DBC: A5 99    >1065 splitsL  lda   rC+sL      ; Get field specifier
1DBE: 29 0F    >1066          and   #$0F       ; L = digit count
1DC0: D0 02    >1067          bne   :notz
1DC2: A9 0A    >1068          lda   #10        ; "0" ==> 10
1DC4: AA       >1069 :notz    tax              ; X = digit count (L)
1DC5: A5 99    >1070          lda   rC+sL
1DC7: 4A       >1071          lsr              ; Isolate field start s
1DC8: 4A       >1072          lsr
1DC9: 4A       >1073          lsr
1DCA: 4A       >1074          lsr
1DCB: D0 02    >1075          bne   :ret
1DCD: A9 0A    >1076          lda   #10        ; "0" ==> 10
1DCF: 60       >1077 :ret     rts              ; A = start digit (s)
               >1078
               >1079 *********************************************************
               >1080 *                                                       *
               >1081 *                    Clear Register                     *
               >1082 *                                                       *
               >1083 * At entry: X = Register address                        *
               >1084 * At exit:  A = 0, X = $FF                              *
               >1085 *                                                       *
               >1086 *********************************************************
               >1087
1DD0: 8E D8 1D >1088 clear    stx   :clrloop+1 ; Save reg address
1DD3: A2 05    >1089          ldx   #5
1DD5: A9 00    >1090          lda   #0
1DD7: 95 00    >1091 :clrloop sta   0*0,x      ; Clear the register.
1DD9: CA       >1092          dex
1DDA: 10 FB    >1093          bpl   :clrloop
1DDC: 60       >1094          rts
               >1095
               >1096 *********************************************************
               >1097 *                                                       *
               >1098 *           Extract NN from 3:2 field of rC             *
               >1099 *                                                       *
               >1100 * Returns NN in BCD in 'NN', in binary in A. X unchanged.*
               >1101 *                                                       *
               >1102 *********************************************************
               >1103
1DDD: A5 99    >1104 midNN    lda   rC+sL      ; Extract NN from xN Nx.
1DDF: 29 0F    >1105          and   #$0F       ; Return binary NN in A
1DE1: A8       >1106          tay
1DE2: 0A       >1107          asl
1DE3: 0A       >1108          asl
1DE4: 0A       >1109          asl
1DE5: 0A       >1110          asl
1DE6: 85 D1    >1111          sta   NN         ; N0
1DE8: A5 9A    >1112          lda   rC+VV      ; Nx (low digit)
1DEA: 4A       >1113          lsr
1DEB: 4A       >1114          lsr
1DEC: 4A       >1115          lsr
1DED: 4A       >1116          lsr              ; 0N
1DEE: 05 D1    >1117          ora   NN
1DF0: 85 D1    >1118          sta   NN         ; 'NN' = BCD NN
1DF2: 38       >1119          sec
1DF3: F9 FB 1D >1120          sbc   bcdcor,y   ; Convert to binary.
1DF6: D0 02    >1121          bne   :ret
1DF8: A9 64    >1122          lda   #100       ; "00" ==> 100
1DFA: 60       >1123 :ret     rts
               >1124
1DFB: 00 06 0C >1125 bcdcor   db    0,6,12,18,24,30,36,42,48,54
```

===== Page 72 =====

```
                 64            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
1E05: 00 00 00 >6    rdroff   ds    2*3        ; 3 bytes * 2 unit (0..1)
1E0B: 00 00 00 >7    pchoff   ds    2*3        ; (Offsets are big-endian)
1E11: 00 00 00 >8    mtoff    ds    2*3        ; 3 bytes x 2 units (0..1)
1E17: 00 00    >9    mtlane   db    0,0        ; MT units lane state 0..1
                >10
                >11   IOstend  equ   *          ; End of I/O state.
                >12
1E19: 00 03 06 >13   fnxoff   db    0,3,6,9,12,12,15,15 ; fnx ==> offset
1E21: 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
1E29: A0       >19            db    $A0        ; $00 = Blank
1E2A: 00       >20            ds    1          ; $01 skip
1E2B: 00       >21            db    $00        ; $02 = Ignore
1E2C: AE A9    >22            asc   ".)"       ; $03..$04
1E2E: 00 00 00 >23            ds    11         ; $05..$0F skip
1E39: A8       >24            asc   "("        ; $10
1E3A: 00 00    >25            ds    2          ; $11..$12 skip
1E3C: AB AA    >26            asc   "+*"       ; $13..$14
1E3E: 8C       >27            db    $8C        ; $15 = Eject
1E3F: 8D       >28            db    $8D        ; $16 = CR
1E40: 00 00 00 >29            ds    3+6        ; $17..$1F skip
1E49: AD AF    >30            asc   "-/"       ; $20..$21
1E4B: 00       >31            ds    1          ; $22 skip
1E4C: AC       >32            asc   ","        ; $23
1E4D: A5       >33            asc   "%"        ; $24 (For SNAP CR translation)
1E4E: 00       >34            ds    1          ; $25 skip
1E4F: 89       >35            db    $89        ; $26 = TAB
1E50: A4       >36            asc   "$"        ; $27
1E51: 00 00 00 >37            ds    2+6+2      ; $28..$31 skip
1E5B: BF BD A7 >38            asc   "?='"      ; $32..$34
1E5E: 00 00 00 >39            ds    5+6+1      ; $35..$40 skip
1E6A: C1 C2 C3 >40            asc   "ABCDEFGHI" ; $41..$49
1E73: 00 00 00 >41            ds    6+1        ; $4A..$50 skip
1E7A: CA CB CC >42            asc   "JKLMNOPQR" ; $51..$59
1E83: 00 00 00 >43            ds    6+2        ; $5A..$61 skip
1E8B: D3 D4 D5 >44            asc   "STUVWXYZ" ; $62..$69
1E93: 00 00 00 >45            ds    6+16       ; $6A..$7F skip
1EA9: 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
1EB3: 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
1EB4: 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
1EB5: 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
1EB6: 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
1EB7: 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
1EB8: 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
1EB9: 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
1EBA: 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
1EBB: 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
1EBC: 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
1EBD: 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
1EBE: 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
```

```
1EBF: 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
1EC0: 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
1EC1: 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
1EC2: 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
1EC3: 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
1EC4: 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
1EC5: 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
1EC6: 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
1EC7: 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
1EC8: 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
1EC9: 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
1ECA: 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
1ECB: 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
1ECC: 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
1ECD: 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
1ECE: 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
1ECF: 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
1ED0: 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
1ED1: 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
1ED2: 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
1ED3: 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
1ED4: 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
1ED5: 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
1ED6: 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
1ED7: 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
1ED8: 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
1ED9: 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
1EDA: 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
1EDB: 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
1EDC: 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
1EDD: 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
1EDE: 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
1EDF: 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
1EE0: 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
1EE1: 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
1EE2: 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
1EE3: 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
1EE4: 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
1EE5: 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
1EE6: 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
1EE7: 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
1EE8: 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
1EE9: 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
1EEA: 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
1EEB: 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
1EEC: 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
1EED: 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
1EEE: 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
1EEF: 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
1EF0: 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
1EF1: 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
1EF2: 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
1EF3: 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
1EF4: 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
1EF5: 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
1EF6: 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
1EF7: 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
1EF8: 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
1EF9: 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
1EFA: 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
1EFB: 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
1EFC: 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
1EFD: 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
1EFE: 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
1EFF: 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
1F00: 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
1F01: 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
```

```
1F02: 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
1F03: 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
1F04: 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
1F05: 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
1F06: 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
1F07: 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
1F08: 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
1F09: 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
1F0A: 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
1F0B: 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
1F0C: 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
1F0D: 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
1F0E: 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
1F0F: 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
1F10: 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
1F11: 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
1F12: 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
1F13: 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
1F14: 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
1F15: 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
1F16: 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
1F17: 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
1F18: 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
1F19: 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
1F1A: 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
1F1B: 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
1F1C: 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
1F1D: 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
1F1E: 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
1F1F: 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
1F20: 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
1F21: 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
1F22: 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
1F23: 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
1F24: 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
1F25: 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
1F26: 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
1F27: 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
1F28: 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
1F29: 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
1F2A: 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
1F2B: 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
1F2C: 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
1F2D: 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
1F2E: 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
1F2F: 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
1F30: 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
1F31: 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
1F32: 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
1F33: 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
1F34: 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
1F35: 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
1F36: 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
1F37: 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
1F38: 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
1F39: 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
1F3A: 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
1F3B: 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
1F3C: 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
1F3D: 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
1F3E: 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
1F3F: 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
1F40: 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
1F41: 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
1F42: 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
1F43: 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
1F44: 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
```

```
1F45: 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
1F46: 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
1F47: 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
1F48: 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
1F49: 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
1F4A: 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
1F4B: 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
1F4C: 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
1F4D: 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
1F4E: 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
1F4F: 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
1F50: 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
1F51: 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
```

```
                              ===== Page 86 =====

1F52: 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
1F53: 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
1F54: 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
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: 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
1F58: 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
1F59: 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
1F5A: 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
1F5B: 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
1F5C: 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
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: 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
1F60: 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
1F61: 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
1F62: 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
1F63: 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
1F64: 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
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: 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
1F68: 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
1F69: 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
1F6A: 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
1F6B: 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
1F6C: 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
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: 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
1F70: 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
1F71: 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
1F72: 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
1F73: 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
1F74: 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
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: 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
1F77: 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
1F78: 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
1F79: 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
1F7A: FF       >70             db    $FF        ; Force overflow on undigits.
```

```
                              ===== Page 89 =====

               >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: 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
1F7C: 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
1F7D: 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
1F7E: 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
1F7F: 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
1F80: 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
1F81: 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
1F82: 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
1F83: 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
1F84: 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
1F85: 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
1F86: 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
1F87: 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
1F88: 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
1F89: 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
1F8A: 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
1F8B: 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
1F8C: 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
1F8D: 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
1F8E: 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
1F8F: 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
1F90: 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
1F91: 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
1F92: 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
1F93: 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
1F94: 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
```

```
1F95: 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
1F96: 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
1F97: 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
1F98: 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
1F99: 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
1F9A: 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
1F9B: 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
1F9C: 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
1F9D: 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
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: 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
1FA0: 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
1FA1: 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
1FA2: 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
1FA3: 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
1FA4: 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
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: 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
1FA8: 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
1FA9: 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
1FAA: 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
1FAB: 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
1FAC: 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
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: 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
1FB0: 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
1FB1: 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
1FB2: 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
1FB3: 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
1FB4: 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
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: 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
1FB8: 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
1FB9: 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
1FBA: 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
1FBB: 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
1FBC: 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
1FBD: 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
1FBE: 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
1FBF: 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
1FC0: 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
1FC1: 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
1FC2: 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
1FC3: 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
1FC4: 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
1FC5: 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
1FC6: 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
1FC7: 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
1FC8: 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
1FC9: 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
1FCA: 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
```

===== Page 95 =====

```
               >67   ]A0      equ   ]T*16      ; ]A0 = index w/ lo digit = 0
               >68   ]U       equ   ]Ax-]A0    ; BCD units digit
1FCB: 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
1FCC: 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
1FCD: 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
1FCE: 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
1FCF: 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
1FD0: 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
1FD1: 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
1FD2: 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
1FD3: 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
1FD4: 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
1FD5: 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
1FD6: 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
1FD7: 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
```

```
1FD8: 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
1FD9: 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
1FDA: 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
1FDB: 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
1FDC: 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
1FDD: 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
1FDE: 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
1FDF: 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
1FE0: 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
1FE1: 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
1FE2: 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
1FE3: 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
1FE4: 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
1FE5: 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
1FE6: 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
1FE7: 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
1FE8: 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
1FE9: 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
1FEA: 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
1FEB: 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
1FEC: 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
1FED: 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
1FEE: 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
1FEF: 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
1FF0: 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
1FF1: 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
1FF2: 00       >83            db    0
               >78   ]Ax      equ   *-BCDHadrl ; ]Ax = index of table entry
```

===== Page 98 =====

```
              >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: 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
1FF4: 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
1FF5: 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
1FF6: 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
1FF7: 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
1FF8: 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
1FF9: 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
1FFA: 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
1FFB: 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
1FFC: 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
1FFD: 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
1FFE: 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
1FFF: 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
2000: 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
2001: 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
2002: 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
2003: 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
2004: 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
2005: 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
2006: 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
2007: 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
2008: 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
2009: 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
200A: 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
200B: 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
200C: 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
200D: F0       >85            db    <]T*10+]U*600
```

```
                              ===== Page 100 =====

               >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
200E: 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
200F: 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
2010: 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
2011: 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
2012: 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
2013: 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
2014: 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
2015: 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
2016: 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
2017: 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
2018: 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
2019: 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
201A: 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
201B: 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
201C: 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
201D: 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
201E: 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
201F: 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
2020: 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
2021: 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
2022: 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
2023: 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
2024: 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
2025: 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
2026: 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
2027: 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
```

```
2028: 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
2029: 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
202A: 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
202B: 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
202C: 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
202D: 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
202E: 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
202F: 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
2030: 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
2031: 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
2032: 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
2033: 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
2034: 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
```

```
2035: 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
2036: 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
2037: 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
2038: 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
2039: 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
203A: 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
203B: 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
203C: 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
203D: 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
203E: 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
203F: 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
2040: 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
2041: 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
2042: 19       >98            db    >]T*10+]U*600
               >91   ]Ax      equ   *-BCDHadrh ; ]Ax = index of table entry
```

===== Page 104 =====

```
               >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: 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
2044: 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
2045: 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
2046: 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
2047: 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
2048: 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
2049: 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
204A: 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
204B: 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
204C: 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
204D: 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
204E: 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
204F: 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
2050: 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
2051: 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
2052: 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
2053: 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
2054: 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
2055: 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
2056: 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
2057: 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
2058: 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
2059: 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
205A: 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
205B: 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
205C: 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
205D: 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
205E: 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
205F: 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
2060: 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
2061: 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
2062: 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
2063: 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
2064: 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
2065: 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
2066: 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
2067: 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
2068: 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
2069: 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
206A: 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
206B: 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
206C: 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
206D: 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
206E: 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
206F: 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
2070: 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
2071: 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
2072: 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
2073: 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
2074: 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
2075: 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
2076: 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
2077: 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
```

```
                              ===== Page 108 =====

2078: 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
2079: 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
207A: 72       >98            db    >]T*10+]U*600
```

```
               **********************************************************
               *                                                        *
               *                      PUTPDCMD                          *
               *                                                        *
               * Append null-terminated string at (A,Y) onto IN,X.      *
               * Command is in hi-ASCII.                                *
               *                                                        *
               * Advances X, destroys A and Y.                          *
               *                                                        *
               **********************************************************

207B: 85 CC    putpdcmd sta   ptr        ; Set up string pointer
207D: 84 CD             sty   ptr+1
207F: A0 00             ldy   #0
2081: B1 CC    :cmdloop lda   (ptr),y    ; Append command string
2083: F0 07             beq   :rts       ;  until null
2085: 9D 00 02          sta   IN,x       ;   to keyboard buffer.
2088: E8                inx              ; Bump pointers.
2089: C8                iny
208A: D0 F5             bne   :cmdloop   ; (always)

208C: 60       :rts     rts              ; Return...

               **********************************************************
               *                                                        *
               *                      PDOSCMD                           *
               *                                                        *
               * Execute null-terminated ProDOS command at (A,Y)        *
               * Command is in hi-ASCII.                                *
               *                                                        *
               * Keyboard buffer, sptr, and Y are changed.              *
               * On error, C is set and A contains error code.          *
               *                                                        *
               **********************************************************

208D: A2 00    pdoscmd  ldx   #0         ; Empty kbd buffer.
208F: 20 7B 20          jsr   putpdcmd   ; Move in the command
                                         ;  and fall into pdosxeq.

               **********************************************************
               *                                                        *
               *                      PDOSXEQ                           *
               *                                                        *
               * Execute ProDOS command in keyboard buffer after        *
               * appending a carriage return.  Command is in hi-ASCII.  *
               *                                                        *
               * On error, C is set and A contains error code.          *
               *                                                        *
               **********************************************************

2092: A9 8D    pdosxeq  lda   #$8D       ; Carriage Return
2094: 9D 00 02          sta   IN,x       ;  at end
2097: AE 42 BE          ldx   BSSTATE    ; Save BASIC.SYSTEM
209A: 86 DB             stx   line8      ;  'state' var & set it
209C: A2 FF             ldx   #$FF       ;    to suppress blank
209E: 8E 42 BE          stx   BSSTATE    ;     line.
20A1: 20 03 BE          jsr   DOSCMD     ; Then do it...
20A4: A6 DB             ldx   line8      ; Restore BASIC.SYSTEM
20A6: 8E 42 BE          stx   BSSTATE    ;  state variable.
20A9: 60                rts

               simend   equ   *-1        ; End of B220SIM code
                        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
20AA: D0 D4 D2 >170  fnametbl asc   "PTRDR0",00 ; Moved to $300 by 'init'.
20B1: 00 00 00 >171           ds    18
20C3: D0 D4 D2 >172           asc   "PTRDR1",00
20CA: 00 00 00 >173           ds    18
20DC: D0 D4 D0 >174           asc   "PTPCH0",00
20E3: 00 00 00 >175           ds    18
20F5: D0 D4 D0 >176           asc   "PTPCH1",00
20FC: 00 00 00 >177           ds    18
210E: CD D4 D5 >178           asc   "MTU0L0",00
2115: 00 00 00 >179           ds    18
2127: CD D4 D5 >180           asc   "MTU0L1",00
212E: 00 00 00 >181           ds    18
2140: CD D4 D5 >182           asc   "MTU1L0",00
2147: 00 00 00 >183           ds    18
2159: CD D4 D5 >184           asc   "MTU1L1",00
2160: 00 00 00 >185           ds    18
               >186  fnend    equ   *


--End assembly, 6514 bytes, Errors: 0


Symbol table - alphabetical order:

   ADCYop  =$79        ADCZop  =$65        ADD     =$13A4      ADDR    =$04
   ADDRerr =$0C40      ADDRerrR=$0B48      ADL     =$1414      ALTCHAR =$C00F
   AR1     =$0700      AR2     =$0680      AR4     =$0600      AR8     =$0580
   ARbord  =$0DD4      ARmid   =$0DFA      ARv     =$0428      Aattr   =$0C9A
   Acol    =$05        Ain     =$091C      Alab    =$0583      Aparm   =$1264
?  B220SIM =$0800      B220col =$0C        B220end =$C7        B220msg =$0DBF
   B220strt=$90        BASCALC =$FBC1      BASL    =$28        BCDHadrh=$2031
   BCDHadrl=$1FE7      BCDLadrh=$1F4D      BCDLadrl=$1EB3      BCE     =$1A4B
   BCH     =$1A37      BCS     =$1AC3      BCSop   =$B0        BEEP    =$FBDD
   BFA     =$1A74      BFR     =$1A70      BITZop  =$24        BNEop   =$D0
   BOF     =$1A1A      BPC1    =$0728      BPC2    =$06A8      BPC4    =$0628
   BPC8    =$05A8      BPCbord =$0E20      BPCmid  =$0E46      BPCv    =$0450
   BRP     =$1A27      BSA     =$1A2D      BSSTATE =$BE42      BUN     =$1A5D
   Battr   =$0CCA      Bcol    =$05        Bin     =$0920      Blab    =$05AB
   Bmodmem =$1196      Bparm   =$126C      Bxxxx   =$125D      CAA     =$139D
   CAD     =$1384      CFA     =$1598      CH      =$24        CLA     =$1B9B
   CLCop   =$18        CLL     =$1BBC      CMPIop  =$C9        COMP    =$C2
   COMPcol =$19        COUT    =$FDED      CROUT   =$FD8E      CSU     =$136F
   CSW     =$B6        Cattr   =$0CEA      Ccol    =$15        Cin     =$0924
   Clab    =$05BB      DBB     =$1A07      DFL     =$18D2      DIV     =$14D3
   DLB     =$18E2      DOSCMD  =$BE03      DOSCON  =$03D0      ERR     =$C1
   ERRcol  =$15        ERRlab  =$0567      EXP     =$01        EXT     =$1570
   FAD     =$1656      FDV     =$1809      FIELDerr=$0C3C      FMU     =$1778
   FSU     =$1763      HLT     =$1118      HOME    =$FC58      Help1   =$0E93
?  Help2   =$0EB8   ?  Help3   =$0EDE   ?  Help4   =$0F01      IBB     =$19F4
   IFL     =$188C      IN      =$0200      INDshow =$0FCC      IOerr   =$0C44
   IOstate =$1E05      IOstend =$1E19      KAD     =$080D      KBD     =$C000
   KBSTROBE=$C010      LDB     =$1B57      LDR     =$1B4B      LSA     =$1B7D
   Lparm   =$1268      MANT    =$02        MEM     =$20D0      MIB     =$1D12
   MIR     =$1CB8      MIW     =$1CB0      MOR     =$1CC5      MOW     =$1CBB
   MPF     =$1CC8      MRD     =$1CA2      MRR     =$1CAD      MTC     =$1C9F
   MTS     =$1C6D      MUL     =$144C      NN      =$D1        NOP     =$1118
   NOPop   =$EA        OFLcol  =$1F        OFLerr  =$0C38      OP      =$03
   OPerr   =$0C34      Ov      =$C3        OvHlt   =$C7        PRB     =$1121
   PRBL2   =$F94A      PRD     =$111B      PRI     =$11B8   ?  PRINTERR=$BE0C
   PWI     =$12E8      PWR     =$11BB      Pattr   =$0CDA      Pcol    =$0D
   Pin     =$0928      Plab    =$05B3   ?  RESTART =$0803      RND     =$154E
   RPTcol  =$22        RTF     =$198E      RUN     =$C0        RUNcol  =$11
   Rattr   =$0CB2      Rcol    =$17        Rin     =$092C      Rlab    =$0595
```

===== Page 111 =====

```
   Rp      =$C4        S       =$00        SBCYop  =$F9        SBCZop  =$E5
   SECop   =$38        SLA     =$1BFC      SOR     =$1AD0      SPKR    =$C030
   SPO     =$12EB      SRA     =$1BC7      STA     =$1AE4      STAT    =$0E6C
   STATlin =$0550      STP     =$1B86      SUB     =$1436      SW1col  =$06
   SWlab   =$0553   ?  TABV    =$FB5B      UNDIGerR=$0B4B      UNDIGerr=$0C4A
   VV      =$02        WNDTOP  =$22     V  ]A0     =$40     V  ]Ax     =$49
V? ]Ov     =$1983   V  ]T      =$04     V  ]U      =$09     V? ]adc    =$193A
V  ]add    =$13B6   V? ]bfr    =$1A76   V? ]clc    =$192B   V? ]cmp    =$193C
V? ]contin =$0BC1   V? ]dfl    =$18F7   V  ]err    =$0C4C   V  ]errpt  =$18DF
V  ]fad    =$1664   V  ]fetch1 =$1B98   V  ]fetch2 =$1B2E   V? ]fetch3 =$1A6D
V  ]fetch4 =$18CF   V  ]keep   =$1D     V  ]kend   =$1D     V? ]nop    =$1964
V? ]prd    =$1135   V  ]stop   =$080D   V? ]sub    =$1967      b220asc =$1E29
   bcdcor  =$1DFB      beepget =$0954      blanklin=$0D8D      blkcnt  =$D7
   changed =$D9        clear   =$1DD0      clearAR =$17FE      cmdfnx  =$D8
   compare =$15B8      cursor  =$57        delete  =$FF        disARmid=$0D95
   disBPCbo=$0DA3      disBPCmi=$0DB1      disiocfg=$0A1E      dispA   =$0F45
   dispB   =$0F53      dispC   =$0F61      dispP   =$0F5A      dispR   =$0F4C
   dispSTAT=$0F68      dispcnt =$64        dispctr =$D2        dispdig =$1033
   disphelp=$0F22      display =$0F33      disppanl=$0D04      dispreg =$0FF5
   divide  =$14D9      dnarrow =$8A        doio    =$11FD      ediocfg =$0A15
   escape  =$9B        exchAR  =$1DAE      execute =$0B93      fetch   =$0B72
   fnamecol=$0C        fnames  =$0300      fnametbl=$20AA      fnend   =$2172
   fnx     =$D3        fnxfn   =$1E21      fnxoff  =$1E19      getMTt1 =$1270
   getdig  =$0957      getfnx  =$11F7      getfnxt1=$1274      incP    =$0C14
   incmem  =$11AC      incoff  =$12CF      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    =$1255      loadrA  =$138E      ltarrow =$88
   memb    =$7530      memptr  =$CA        midNN   =$1DDD      mtlane  =$1E17
?  mtoff   =$1E11      mtrw    =$1D23      mtwrite =$1D1E      multiply=$1452
   newP    =$0B54      newp    =$C5        noAD    =$8000      off     =$A0
   on      =$AA        operr   =$8C34      optabh  =$10BE      optabl  =$1064
?  pchoff  =$1E0B   ?  pdoscmd =$208D      pdosxeq =$2092      ptr     =$CC
   ptrdwrt =$11D3      ptread  =$11C6      ptwrite =$11CE      putbyte =$129F
   puthx   =$129B      putoff  =$12B7      putpdcmd=$207B      rA      =$9E
   rB      =$94        rBx     =$90        rC      =$98        rD      =$AA
   rD10    =$B0        rP      =$96        rR      =$A4        rdroff  =$1E05
   reset   =$0C7C   MD resi    =$8000      restart =$0C91      rtmargin=$04
   sL      =$01        save    =$1259      savex   =$D6        selBASL =$DB
   selch   =$D7        selected=$D4        selsave =$D5     MD seti    =$8000
   setread =$11DB      setwrite=$11EA      shleft1 =$0930      signtbl =$1588
   simend  =$20A9      skipincP=$C6        slA     =$1DA3      slT     =$1D99
   splitsL =$1DBC      srA     =$1D6A      srAM    =$1D6C      srAMR   =$1D77
   srAS    =$1D68   ?  srR     =$1D7A      srT     =$1D75      srT2    =$1D85
   stopR   =$0B51      t1      =$D0        tabs    =$136A      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     V  ]keep   =$1D
V  ]kend   =$1D        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      KAD     =$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=$0F22
   display =$0F33      dispA   =$0F45      dispR   =$0F4C      dispB   =$0F53
   dispP   =$0F5A      dispC   =$0F61      dispSTAT=$0F68      INDshow =$0FCC
   dispreg =$0FF5      dispdig =$1033      optabl  =$1064      optabh  =$10BE
   HLT     =$1118      NOP     =$1118      PRD     =$111B      PRB     =$1121
V? ]prd    =$1135      Bmodmem =$1196      incmem  =$11AC      PRI     =$11B8
   PWR     =$11BB      ptread  =$11C6      ptwrite =$11CE      ptrdwrt =$11D3
   setread =$11DB      setwrite=$11EA      getfnx  =$11F7      doio    =$11FD
   load    =$1255      save    =$1259      Bxxxx   =$125D      Aparm   =$1264
   Lparm   =$1268      Bparm   =$126C      getMTt1 =$1270      getfnxt1=$1274
   puthx   =$129B      putbyte =$129F      putoff  =$12B7      incoff  =$12CF
   PWI     =$12E8      SPO     =$12EB      tabs    =$136A      CSU     =$136F
   CAD     =$1384      loadrA  =$138E      CAA     =$139D      ADD     =$13A4
V  ]add    =$13B6      ADL     =$1414      SUB     =$1436      MUL     =$144C
   multiply=$1452      DIV     =$14D3      divide  =$14D9      RND     =$154E
   EXT     =$1570      signtbl =$1588      CFA     =$1598      compare =$15B8
   FAD     =$1656   V  ]fad    =$1664      FSU     =$1763      FMU     =$1778
   clearAR =$17FE      FDV     =$1809      IFL     =$188C   V  ]fetch4 =$18CF
   DFL     =$18D2   V  ]errpt  =$18DF      DLB     =$18E2   V? ]dfl    =$18F7
V? ]clc    =$192B   V? ]adc    =$193A   V? ]cmp    =$193C   V? ]nop    =$1964
V? ]sub    =$1967   V? ]Ov     =$1983      RTF     =$198E      IBB     =$19F4
   DBB     =$1A07      BOF     =$1A1A      BRP     =$1A27      BSA     =$1A2D
   BCH     =$1A37      BCE     =$1A4B      BUN     =$1A5D   V? ]fetch3 =$1A6D
   BFR     =$1A70      BFA     =$1A74   V? ]bfr    =$1A76      BCS     =$1AC3
   SOR     =$1AD0      STA     =$1AE4   V  ]fetch2 =$1B2E      LDR     =$1B4B
   LDB     =$1B57      LSA     =$1B7D      STP     =$1B86   V  ]fetch1 =$1B98
   CLA     =$1B9B      CLL     =$1BBC      SRA     =$1BC7      SLA     =$1BFC
   MTS     =$1C6D      MTC     =$1C9F      MRD     =$1CA2      MRR     =$1CAD
   MIW     =$1CB0      MIR     =$1CB8      MOW     =$1CBB      MOR     =$1CC5
   MPF     =$1CC8      MIB     =$1D12      mtwrite =$1D1E      mtrw    =$1D23
   srAS    =$1D68      srA     =$1D6A      srAM    =$1D6C      srT     =$1D75
   srAMR   =$1D77   ?  srR     =$1D7A      srT2    =$1D85      slT     =$1D99
   slA     =$1DA3      exchAR  =$1DAE      splitsL =$1DBC      clear   =$1DD0
   midNN   =$1DDD      bcdcor  =$1DFB      IOstate =$1E05      rdroff  =$1E05
?  pchoff  =$1E0B   ?  mtoff   =$1E11      mtlane  =$1E17      IOstend =$1E19
   fnxoff  =$1E19      fnxfn   =$1E21      b220asc =$1E29      BCDLadrl=$1EB3
   BCDLadrh=$1F4D      BCDHadrl=$1FE7      BCDHadrh=$2031      putpdcmd=$207B
?  pdoscmd =$208D      pdosxeq =$2092      simend  =$20A9      fnametbl=$20AA
   MEM     =$20D0      fnend   =$2172      memb    =$7530      noAD    =$8000
   operr   =$8C34   MD resi    =$8000   MD seti    =$8000      DOSCMD  =$BE03
?  PRINTERR=$BE0C      BSSTATE =$BE42      KBD     =$C000      ALTCHAR =$C00F
```

===== Page 113 =====

```
KBSTROBE=$C010      SPKR    =$C030      PRBL2   =$F94A   ?  TABV    =$FB5B
BASCALC =$FBC1      BEEP    =$FBDD      HOME    =$FC58      CROUT   =$FD8E
COUT    =$FDED
```