

```

CE24:D0 11 CE37 20      BNE  INVX      ;=>cursor off, don't invert
CE26:48      21 INVERT PHA          ;save AC
CE27:98      22      TYA            ; AND Y
CE28:48      23      PHA
CE29:AC 7B 05 24      LDY  OURCH      ;GET CH
CE2C:20 44 CE 25      JSR  PICK      ;GET CHARACTER
CE2F:49 80    26      EOR  #S80      ;FLIP INVERSE/NORMAL
CE31:20 70 CE 27      JSR  STORIT     ; ONTO SCREEN
CE34:68      28      PLA            ;RESTORE Y
CE35:A8      29      TAY            ; AND AC
CE36:68      30      PLA
CE37:60      31 INVX  RTS
CE38:      32 *****
CE38:      33 * NAME      : STORCHAR
CE38:      34 * FUNCTION: STORE A CHAR ON SCREEN
CE38:      35 * INPUT      : AC=CHAR
CE38:      36 *          : Y=CH POSITION
CE38:      37 * OUTPUT      : CHAR ON SCREEN
CE38:      38 * VOLATILE: NOTHING
CE38:      39 * CALLS      : SCREENIT
CE38:      40 *****
CE38:      41 *
CE38:      42 STORCHAR EQU *
CE38:48      43      PHA            ;SAVE AC
CE39:24 32    44      BIT  INVFLG     ;NORMAL OR INVERSE?
CE3B:30 02 CE3F 45      BMI  STOR2     ;=>NORMAL
CE3D:29 7F    46      AND  #S7F      ;inverse it
CE3F:      CE3F 47 STOR2 EQU *
CE3F:20 70 CE 48      JSR  STORIT     ;=>do it!!
CE42:68      49      PLA            ;RESTORE AC
CE43:60      50 SEV   RTS
CE44:      51 *****
CE44:      52 * NAME      : PICK
CE44:      53 * FUNCTION: GET A CHAR FROM SCREEN
CE44:      54 * INPUT      : Y=CH POSITION
CE44:      55 * OUTPUT      : AC=CHARACTER
CE44:      56 * VOLATILE: NOTHING
CE44:      57 * CALLS      : SCREENIT
CE44:      58 *****
CE44:      59 *
CE44:B1 28    60 PICK  LDA  (BASL),Y ;get 40 column character
CE46:2C 1F CO 61      BIT  RD80VID    ;80 columns?
CE49:10 19 CE64 62      BPL  PICK3     ;=>no, do text shift
CE4B:8D 01 CO 63      STA  SET80COL    ;force 80STORE for 80 columns
CE4E:84 2A    64      STY  BAS2L      ;temp store for position
CE50:98      65      TYA            ;divide CH by two
CE51:45 20    66      EOR  WNDLFT     ;C=1 if char in main RAM
CE53:6A      67      ROR  A          ;get low bit into carry
CE54:B0 04 CE5A 68      BCS  PICK1     ;=>store in main memory
CE56:AD 55 CO 69      LDA  TXTPAGE2   ;else switch in page 2
CE59:C8      70      INY            ;for odd left, aux bytes
CE5A:98      71 PICK1 TYA            ;divide position by 2
CE5B:4A      72      LSR  A          ;and use carry as
CE5C:A8      73      TAY            ;page indicator

```

```

CDD6:98      351     TYA            ;div 2 for 80 column index
CDD7:4A      352     LSR  A
CDD8:B0 03 CDDD 353     BCS  SCR7      ;save on page1
CDDA:8D 55 CO 354     STA  TXTPAGE2
CDDD:A8      355 SCR7 TAY            ;get 80 column index
CDDF:68      356     PLA            ;now save character
CDDF:91 28    357     STA  (BASL),Y
CDE1:8D 54 CO 358     STA  TXTPAGE1   ;flip page1
CDE4:A4 2A    359     LDY  BAS2L      ;restore 40 column index
CDE6:C8      360     INY            ;move to the right
CDE7:CO 28    361     CPY  #40        ;at right yet?
CDE9:90 E6 CDD1 362     BCC  SCR6     ;=>no, do next column
CDEB:20 80 CC 363     JSR  CLRHAF     ;clear half of screen
CDEE:CA      364     DEX            ;else do next line of screen
CDEF:30 04 CDF5 365     BMI  SCR9     ;=>done with top line
CDF1:E4 22    366     CPX  WNDTOP    ;at top yet?
CDF3:B0 D3 CDC8 367     BCS  SCR5
CDF5:8D 0D CO 368 SCR9 STA  SET80VID ;convert to 80 columns
CDF8:20 FE CD 369 SCRNET JSR  VTAB   ;update base
CDFB:68      370     PLA            ;restore X
CDFC:AA      371     TAX
CDFD:60      372     RTS
CDFE:      373 *
CDFE:A5 25    374 VTAB LDA  CV          ;get 80 column CV
CE00:8D FB 05 375     STA  OURCV     ;copy to OURCV
CE03:20 BA CA 376 VTABZ JSR  BASCALC   ;calc base address
CE06:A5 20    377     LDA  WNDLFT     ;and add window left to it
CE08:2C 1F CO 378     BIT  RD80VID   ;is it 80 columns?
CE0B:10 01 CEOE 379     BPL  VTAB40   ;window width ok
CE0D:4A      380     LSR  A          ;else divide width by 2
CE0E:18      381 VTAB40 CLC
CE0F:65 28    382     ADC  BASL      ;prepare to add
CE11:85 28    383     STA  BASL      ;add in window left
CE13:60      384 VTABX RTS            ;and update base
CE14:      29      INCLUDE SUBS3
CE14:C9 E1    1 UPSHFT CMP  #SE1     ;is it lowercase?
CE16:90 06 CE1E 2      BCC  UPSHFT2   ;=>nope
CE18:C9 FB    3      CMP  #SFB      ;lowercase?
CE1A:B0 02 CE1E 4      BCS  UPSHFT2   ;=>nope
CE1C:29 DF    5      AND  #SDF      ;else upshift
CE1E:60      6 UPSHFT2 RTS
CE1F:      7 *
CE1F:      8 *****
CE1F:      9 * NAME      : INVERT
CE1F:     10 * FUNCTION: INVERT CHAR AT CH/CV
CE1F:     11 *          : Unless Pascal and M.CURSOR=1
CE1F:     12 * INPUT      : NOTHING
CE1F:     13 * OUTPUT      : CHAR AT CH/CV INVERTED
CE1F:     14 * VOLATILE: NOTHING
CE1F:     15 * CALLS      : PICK, STORCHAR
CE1F:     16 *****
CE1F:     17 *
CE1F:AD FB 04 18 PASINV LDA  MODE     ;check pascal cursor flag
CE22:29 10    19      AND  #M.CURSOR ;before displaying cursor

```