

HP-86/87 ASSEMBLER BREAKPOINT SUBROUTINE by David Efron

The following assembly language routine is meant to be used as a program development and debugging tool for HP-86/87 programmer's who don't own a systems monitor ROM. Since this is a subroutine, it doesn't have the "shell" that is necessary to create a full Series 80 Binary program. The comments in the code explain what it does and what the output should look like. After debugging of the binary program, references to this subroutine and the subroutine itself can be eliminated and the binary can be re-assembled into a "production" version.

```

9000 BRKPT   SAD                ! Save before we do anything
9005         PUMD R70,+R12      ! Save the 70's, 60's and 20's
9010         PUMD R60,+R12      ! because we have to use them
9015         LDM R70,R20
9020         PUMD R70,+R12
9025         BCD
9030         CLB R22            ! Shift E into a zero'd byte
9035         ELB R22            ! (four bits at once in BCD mode)
9040         BIN
9045         PUBD R22,+R6       ! Put E onto stack
9050         PUMD R70,+R6       ! Save the 70's again, so we can use them
9055         LDM R22,=BRKTABLE  ! We need R22-23; they're clobbered,
9060         STM R22,R75        ! but we save them above.
9065         CLB R77            ! Set up for PTR2 stores
9070         ADMD R75,=BINTAB   ! (The absolute address of our save area)
9075         STMD R75,=PTR2
9080         POMD R70,-R6       ! Now . . . Save the 70's in the save area
9085         STMI R70,=PTR2-
9090         STMI R60,=PTR2-   ! And the 60's
9095         STMI R50,=PTR2-
9100         STMI R40,=PTR2-   ! The 50's and 40's
9105         LDM R70,R30
9110         STMI R70,=PTR2-   ! For the 30's on down to the 0's, first
9115         POMD R70,-R12      ! move them to the 70's, EIGHT registers
9120         PUMD R70,+R12      ! at a time.
9125         STMI R70,=PTR2-   ! The 20's came off the stack, but went
9130         LDM R70,R10        ! back on for later restoration
9135         STMI R70,=PTR2-
9140         LDM R70,R0
9145         STMI R70,=PTR2-
9150         CLM R60            ! Zero out some space
9155         POBD R67,-R6       ! Get the E register back
9160         POMD R75,-R6       ! Get back the SAD (arp,drp,status bits)
9165         LDM R20,=77,77     ! Mask for the arp and drp
9170         LDM R22,R75        ! R75-76 have the ARP and DRP
9175         ANM R22,R20
9180         STB R22,R66       ! Save them
9185         STB R23,R65       ! in reverse order for the screen display
9190         STMI R60,=PTR2-
9195         CLM R60
9200         LDB R22,=1         ! To invert LZ,ZR,RZ (SAD inverted them)
9205         STB R75,R23       ! Move the ARP part of the SAD to get the

```

```

9210      ELB R23      ! OVF and CY bits
9215      ELB R60      ! Shift each one into the CY, then
9220      ELB R23      ! into one of the 60's, to end up with
9225      ELB R61      ! eight bytes of 0 or 1
9230      STB R77,R23  ! FOR NG,LZ,ZR,RZ,OD
9235      ELB R23
9240      ELB R62      ! . . . do the same thing
9245      ELB R23      !
9250      ELB R63      ! ----- S t a t u s   F l a g s -----
9255      XRB R63,R22  !      OV  CY  NG  LZ  ZR  RZ  OD  DC
9260      ELB R23      !      ---  ---  ---  ---  ---  ---  DRP  ARP  E
9265      ELB R23      ! R00:000 000 000 000 000 000 000 000
9270      ELB R23      ! R10:000 000                                000 000
9275      ELB R23      ! R20:000 000 This is what it 000 000
9280      ELB R64      ! R30:000 000 will look like. 000 000
9285      XRB R64,R22  ! R40: .   .   .   .   .   .   .   .
9290      ELB R23      ! R50: .   .   .   .   .   .   .   .
9295      ELB R65      ! R60: .   .   .   .   .   .   .   .
9300      XRB R65,R22  ! R70: .   .   .   .   .   .   .   .
9305      ELB R23      ! R80:000 000 000 000 000 000 000 000
9310      ELB R66      ! -----
9315      STB R76,R23  ! FOR DCM
9320      ELB R23      ! get passed the OVF
9325      ELB R23
9330      ELB R67
9335      STMI R60,=PTR2- ! Save the 8 status bits
9340      LDMD R75,=PTR2- ! Save start-of-table in 100470 (octal)
9345      STMD R75,=100470 ! *** Use MEMD 100470,120 to look at
9350 !                    the dump of registers and flags
9355      POMD R70,-R12 ! Restore the registers we used
9360      STM R70,R20
9365      POMD R60,-R12
9370      POMD R70,-R12
9375      RTN
9380 ! ===== Use the subroutine as follows: JSB X22,BRKPT ! where R22-23 has
9385 ! ===== the BINTAB address. Then : RTN ! or continue with
9390 ! ===== the program
9395 ! ===== Finally, use the command MEMD 100470,120
9399 ! =====
9400 BRKDUMP BSZ 120
9405 BRKTABLE BSZ 0
9999 ! =====
10000 BINTAB DAD 104070
10010 PTR2 DAD 177714
10020 PTR2- DAD 177715
20000 FIN

```