6502 DISASSEMBLER

by ALLEN BAUM &
STEPHEN WOZNIAK
Apple Computer Co.
Palo Alto, CA.

Description

This subroutine package is used to display single or sequential 6502 instructions in mnemonic form. The subroutines are tailored to disassemblers and debugging aids, but tables with more general usage (assemblers) are included. The subroutines occupy one page (256 bytes) and tables most of another. Seven page zero locations are used.

Features

Four output fields are generated for each disassembled instruction: (1) Address of instruction, in hexadecimal (hex); (2) Hex code listing of instruction, 1 to 3 bytes; (3) 3-character mnemonic, or "???" for invalid ops (which assume a length of byte); and (4) Address field, in one of the following formats.

Format	Address Mode
(empty)	Invalid, Implied, Accumulator.
$12	Page zero.
$1234	Absolute, Branch (*target* printed).
#$12	Immediate
$12, X	Zero page, indexed by X.
$12, Y	Zero page, indexed by Y.
$1234, X	Absolute, indexed by X.
$1234, Y	Absolute, indexed by Y.
($1234)	Indirect
($12, X)	Indexed Indirect.
($12), Y	Indirect Indexed.

Note that unlike MOS TECHNOLOGY assemblers, which use "A" for accumulator addressing, the APPLE disassembler outputs an empty field to avoid confusion and facilitate byte counting.

Usage

The following subroutine entries are useful:

(a) DSMBL: Disassembles and displays 20 sequential instructions beginning at the address specified by the page zero variables PCL and PCH. For example, if called with $D2 in PCL and $38 in PCH, 20 instructions beginning at address $38D2 will be disassembled. PCL and PCH are updated to contain the address of the last disassembled instruction. Must be called with 6502 in hexadecimal mode ('D' status bit clear). All processor registers are altered (except S—stack pointer). Uses INSTDSP and PCADJ.

(b) INSTDSP: Disassembles and displays a single instruction whose address is specified by PCL and PCH. Must be called in hexadecimal mode. All processor registers (except S) are altered. Uses PCADJ3, PRPC, PRBLNK, PRBL2, PRNTAX, PRBYTE, and CHAROUT.

(c) PRPC: Outputs a carriage return, 4 hex digits corresponding to PCH and PCL, a dash, and 3 blanks. Alters A, clears X. Uses PRNTAX and CHAROUT.

(d) PRNTX: Outputs the contents of X as two hex digits. Alters A. Uses CHAROUT.

(e) PRNTAX: Outputs two hex digits for the contents of A, then two hex digits for the contents of X. A is altered. Uses CHAROUT.

(f) PRNTYX: Same as PRNTAX except that Y and X are output. Alters A. Uses CHAROUT.

(g) PRBLNK: Outputs 3 blanks. Alters A, clears X. Uses CHAROUT.

(h) PRBL2: Outputs the number of blanks specified by the contents of X (0 for 256 blanks). Alters A, clears X. Uses CHAROUT.

(i) PRBL3: Outputs a character from the A register followed by X-1 blanks. In other words, X specifies the total number of characters output. (0 for 256 blanks). Alters A, clears X. Uses CHAROUT.

(j) PCADJ: (PCL, PCH) + 1 + (contents of page zero variable LENGTH) →Y & A (low order byte in Y). For example, if PCL = $D2, PCH = $38, and LENGTH = 1 (corresponding to a 2 byte instruction), PCADJ will leave Y = $D4 and A = $38. X is always loaded with PCH.

(k) PCADJ2: Same as PCADJ except that A is used in place of LENGTH.

(l) PCADJ3: Same as PCADJ2 except that the increment (+1) is specified by the carry (set = +1, clear = +0).

Running as a Program

The following program will run a disassembly.

```
9F0      20     0      8       JSR DSMBL
9F3      4C     1F     FF      JMP MONITOR
```

$\underbrace{\qquad\qquad\qquad}_{\text{Supplied on APPLE-1 cassette tapes.}}$

First, put the starting address of code you want disassembled in PCL (low order byte) and PCH (high order byte). Then type 9F0 R CR (on APPLE-1 system). 20 instructions will be disassembled. Hitting R CR again will give the next 20, etc.

Cassette tapes supplied for the ACI-1 (APPLE Cassette Interface) are intended to be loaded from $800 to $9FF.

Non-APPLE Systems

Source and object code supplied occupies page 8 and 9. All code is on page 8, tables on page 9. These tables may be relocated at will: MODE, MODE2, CHAR1, CHAR2, MNEML, and MNEMR. The code may also be relocated. Be careful if you use pages 0 or 1. Page 1 is the subroutine return stack and page 0 must contain 7 variables (to use DSMBL). These may be relocated on page 0 but PCL must always immediately precede PCH for (Z-page) Y addressing.

```
          ⎧$40    FORMAT ⎫
locations ⎪$41    LENGTH ⎬ Used by INSTDSP,
used      ⎪$42    LMNEM  ⎪ DSMBL
by        ⎨$43    RMNEM  ⎭
supplied  ⎪$44    PCL    ⎫ Used by PCADJ,
code      ⎪$45    PCH    ⎭ INSTDSP, DSMBL
          ⎩$46    COUNT    Used by DSMBL only
```

Modifications

(a) To change '#' to '=' for immediate mode change location $955 (on code enclosed) from a $A3 to a $BD.

(b) To skip the '$' (meaning hex) preceding disassembled values make the following changes:

```
946: 01   (was 81)
947: 02   (was 82)
94C: 11   (was 91)
94D: 12   (was 92)
94E: 06   (was 86)
95C: 05   (was 85)
951: 1D   (was 9D)
95B: 00   (was A4)
95C: 00   (was A4)
```

(c) To have address field of accumulator-addressed instructions print as 'A'.

(1) Must skip $ preceding disassembled values by making modification (b) above.
(2) Change the following locations.

```
949: 80   (was 00)
957: C1   (was A4)
```

(d) To add ROR and addressing modes change the following locations:

```
991: 9C    (was 00)     919: 02    (was 00)
9D1: 26    (was 00)     91A: 45    (was 40)
                        91B: B3    (was B0)
                        91D: 08    (was 00)
                        91F: 09    (was 00)
```

```
ASM65 ASSEMBLER   REV 1.4   8/11/76


                     XREF
              FORMAT EQU  $40
              LENGTH EQU  $41
              LMNEM  EQU  $42
              RMNEM  EQU  $43
              PCL    EQU  $44
              PCH    EQU  $45
              COUNT  EQU  $46
              PRBYTE EQU  $FFDC
              CHAROU EQU  $FFEF
                     ORG  $800
0800 A9 13    DSMBL  LDA  #$13        COUNT FOR 20 INSTR DSMBLY.
0802 85 46           STA  COUNT
0804 20 12 08 DSMBL2 JSR  INSTDSP     DISASSEMBLE AND DISPLAY INSTR.
0807 20 EF 08        JSR  PCADJ
080A 85 44           STA  PCL         UPDATE PCL,H TO NEXT INSTR.
080C 84 45           STY  PCH
080E C6 46           DEC  COUNT       DONE FIRST 19 INSTRS.?
0810 D0 F2           BNE  DSMBL2      * YES, LOOP.  ELSE DSMBL 20TH.
0812 20 D3 08 INSTDS JSR  PRPC        PRINT PCL,H.
0815 A1 44           LDA  (PCL,X)     GET OP CODE.
0817 A8              TAY
0818 4A              LSR  A           * EVEN/ODD TEST.
0819 90 0B           BCC  IEVEN
081B 4A              LSR  A           *  TEST B1.
081C B0 17           BCS  ERR         *  XXXXXX11 INSTR INVALID.
081E C9 22           CMP  #$22
0820 F0 13           BEQ  ERR         *  10001001 INSTR INVALID.
0822 29 07           AND  #$7         MASK 3 BITS FOR ADDRESS MODE &
0824 09 80           ORA  #$80        *  ADD INDEXING OFFSET.
0826 4A       IEVEN  LSR  A           * LSB INTO CARRY FOR
0827 AA              TAX              * LEFT/RIGHT TEST BELOW.
0828 BD 00 09        LDA  MODE,X      INDEX INTO ADDRESS MODE TABL.
082B B0 04           BCS  RTMODE      IF CARRY SET USE LSD FOR
082D 4A              LSR  A           * PRINT FORMAT INDEX.
082E 4A              LSR  A
082F 4A              LSR  A           * IF CARRY CLEAR USE MSD.
0830 4A              LSR  A
0831 29 0F    RTMODE AND  #$F         MASK FOR 4-BIT INDEX.
0833 D0 04           BNE  GETFMT      $0 FOR INVALID OPCODES.
0835 A0 80    ERR    LDY  #$80        SUBSTITUTE $80 FOR INVALID OP,
0837 A9 00           LDA  #$0         SET PRINT FORMAT INDEX TO 0.
0839 AA       GETFMT TAX
083A BD 44 09        LDA  MODE2,X     INDEX INTO PRINT FORMAT TABLE.
083D 85 40           STA  FORMAT      SAVE FOR ADDRESS FIELD FORMAT.
083F 29 03           AND  #$3         MASK 2-BIT LENGTH.  0=1-BYTE,
0841 85 41           STA  LENGTH      *  1=2-BYTE, 2=3-BYTE.
0843 98              TYA              * OP CODE.
0844 29 8F           AND  #$8F        MASK IT FOR 1XXX1010 TEST.
0846 AA              TAX              * SAVE IT.
0847 98              TYA              * OP CODE TO A AGAIN.
0848 A0 03           LDY  #$3
084A E0 8A           CPX  #$8A
084C F0 0B           BEQ  MNNDX3
084E 4A       MNNDX1 LSR  A
084F 90 08           BCC  MNNDX3      FORM INDEX INTO MNEMONIC TABL.
0851 4A              LSR  A
0852 4A       MNNDX2 LSR  A           * 1XXX1010 -> 00101XXX
0853 09 20           ORA  #$20        * XXXYYY01 -> 00111XXX
0855 88              DEY              * XXXYYY10 -> 00110XXX
0856 D0 FA           BNE  MNNDX2      * XXXYY100 -> 00100XXX
0858 C8              INY              * XXXXX000 -> 000XXXXX
0859 88       MNNDX3 DEY
085A D0 F2           BNE  MNNDX1
085C 48              PHA              * SAVE MNEMONIC TABLE INDEX.
```

```
085D B1 44     PROP   LDA  (PCL),Y
085F 20 DC FF         JSR  PRBYTE
0862 A2 01            LDX  #$1
0864 20 E6 08  PROPBL JSR  PRBL2
0867 C4 41            CPY  LENGTH        PRINT INSTR (1 TO 3 BYTES)
0869 C8               INY                *  IN A 12-CHARACTER FIELD.
086A 90 F1            BCC  PROP
086C A2 03            LDX  #$3           CHAR COUNT FOR MNEMONIC PRINT.
086E C0 04            CPY  #$4
0870 90 F2            BCC  PROPBL
0872 68               PLA                * RECOVER MNEMONIC INDEX.
0873 A8               TAY
0874 B9 5E 09         LDA  MNEML,Y
0877 85 42            STA  LMNEM         FETCH 3-CHAR MNEMONIC.
0879 B9 9E 09         LDA  MNEMR,Y       *  (PACKED IN 2 BYTES)
087C 85 43            STA  RMNEM
087E A9 00     PRMN1  LDA  #$0
0880 A0 05            LDY  #$5
0882 06 43     PRMN2  ASL  RMNEM
0884 26 42            ROL  LMNEM         SHIFT 5 BITS OF CHAR INTO A.
0886 2A               ROL  A             *   (CLEARS CARRY)
0887 88               DEY
0888 D0 F8            BNE  PRMN2
088A 69 BF            ADC  #$BF          ADD '?' OFFSET.
088C 20 EF FF         JSR  CHAROUT       OUTPUT A CHARACTER OF MNEMONIC
088F CA               DEX
0890 D0 EC            BNE  PRMN1
0892 20 E4 08         JSR  PRBLNK        OUTPUT 3 BLANKS.
0895 A2 06            LDX  #$6           COUNT FOR 6 PRINT FORMAT BITS.
0897 E0 03     PRADR1 CPX  #$3
0899 D0 12            BNE  PRADR3        IF X=3 THEN PRINT ADDRESS VAL.
089B A4 41            LDY  LENGTH
089D F0 0E            BEQ  PRADR3        NO PRINT IF LENGTH=0.
089F A5 40     PRADR2 LDA  FORMAT
08A1 C9 E8            CMP  #$E8          HANDLE REL ADDRESSING MODE
08A3 B1 44            LDA  (PCL),Y       SPECIAL (PRINT TARGET ADR)
08A5 B0 1C            BCS  RELADR        *   (NOT DISPLACEMENT)
08A7 20 DC FF         JSR  PRBYTE        OUTPUT 1- OR 2-BYTE ADDRESS.
08AA 88               DEY                *  MORE SIGNIFICANT BYTE FIRST
08AB D0 F2            BNE  PRADR2
08AD 06 40     PRADR3 ASL  FORMAT        TEST NEXT PRINT FORMAT BIT.
08AF 90 0E            BCC  PRADR4        IF 0, DON'T PRINT
08B1 BD 51 09         LDA  CHAR1-1,X     *  CORRESPONDING CHARS.
08B4 20 EF FF         JSR  CHAROUT       OUTPUT 1 OR 2 CHARS.
08B7 BD 57 09         LDA  CHAR2-1,X     *  (IF CHAR FROM CHAR2 IS 0,
08BA F0 03            BEQ  PRADR4        *     DON'T OUTPUT IT)
08BC 20 EF FF         JSR  CHAROUT
08BF CA        PRADR4 DEX
08C0 D0 D5            BNE  PRADR1
08C2 60               RTS                *RETURN IF DONE 6 FORMAT BITS.
08C3 20 F2 08  RELADR JSR  PCADJ3        PCL,H + DISPL + 1 TO A,Y.
08C6 AA               TAX
08C7 E8               INX
08C8 D0 01            BNE  PRNTYX        *     +1 TO X,Y.
08CA C8               INY
08CB 98        PRNTYX TYA
08CC 20 DC FF  PRNTAX JSR  PRBYTE        PRINT TARGET ADR OF BRANCH
08CF 8A        PRNTX  TXA                *  AND RETURN
08D0 4C DC FF         JMP  PRBYTE
08D3 A9 8D     PRPC   LDA  #$8D
08D5 20 EF FF         JSR  CHAROUT       OUTPUT CARRIAGE RETURN.
08D8 A5 45            LDA  PCH
08DA A6 44            LDX  PCL
08DC 20 CC 08         JSR  PRNTAX        OUTPUT PCH AND PCL.
08DF A9 AD            LDA  #$AD
08E1 20 EF FF         JSR  CHAROUT       OUTPUT '-'
08E4 A2 03     PRBLNK LDX  #$3           BLANK COUNT.
08E6 A9 A0     PRBL2  LDA  #$A0
08E8 20 EF FF  PRBL3  JSR  CHAROUT       OUTPUT A BLANK.
```

```
08EB CA                DEX
08EC D0 F8             BNE    PRBL2         LOOP UNTIL COUNT = 0.
08EE 60                RTS
08EF A5 41    PCADJ    LDA    LENGTH        0=1-BYTE, 1=2-BYTE, 2=3-BYTE.
08F1 38       PCADJ2   SEC
08F2 A4 45    PCADJ3   LDY    PCH
08F4 AA                TAX                  * TEST DISPL SIGN (FOR REL
08F5 10 01             BPL    PCADJ4        *   BRANCH).  EXTEND NEG
08F7 88                DEY                  *   BY DECREMENTING PCH.
08F8 65 44    PCADJ4   ADC    PCL
08FA 90 01             BCC    RTS1          PCL+LENGTH (OR DISPL) +1 TO A.
08FC C8                INY                  *  CARRY INTO Y (PCH)
08FD 60       RTS1     RTS
                       ORG    $900
0900 40       MODE     DFB    $40
0901 02                DFB    $2
0902 45                DFB    $45
0903 03                DFB    $3
0904 D0                DFB    $D0
0905 08                DFB    $8
0906 40                DFB    $40
0907 09                DFB    $9
0908 30                DFB    $30           XXXXXXZ0 INSTRS.
0909 22                DFB    $22
090A 45                DFB    $45           *  Z=0, LEFT HALF-BYTE
090B 33                DFB    $33           *  Z=1, RIGHT HALF-BYTE
090C D0                DFB    $D0
090D 08                DFB    $8
090E 40                DFB    $40
090F 09                DFB    $9
0910 40                DFB    $40
0911 02                DFB    $2
0912 45                DFB    $45
0913 33                DFB    $33
0914 D0                DFB    $D0
0915 08                DFB    $8
0916 40                DFB    $40
0917 09                DFB    $9
0918 40                DFB    $40
0919 00                DFB    $0
091A 40                DFB    $40
091B B0                DFB    $B0
091C D0                DFB    $D0
091D 00                DFB    $0
091E 40                DFB    $40
091F 00                DFB    $0
0920 00                DFB    $0
0921 22                DFB    $22
0922 44                DFB    $44
0923 33                DFB    $33
0924 D0                DFB    $D0
0925 8C                DFB    $8C
0926 44                DFB    $44
0927 00                DFB    $0
0928 11                DFB    $11
0929 22                DFB    $22
092A 44                DFB    $44
092B 33                DFB    $33
092C D0                DFB    $D0
092D 8C                DFB    $8C
092E 44                DFB    $44
092F 9A                DFB    $9A
0930 10                DFB    $10
0931 22                DFB    $22
0932 44                DFB    $44
0933 33                DFB    $33
0934 D0                DFB    $D0
0935 08                DFB    $8
0936 40                DFB    $40
```

BRANCH to . . . pg. 20

Disassembler . . . VECTORED from pg. 18

```
0937 09             DFB    $9
0938 10             DFB    $10
0939 22             DFB    $22
093A 44             DFB    $44
093B 33             DFB    $33
093C D0             DFB    $D0
093D 08             DFB    $8
093E 40             DFB    $40
093F 09             DFB    $9
0940 62             DFB    $62
0941 13             DFB    $13                YYXXXZ01 INSTRS.
0942 78             DFB    $78
0943 A9             DFB    $A9
0944 00      MODE2  DFB    $0                 ERR
0945 21             DFB    $21                IMM
0946 81             DFB    $81                Z-PAG
0947 82             DFB    $82                ABS
0948 00             DFB    $0                 IMPL
0949 00             DFB    $0                 ACC
094A 59             DFB    $59                (Z-PAG,X)
094B 4D             DFB    $4D                (Z-PAG),Y
094C 91             DFB    $91                Z-PAG,X
094D 92             DFB    $92                ABS,X
094E 86             DFB    $86                ABS,Y
094F 4A             DFB    $4A                (ABS)
0950 85             DFB    $85                Z-PAG,Y
0951 9D             DFB    $9D                REL
0952 AC      CHAR1  DFB    $AC                ','
0953 A9             DFB    $A9                ')'
0954 AC             DFB    $AC                ','
0955 A3             DFB    $A3                '#'
0956 A8             DFB    $A8                '('
0957 A4             DFB    $A4                '$'
0958 D9      CHAR2  DFB    $D9                'Y'
0959 00             DFB    $0
095A D8             DFB    $D8                'X'
095B A4             DFB    $A4                '$'
095C A4             DFB    $A4                '$'
095D 00             DFB    $0
095E 1C      MNEML  DFB    $1C                XXXXX000 INSTRS.
095F 8A             DFB    $8A
0960 1C             DFB    $1C
0961 23             DFB    $23
0962 5D             DFB    $5D
0963 8B             DFB    $8B
0964 1B             DFB    $1B
0965 A1             DFB    $A1
0966 9D             DFB    $9D
0967 8A             DFB    $8A
0968 1D             DFB    $1D
0969 23             DFB    $23
096A 9D             DFB    $9D
096B 8B             DFB    $8B
096C 1D             DFB    $1D
096D A1             DFB    $A1
096E 00             DFB    $0
096F 29             DFB    $29
0970 19             DFB    $19
0971 AE             DFB    $AE
0972 69             DFB    $69
0973 A8             DFB    $A8
0974 19             DFB    $19
0975 23             DFB    $23
0976 24             DFB    $24
0977 53             DFB    $53
0978 1B             DFB    $1B
0979 23             DFB    $23
097A 24             DFB    $24
097B 53             DFB    $53
```

```
097C 19               DFB   $19
097D A1               DFB   $A1
097E 00               DFB   $0              XXXYY100 INSTRS.
097F 1A               DFB   $1A
0980 5B               DFB   $5B
0981 5B               DFB   $5B
0982 A5               DFB   $A5
0983 69               DFB   $69
0984 24               DFB   $24
0985 24               DFB   $24
0986 AE               DFB   $AE             1XXX1010 INSTRS.
0987 AE               DFB   $AE
0988 A8               DFB   $A8
0989 AD               DFB   $AD
098A 29               DFB   $29
098B 00               DFB   $0
098C 7C               DFB   $7C
098D 00               DFB   $0
098E 15               DFB   $15             XXXYYY10 INSTRS.
098F 9C               DFB   $9C
0990 6D               DFB   $6D
0991 00               DFB   $0
0992 A5               DFB   $A5
0993 69               DFB   $69
0994 29               DFB   $29
0995 53               DFB   $53
0996 84               DFB   $84             XXXYYY01 INSTRS.
0997 13               DFB   $13
0998 34               DFB   $34
0999 11               DFB   $11
099A A5               DFB   $A5
099B 69               DFB   $69
099C 23               DFB   $23
099D A0               DFB   $A0
099E D8        MNEMR  DFB   $D8             XXXXX000 INSTRS
099F 62               DFB   $62
09A0 5A               DFB   $5A
09A1 48               DFB   $48
09A2 26               DFB   $26
09A3 62               DFB   $62
09A4 94               DFB   $94
09A5 88               DFB   $88
09A6 54               DFB   $54
09A7 44               DFB   $44
09A8 C8               DFB   $C8
09A9 54               DFB   $54
09AA 68               DFB   $68
09AB 44               DFB   $44
09AC E8               DFB   $E8
09AD 94               DFB   $94
09AE 00               DFB   $0
09AF B4               DFB   $B4
09B0 08               DFB   $8
09B1 84               DFB   $84
09B2 74               DFB   $74
09B3 B4               DFB   $B4
09B4 28               DFB   $28
09B5 6E               DFB   $6E
09B6 74               DFB   $74
09B7 F4               DFB   $F4
09B8 CC               DFB   $CC
09B9 4A               DFB   $4A
09BA 72               DFB   $72
09BB F2               DFB   $F2
09BC A4               DFB   $A4
09BD 8A               DFB   $8A
09BE 00               DFB   $0              XXXYY100 INSTRS
09BF AA               DFB   $AA
09C0 A2               DFB   $A2
```

```
342  09C1 A2            DFB    $A2
343  09C2 74            DFB    $74
344  09C3 74            DFB    $74
345  09C4 74            DFB    $74
346  09C5 72            DFB    $72
347  09C6 44            DFB    $44           1XXX1010 INSTRS.
348  09C7 68            DFB    $68
349  09C8 B2            DFB    $B2
350  09C9 32            DFB    $32
351  09CA B2            DFB    $B2
352  09CB 00            DFB    $0
353  09CC 22            DFB    $22
354  09CD 00            DFB    $0
355  09CE 1A            DFB    $1A           XXXYYY10 INSTRS.
356  09CF 1A            DFB    $1A
357  09D0 26            DFB    $26
358  09D1 00            DFB    $0
359  09D2 72            DFB    $72
360  09D3 72            DFB    $72
361  09D4 88            DFB    $88
362  09D5 C8            DFB    $C8
363  09D6 C4            DFB    $C4           XXXYYY01 INSTRS.
364  09D7 CA            DFB    $CA
365  09D8 26            DFB    $26
366  09D9 48            DFB    $48
367  09DA 44            DFB    $44
368  09DB 44            DFB    $44
369  09DC A2            DFB    $A2
370  09DD C8            DFB    $C8
371                     END
 END PASS 2    0 ERRORS

CROSS REFERENCE TABLE     46 SYMBOLS  DEFINED

CHAR1    0952     0231     0108
CHAR2    0958     0237     0110
CHAROU   FFEF     0010     0090     0109     0112     0126     0131     0134
COUNT    0046     0008     0013     0018
DSMBL    0800     0012
DSMBL2   0804     0014     0019
ERR      0835     0041     0026     0028
FORMAT   0040     0002     0045     0099     0106
GETFMT   0839     0043     0040
IEVEN    0826     0031     0024
INSTDS   0812     0020     0014
LENGTH   0041     0003     0047     0070     0097     0138
LMNEM    0042     0004     0079     0085
MNEML    095E     0243     0078
MNEMR    099E     0307     0080
MNNDX1   084E     0055     0064
MNNDX2   0852     0058     0061
MNNDX3   0859     0063     0054     0056
MODE     0900     0149     0033
MODE2    0944     0217     0044
PCADJ    08EF     0138     0015
PCADJ2   08F1     0139
PCADJ3   08F2     0140     0116
PCADJ4   08F8     0144     0142
PCH      0045     0007     0017     0127     0140
PCL      0044     0006     0016     0128     0144     0021     0066     0101
PRADR1   0897     0095     0114
PRADR2   089F     0099     0105
PRADR3   08AD     0106     0096     0098
PRADR4   08BF     0113     0107     0111
PRBL2    08E6     0133     0069     0136
PRBL3    08E8     0134
PRBLNK   08E4     0132     0093
PRBYTE   FFDC     0009     0067     0103     0122     0124
PRMN1    087E     0082     0092
```

SEARCH SUBROUTINE FOR THE 6502 DISASSEMBLER*

by Arthur L. Schawlow

The following is a description, listing and sample run of an object code search subroutine for use with the 6502 Disassembler published in your September 1976 issue. —author

This subroutine can search an assembled program for any combination of characters. It then jumps to the disassembler and displays the command sought. To use it, store the starting address of the program to be examined at 0044. Then at 0050 store the number of bytes to be sought and the bytes themselves.

For example, the November 1976 Apple BASIC used the BACKUP key (HEX code DF) to erase, but the Datanetics ASR-33 keyboard has no BACK UP key. However, it does have a RUB OUT key (HEX code FF). Thus, we wish to find where the long BASIC program checks to see if a character is a DF. That is, we want to find CMP #$DF or in HEX code C9 DF.

```
We enter   44: 00 E0     (ret)
and        50: 02 C9 DF (ret)
Then       7C8R
```

(7C8 is the starting address of the subroutine.) The 02 is the number of bytes being sought.

Almost instantly, the computer displays

```
E286—     C9 DF      CMP #$DF
E288—     F0 11      BEQ $E29B
          etc.
```

Enter R (ret), and the computer displays

```
E4BA—     C9 DF      CMP #$DF
E4BC—     F0 06      BEQ $E4C4
          etc.
```

Thus if we change E287 and E4BB to FF, we are able to use the RUB OUT key to erase a character in a BASIC instruction.

PROGRAM LISTING

```
09F0:     20
07C8-       A0 00              LDY     #$00
07CA-       A2 00              LDX     #$00
07CC-       B1 44              LDA     ($44-,Y
07CE-       D5 51              CMP     $51,X
07D0-       F0 0D              BEQ     $07DF
07D2-       E6 44              INC     $44
07D4-       A9 00              LDA     #$00
07D6-       C5 44              CMP     $44
07D8-       D0 02              BNE     $07DC
07DA-       E6 45              INC     $45
07DC-       4C CC 07           JMP     $07CC
07DF-       E8                 INX
07E0-       E4 50              CPX     $50
07E2-       F0 14              BEQ     $07F8
07E4-       C8                 INY
07E5-       B1 44              LDA     ($44-,Y
07E7-       D5 51              CMP     $51,X
07E9-       F0 F4              BEQ     $07DF
07EB-       E6 44              INC     $44
07ED-       A9 00              LDA     #$00
07EF-       C5 44              CMP     $44
07F1-       D0 02              BNE     $07F5
07F3-       E6 45              INC     $45
07F5-       4C C8 07           JMP     $07C8
07F8-       4C F0 09           JMP     $09F0
```

SAMPLE RUN

```
44:00     E0

0044:     F8
50:02     C9 DF

0050:     00
7C8R

07C8:     A0
E286-       C9 DF              CMP     #$DF
E288-       F0 11              BEQ     $E29B
R
E4BA-       C9 DF              CMP     #$DF
E4BC-       F0 06              BEQ     $E4C4
```

*INTERFACE AGE, Sept. 1976, P. 14.