A 6502 Disassembler from Apple

by Steve Wozniak & Allen Baum
Apple Computer Co., 770 Welch Rd., No. 154
Palo Alto CA 94304; (415) 326-4248

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 1 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.

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.
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.
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.
PRNTX	Outputs the contents of X as two hex digits. Alters A. Uses CHAROUT.
PRNTAX	Outputs two hex digits for the contents of A, then two hex digits for the contents of X. A is altered. Uses CHAROUT.
PRNTYX	Same as PRNTAX except that Y and X are output. Alters A. Uses CHAROUT.
PRBLNK	Outputs 3 blanks. Alters A, clears X. Uses CHAROUT.
PRBL2	Outputs the number of blanks specified by the contents of X (0 for 256 blanks). Alters A, clears X. Uses CHAROUT.
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.
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.
PCADJ2	Same as PCADJ except that A is used in place of LENGTH.
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.

Supplied on APPLE-1 cassette tapes.

```
9F0  20 0 8    JSR DSMBL
9F3  4C1FFF    JMP MONITOR
```

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 pages 8 and 9. All code is on page 8, tables are 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.

locations used by supplied code

```
$40  FORMAT   Used
$41  LENGTH   } by
$42  LMNEM    } INSTDSP,
$43  RMNEM    DSMBL
$44  PCL      } Used by PCADJ,
$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)
950: 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)
9D1: 26 (was 00)
919: 02 (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   #$80      *  ADD INDEXING OFFSET.
0826 4A          IEVEN   LSR   A         * LSB INTO CARRY FOR
0827 AA                  TAX             * LEFT/RIGHT TEST BELOW.
0828 BD FE 08            LDA   MODE,X    INDEX INTO ADDRESS MODE TABL.
082B B0 04               BCS   RTMODE    IF CARRY SET USE LSD FOR
082D 4A                  LSR   A         * PRINT FORMAT INDEX.
082E 4A                  LSR   A
082F 4A                  LSR   A         * IF CARRY CLEAR USE MSD.
0830 4A                  LSR   A
0831 29 0F       RTMODE  AND   #$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 42 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 5C 09            LDA   MNEML,Y
0877 85 42               STA   LMNEM     FETCH 3-CHAR MNEMONIC.
0879 B9 9C 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 4F 09            LDA   CHAR1-1,X *  CORRESPONDING CHARS.
08B4 20 EF FF            JSR   CHAROUT   OUTPUT 1 OR 2 CHARS.
08B7 BD 55 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
08FE 40        MODE    DFB    $40
08FF 02                DFB    $2
0900 45                DFB    $45
0901 03                DFB    $3
0902 D0                DFB    $D0
0903 08                DFB    $8
0904 40                DFB    $40
0905 09                DFB    $9
0906 30                DFB    $30            XXXXXXZ0 INSTRS.
0907 22                DFB    $22
0908 45                DFB    $45            *  Z=0, LEFT HALF-BYTE
0909 33                DFB    $33            *  Z=1, RIGHT HALF-BYTE
090A D0                DFB    $D0
090B 08                DFB    $8
090C 40                DFB    $40
090D 09                DFB    $9
090E 40                DFB    $40
090F 02                DFB    $2
0910 45                DFB    $45
0911 33                DFB    $33
0912 D0                DFB    $D0
0913 08                DFB    $8
0914 40                DFB    $40
0915 09                DFB    $9
0916 40                DFB    $40
0917 00                DFB    $0
0918 40                DFB    $40
0919 B0                DFB    $B0
091A D0                DFB    $D0
091B 00                DFB    $0
091C 40                DFB    $40
091D 00                DFB    $0
091E 00                DFB    $0
091F 22                DFB    $22
0920 44                DFB    $44
0921 33                DFB    $33
0922 D0                DFB    $D0
0923 8C                DFB    $8C
0924 44                DFB    $44
0925 00                DFB    $0
0926 11                DFB    $11
0927 22                DFB    $22
0928 44                DFB    $44
0929 33                DFB    $33
092A D0                DFB    $D0
092B 8C                DFB    $8C
092C 44                DFB    $44
092D 9A                DFB    $9A
092E 10                DFB    $10
092F 22                DFB    $22
0930 44                DFB    $44
0931 33                DFB    $33
0932 D0                DFB    $D0
0933 08                DFB    $8
0934 40                DFB    $40
0935 09                DFB    $9
0936 10                DFB    $10
0937 22                DFB    $22
0938 44                DFB    $44
0939 33                DFB    $33
093A D0                DFB    $D0
093B 08                DFB    $8
093C 40                DFB    $40
093D 09                DFB    $9
093E 62                DFB    $62
093F 13                DFB    $13            YYXXXZ01 INSTRS.
0940 78                DFB    $78
0941 A9                DFB    $A9
0942 00        MODE2   DFB    $0             ERR
0943 21                DFB    $21            IMM
0944 81                DFB    $81            Z-PAG
0945 82                DFB    $82            ABS
0946 00                DFB    $0             IMPL
0947 00                DFB    $0             ACC
0948 59                DFB    $59            (Z-PAG,X)
0949 4D                DFB    $4D            (Z-PAG),Y
094A 91                DFB    $91            Z-PAG,X
094B 92                DFB    $92            ABS,X
094C 86                DFB    $86            ABS,Y
094D 4A                DFB    $4A            (ABS)
094E 85                DFB    $85            Z-PAG,Y
094F 9D                DFB    $9D            REL
0950 AC        CHAR1   DFB    $AC            ','
0951 A9                DFB    $A9            ')'
0952 AC                DFB    $AC            ','
0953 A3                DFB    $A3            '#'
0954 A8                DFB    $A8            '('
0955 A4                DFB    $A4            '$'
0956 D9        CHAR2   DFB    $D9            'Y'
0957 00                DFB    $0
0958 D8                DFB    $D8            'X'
0959 A4                DFB    $A4            '$'
095A A4                DFB    $A4            '$'
095B 00                DFB    $0
095C 1C        MNEML   DFB    $1C            XXXXX000 INSTRS.
095D 8A                DFB    $8A
095E 1C                DFB    $1C
095F 23                DFB    $23
0960 5D                DFB    $5D
0961 8B                DFB    $8B
0962 1B                DFB    $1B
0963 A1                DFB    $A1
0964 9D                DFB    $9D
0965 8A                DFB    $8A
0966 1D                DFB    $1D
0967 23                DFB    $23
0968 9D                DFB    $9D
0969 8B                DFB    $8B
096A 1D                DFB    $1D
096B A1                DFB    $A1
096C 00                DFB    $0
096D 29                DFB    $29
096E 19                DFB    $19
096F AE                DFB    $AE
0970 69                DFB    $69
0971 A8                DFB    $A8
0972 19                DFB    $19
0973 23                DFB    $23
0974 24                DFB    $24
0975 53                DFB    $53
0976 1B                DFB    $1B
0977 23                DFB    $23
```