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.

<table>
<thead>
<tr>
<th>Format</th>
<th>Address Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>(empty)</td>
<td>Invalid, Implied, Accumulator</td>
</tr>
<tr>
<td>$12</td>
<td>Page zero</td>
</tr>
<tr>
<td>$1234</td>
<td>Absolute, Branch (target printed)</td>
</tr>
<tr>
<td>#$12</td>
<td>Immediate</td>
</tr>
<tr>
<td>$12,X</td>
<td>Zero page, indexed by X</td>
</tr>
<tr>
<td>$12,Y</td>
<td>Zero page, indexed by Y</td>
</tr>
<tr>
<td>$1234,X</td>
<td>Absolute, indexed by X</td>
</tr>
<tr>
<td>$1234,Y</td>
<td>Absolute, indexed by Y</td>
</tr>
<tr>
<td>($1234)</td>
<td>Indirect</td>
</tr>
<tr>
<td>($12,X)</td>
<td>Indexed Indirect</td>
</tr>
<tr>
<td>($12),Y</td>
<td>Indexed Indirect</td>
</tr>
</tbody>
</table>

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 Disassemblies 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
{ 9F0 200 8 JSR DSMBL } cassette tapes.
9F3 4C1FF 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.

```
{ $43 RMNEM DSMBL } INSTDSP
$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 ‘S’ (meaning hex) preceding disassembled values make the following changes:
"O P CODE TO A AGAIN.

FORM INDEX INTO MICHONIC TABLE

* 10010100 -> 00110000
* 1001111000 -> 00110000
* 1001110100 -> 00110000
* 1001100000 -> 00000000

* SAVE MICHONIC TABLE INDEX.

PRINT INST (1 TO 3 BYTES)
* IN A 12-CHARACTER FIELD.
ASNS5 ASSEMBLER
CHAR COUNT FOR MICHONIC PRINT.
REV 1.4 8/1/76

* RECOVER MICHONIC INDEX.

FETCH 3-CHAR MICHONIC.
* (PACKED IN 2 BYTES)

SHIFT 5 BITS OF CHAR INTO A.
* (CLEAR CARRY)

ADD 'X' OFFSET.
OUTPUT A CHARACTER OF MICHONIC

OUTPUT 3 BLOCKS.
COUNT FOR 6 PRINT FORMAT BITS.
IF X=3 THEN PRINT ADDRESS VAL.
NO PRINT IF LENGTH=0.

HANDLE REL ADDRESSING MODE
* (NOT DISPLACEMENT)
* MORE SIGNIFICANT BYTE FIRST

TEST NEXT PRINT FORMAT BIT.
* CORRESPONDING CHAR.
OUTPUT 1 OR 2 CHAR.
* (IF CHAR FROM CHAR IS 0)
* DON'T OUTPUT IT

RETURN IF DONE 6 FORMAT BITS.
PCLH = DISPL + 1 TO A.V.

* ADD $X PRECEEDING DISASSEMBLED VALUES

COUNT FOR 20 INST DMALY.
DISASSEMBLE AND DISPLAY INST.
UPDATE PCLH TO NEXT INST.
"DONE FIRST 19 INSTS.
* YES, LOOP ELSE DMALY 20TH.
PRINT PCLH SFR GET OP CODE.

* EVEN-Odd TEST.
* TEST B1.
* XX00011 INSTR INVALID.
* 100001001 INSTR INVALID.
* MASK BITS FOR ADDRESS MODE &
* ADD INDEXING OFFSET.
* LEB INTO CARRY FOR
* LEFT-RIGHT TEST BELOW.
* IF PRINT FORMAT INDEX.

IF CARRY SET USE LSD FOR
* PRINT CLEAR USE MSD.

MASK FOR 4-BIT INDEX.
20 FOR INVALID OCODES.
SUBSTITUTE $00 FOR INVALID OP.
SET PRINT FORMAT INDEX TO 0.

INDEX INTO PRINT FORMAT TABLE.
SAVE FOR ADDRESS FIELD FORMAT.
SAVE 2-BIT LENGTH.
* 1=2-BYTE, 2=3-BYTE.
* OP CODE.
MASK IT FOR 10001010 TEST.
* SAVE IT.