MICRO WORKS

# 80C DISASSEMBLER

# OWNER'S MANUAL

P.O. BOX 4142 DEL MAR, CA 92014 714-942-2400

TABLE OF CONTENTS

80C DISASSEMBLER FOR THE COLOR COMPUTER





WARRANTY STATEMENT

80C Disassembler is provided as is without warranty. Reasonable care has been taken to insure that the program operates as described in this manual. If you find a discrepancy in which it does not operate as such, please notify us. We will attempt to correct any errors brought to our attention, however, we make no guarantee to do so.

# THE COLOR COMPUTER DISASSEMBLER FROM THE MICRO WORKS

The Color Computer Disassembler is a program which is designed to run in the Radio Shack Color Computer and to provide readable listings of machine-language programs in the memory of the computer. These listings may be displayed on the computer's screen or sent to a printer, and may be in any of several formats. The code to be disassembled may be resident in the computer or may be any 6809 code which is loaded into the computer's memory. This document describes the operation of the disassembler, and should enable you to quickly begin use of the program as well as allowing you later to understand and fully use the many options available.

This program is on a cassette tape which should be loaded with the CLOADM command. It will load starting at location $0600 and will wipe out any BASIC program that is there. BASIC should not be run after the tape is loaded; type EXEC to run the disassembler.

You will be prompted for a series of parameters, starting with "START ADDRESS". To get started, simply type a carriage return ("ENTER" key) in response to every question. In this program, all answers may be "defaulted" with a carriage return. After the last question, there will be a pause (for pass 1, the symbol table being built). When all the questions are defaulted, the entire BASIC ROM will be disassembled and pass 1 will take about 45 seconds. Then the listing will start.

To control the speed of the listing, the following keys may be used:

- Space bar will put the listing in single step mode; another key will put it back.
- Shift-@ will stop the listing as it does in BASIC.
- "S" will speed up and slow down the listing.
- BREAK key will allow the listing to be restarted at another address.

The question "RESTART WHERE?" will appear at the end of the listing or when BREAK is pressed. If it is defaulted (return is pressed) the program will restart from the beginning. Some of the questions which the program asks pertain to formatting and will be used often; some pertain to exactly what should be disassembled and will depend upon your application; still others are only for special cases and you may never need to answer them.

For all of your responses, your options are as follows:

(1) You may default by simply pressing RETURN. All questions may be defaulted. When in doubt as to the meaning of a question, just press RETURN.

(2) Addresses may be entered as a string of hex digits. If more than four digits are entered, the last four are used. Normal editing characters such as backspace are allowed.

(3) Addresses may be entered in base ten by prefacing them with a period (eg. ".10" is the same as "A".)

(4) Yes / No questions may be answered with "Y" or "YES" or "N" or "NO". Default (RETURN) is the same as NO.

(5) The question "AREA OPTIONS" has a different format and is

discussed below. When in doubt, default.

(6) You may press the BREAK key. This will restart the program at the beginning.

The first question asks for an address at which to start disassembling, and the next for an address at which to stop. If these are both defaulted, you will be asked later if you want to default the entire definition of what to disassemble (see below).

The next question asks for an offset to where the code can be found. This is only used if some code has been copied to an address where it does not ordinarily run, and is usually defaulted (which is the same as a zero).

Next is the symbol table start and end address. This specifies some unused area of RAM which the program may use freely. The start and end default respectively to just after the end of the disassembler and 50 bytes below the stack. They only need to be entered if these values will interfere with a program being disassembled.

Next is the area options, so we had best digress a little into the idea behind them. A program is generally made up of machine code, data tables, address tables, and so forth, all intermixed at the discretion of whoever wrote the program. Since there is no reason why data can't look like code, it is not possible for any disassembler to automatically figure out the boundaries of these areas. The "AREA OPTIONS" in this disassembler allow you to specify how to treat each area within the block being disassembled.

Disassembly is normally a two-step process. First, you disassemble the entire block treating everything as code. Certain blocks will stand out as being data, and the ASCII column on the output will help to identify text strings. You note a list of these areas and then enter them to make a new listing which is much "cleaner". If a perfect listing is desired, the new listing is studied at length until a complete list of areas is discovered, and the disassembler is run yet again.

The area types allowed by this program are as follows:

P - program area (machine code)
D - data area (FCB mnemonics)
A - address area (FDB mnemonics)
S - text string area (FCC mnemonics)
V - variable area (RMB mnemonics - contents of memory ignored)
T - table area (alternating FDB and FCB)
E - end of last area

To enter an area, type the letter of the area type, a space, and an address. For example, if there is data at addresses 4567 through 4568, type:

```
D 4567          (data area starts at 4567)
P 4569          (program area resumes at 4569, one byte past the
                 last data byte)
```

After the last option is entered, simply press RETURN.

The actual effect of entering a starting address (in answer to the first question) is to have that address entered in the area table as a "P" area. If that address is later specified as another type (or if any area is respecified) the new definition simply takes the place of the old one. The effect of entering an ending address (in answer to the second question) is to have that entered as an "E" area. When a RETURN is entered in response to the AREA OPTION question (whether or not it is the first time it was asked) the program checks that there are at least two boundaries specified, and if not the question is repeated.

If nothing has been entered, however, you are given a choice of copying the last set of areas used. The question is phrased so that a NO or default answer will copy in the previous set of areas. This table may then be added to. This is useful in building a set of areas, and restarting the program whenever a new area is discovered. If this option is used when the program is first loaded, however, the default set of areas will be set to those corresponding to the Level 1 BASIC interpreter ROM.

The remaining questions deal with the format of the output. You may select the full output mode, the scan format, or the default format. The full output takes two lines on the screen for each line of generated source, but contains the complete output with reference and cross reference addresses. If an 80-column printer is available, it is recommended that this format be used. The scan format contains the ASCII column and complete data columns at the expense of labels. It is useful for determining where the various data and code areas are. The default output mode gives only the first two bytes of the hex value in order to make room for labels. Both the scan and default format listings will fit in one line across the screen or across a 32-column printer, and so will be half as long as the full listing.

The next question is whether or not to send the output to the printer. Any printer that works with BASIC will work with the disassembler. If the printer is requested, then you are asked if it is 80 columns. Actually anything wider that 64 columns will work in 80-column mode. For a narrow printer, the next question is: "NO CR ON COL 32?". This is for the benefit of those printers which automatically produce a carriage return / line feed on column 32 and for which the program-generated carriage return would be redundant. If you type "NO" (or default), and the listing contains unwanted blank lines, try typing "YES" to this question next time.

The program now executes Pass 1. This will take anywhere from no apparent time on a small disassembly to 45 seconds on the entire BASIC ROM. When Pass 1 is complete, Pass 2 starts and the listing will be produced.

When Pass 2 finishes, or is stopped by the BREAK key, it asks where to restart. Any address may be given within the area covered by Pass 1. If an address is given beyond the end of Pass 1 the question will be repeated. If it is before Pass 1, however, the disassembler will not object and will disassemble using the last area type it was left in. This last feature allows the disassembler to be used like

a one pass disassembler by specifying a short Pass 1 at the top of memory, then restarting wherever you want to disassemble. If you do not give an address to the restart question, the whole program is restarted.

The cross reference produced by the full format listing is used to find every explicit reference to any address. It is used as follows: Find the address of the label in question. Look it up in the table at the end, which is sorted by address. (Labels within the program are listed first, and externals listed separately.) The number given after the address in that table is the address of the last reference to that label. Now look at that reference. A number given in the cross-reference column at that line will point to the next prior reference, and so on. Four dots in the cross-reference column indicates that that is the first reference to that address, and is the end of the chain. A blank in the cross-reference column indicates that that is the only reference to an address and saves looking up the address in the table at the end.

While the listing is being generated, numeric keys may be pressed to change listing modes. 1, 2, or 3 may be pressed to change into Full, Scan, and Default modes respectively. 4, 5, and 6 may be pressed to change into three more modes which are seldom used: Source Only, Reference, and XReference; these are all one-line modes which sacrifice various fields in order to include others within the limited width of the screen.

Sometimes there is a reference to a label that is not on the first byte of an instruction. This happens often when disassembling a program where the data and variable areas are not known. It also occurs in perfectly disassembled listings when the programmer used such dirty tricks as using a Compare X Immediate opcode as "Skip over two bytes" and following it with a two byte instruction. There are two ways this disassembler deals with this. To increase readability, it normally will set the program counter back so as to disassemble the instruction at the label. This method, though nice, is not correct in that the listing produced will not then reassemble to the original code, and for this reason the "Source Only" format causes it to print labels of the form:

```
   L1234 EQU *-1
```

In scan mode, where such backward referenced labels are mostly due to lack of area specifications, they will also print as EQUs.

APPENDIX I: EXAMPLES OF THE DISASSEMBLER'S OUTPUT

THE FOLLOWING ARE EXAMPLES OF THE OUTPUT OF THE DISASSEMBLER. THE THREE FORMATS ARE SHOWN. THE FULL FORMAT, WHICH TAKES DOUBLE LINES ON A 32-COLUMN PRINTER, IS SHOWN PRINTED ON BOTH A 32-COLUMN PRINTER AND AN 80-COLUMN PRINTER.

IN ADDITION, AN EXAMPLE IS GIVEN OF THE HEADER OUTPUT WHICH IS PRINTED AT THE START OF LISTINGS. THIS INCLUDES NAM, ORG, AND EQU STATEMENTS WHICH WOULD ALLOW THE DISASSEMBLY TO BE REASSEMBLED.

```
      NAM   DISASM
      ORG   $3000
0000  EQU   $0000
3119  EQU   $3119
313C  EQU   $313C
3141  EQU   $3141
31C1  EQU   $31C1
33DC  EQU   $33DC
3C2A  EQU   $3C2A
3E5A  EQU   $3E5A
3EF5  EQU   $3EF5
3EF9  EQU   $3EF9
3F0A  EQU   $3F0A
3F1C  EQU   $3F1C
3F35  EQU   $3F35
3F36  EQU   $3F36
4484  EQU   $4484
CD00  EQU   $CD00
D286  EQU   $D286
D2B5  EQU   $D2B5
D2C4  EQU   $D2C4
D2C7  EQU   $D2C7
D30F  EQU   $D30F
D312  EQU   $D312
```

HEADER WHICH ALLOWS REASSEMBLY OF OUTPUT

EXTERNAL EQUATES

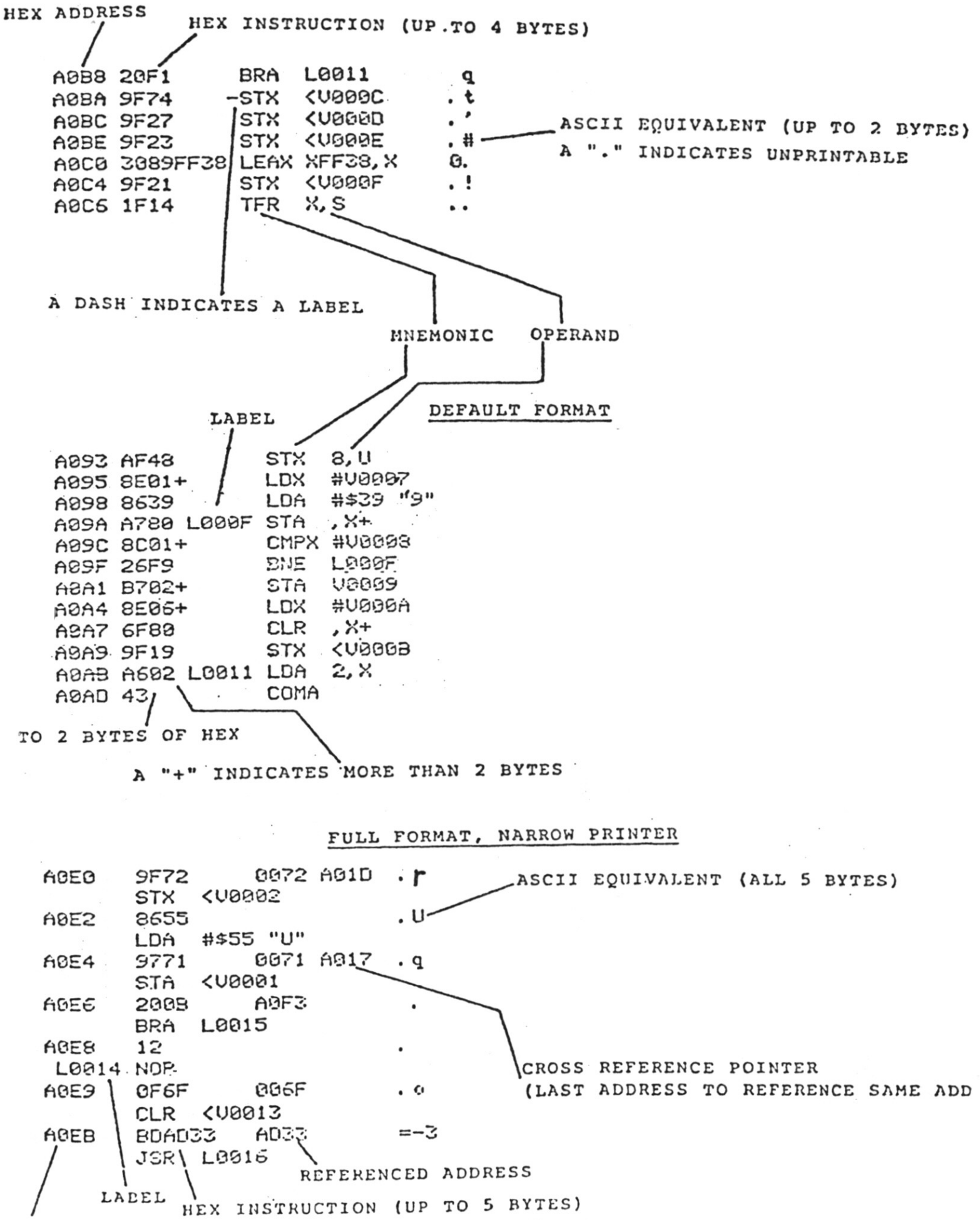
FULL FORMAT, 80-COLUMN PRINTER

```
302C   BDD2C7   D2C7      =RG          JSR  XD2C7
302F   2502     3033      %.           BLO  L0004
3031   4F                 O            CLRA
3032   39                 9            RTS
3033   BDD2C4   D2C4      =RD    L0004 JSR  XD2C4
3036   4D                 M            TSTA
3037   39                 9            RTS
3038   01                 .            FCB  $01 <<
3039   2C10     304B      ,.           BGE  L0005
303A                                L0000 EQU  *-1
303B   FF3F36   3F36      .?6          STU  X3F36
303E   BD3C2A   3C2A      =<*          JSR  X3C2A
3041   53                 S            COMB
3042   4F                 O            CLRA
3043   55                 U            FCB  $55 <<
3044   52                 R            FCB  $52 <<
3045   43                 C            COMA
```

UN-DISASSEMBLABLE BY ARE FLAGGED WITH "<<

A LABEL ON THE SE BYTE OF INSTRUCTI

TEXT STRINGS SHOW UP IN THE THIRD


HEX ADDRESS
HEX INSTRUCTION (UP TO 4 BYTES)
A0B8 20F1 BRA L0011 q
A0BA 9F74 -STX <V000C .t
A0BC 9F27 STX <V000D .'
A0BE 9F23 STX <V000E .#
A0C0 3089FF38 LEAX XFF38,X 0.
A0C4 9F21 STX <V000F .!
A0C6 1F14 TFR X,S ..
ASCII EQUIVALENT (UP TO 2 BYTES)
A "." INDICATES UNPRINTABLE
A DASH INDICATES A LABEL
MNEMONIC
OPERAND
LABEL
DEFAULT FORMAT
A093 AF48 STX 8,U
A095 8E01+ LDX #V0007
A098 8639 LDA #$39 "9"
A09A A780 L000F STA ,X+
A09C 8C01+ CMPX #V0008
A09F 26F9 BNE L000F
A0A1 B702+ STA V0009
A0A4 8E06+ LDX #V000A
A0A7 6F80 CLR ,X+
A0A9 9F19 STX <V000B
A0AB A602 L0011 LDA 2,X
A0AD 43 COMA
TO 2 BYTES OF HEX
A "+" INDICATES MORE THAN 2 BYTES
FULL FORMAT, NARROW PRINTER
A0E0 9F72 0072 A01D .r
STX <V0002
A0E2 8655 .U
LDA #$55 "U"
A0E4 9771 0071 A017 .q
STA <V0001
A0E6 200B A0F3 .
BRA L0015
A0E8 12 .
L0014 NOP
A0E9 0F6F 006F .o
CLR <V0013
A0EB BDA033 A033 =-3
JSR L0016
ASCII EQUIVALENT (ALL 5 BYTES)
CROSS REFERENCE POINTER
(LAST ADDRESS TO REFERENCE SAME ADD
REFERENCED ADDRESS
LABEL
HEX INSTRUCTION (UP TO 5 BYTES)
ADDRESS