

MANUAL

ZDT—A Debugging Program for the Z80



ZDT - A DEBUGGING PROGRAM FOR THE Z80

Distributed by:

Lifeboat Associates
1651 Third Avenue
New York
N.Y. 10028

212-860-0300

ZDT Z80 Debugging Utility

ZDT

The Micronics, Inc. Z-80 Disassembler (ZDT) is a flexible operating aid for use in documenting and debugging programs, as well as assisting in machine language programming. ZDT has 25 separate commands, all consisting of one letter of the alphabet, and followed optionally with address and data, according to the format of each command. The Disassembler runs under any memory size and it automatically relocates itself to the upper end of memory, 1900h bytes below the base of the CP/M or equivalent operating system.

INITIALIZATION

The only change needed is to set the number of 128 byte sectors per track for disk reads and writes (W and R commands). The standard version of ZDT is set for 26 sectors. If you are using it with a system with other than 26 sectors per track, load ZDT.COM with the L command and program 074AH (494AH for versions loading at 4300H) to the correct value with the P command. Then return to the operating system (B command), and SAVE 24 ZDT.COM, as follows:

A>ZDT

ZDT Version 1.40 (C) Micronics, Inc. 1980

M>L ZDT.COM

M>P 074A

074A 1A 40 (Enter correct number of sectors in hex)

074B BA ^C (Enter control -C)

M>B

A>SAVE 24 ZDT.COM

A>

All functions of ZDT will correctly operate except for R and W of more than the number of sector for one track until this change is made. (Please note however that these changes have already been implemented for the ZENITH-HEATH H-89, TRS-80 Model 1, and Intertec SUPERBRAIN.)

Z80 is a registered trademark of Zilog, Inc. of Cupertino, CA.

CP/M is a trademark of Digital Research of Pacific Grove, CA.

ZDT Z80 Debugging Utility

USING ZDT

Upon loading the system, by typing ZDT CR after the system (A>) prompt the prompt M> will signify command status. Typing any character will enter a command or will result in M>?, signifying a bad command or character other than A through Z being typed. CTRL-P is ignored and passed to the operating system for printer on-off switch, and CTRL-S will stop the display. Entering CTRL-C will terminate the current operation and return to the M> prompt. Upon typing a command, the program will execute the command, or insert a space, and accept address or data. If the command requires more than address or data, another space will be added between each separate entry, then execution begins immediately upon entering the final data keystroke. If an error is made on entry, a delete or rub will delete the character and backspace the display, ready to accept a correct entry. On entry of address and byte data, characters other than 0-9 or A-F are edited out. At any time before data entry is complete, the command can be aborted by typing CR or CTRL-C, returning to the M> prompt. During the display of data, such as a disassembler listing or dump, the display can be halted by hitting the space bar. Any character will then continue the display, or CTRL-C will terminate it.

Reentering ZDT can be accomplished by jumping to the address computed by adding 043 to the address at 0006h

BREAKPOINT ROUTINE

The breakpoint routine consists of commands G, U, V and X (see below). Using this module allows running a machine language program with up to six halts or breakpoints. The points must be chosen to fall at the beginning of an instruction. Use the S (symbolic dump) to verify the address, and use U to set the breakpoint. UP to six breakpoints may be set at any one time, and V will display the ones currently set. The purpose of the routine is to allow controlled execution of a program in memory, with debugging each instruction or module by setting the breakpoint, and observing the register dump which occurs as each breakpoint is executed. When the register dump occurs, the contents of memory may be examined with the D command, and execution of the program continued with the G command, automatically restoring the old instruction and register contents. Type X to see the contents of the registers displayed as they were upon encountering the breakpoint. After the last breakpoint

ZDT Z80 Debugging Utility

has been encountered, the G command prints a message: NO BREAKPOINT SET, CONTINUE (Y/N). Answer Y to continue. If G is accidentally typed in other circumstances and a Y is given, the registers are loaded with whatever junk is in the usual register storage area and the system jumps to a random address. This is not going to give good results.

DISASM.TST

DISASM.TST is a file containing Z-80 instructions. Load with the L command (L DISASM.TST). With no address specified, the file will be loaded at 0100H (4300H for relocated versions), so disassemble this test program by: S 0100 04D6. (S 4300 46D6.). This is just a sample of code to show the format of the S symbolic dump.

ZDT COMMANDS

The following is a summary of commands in the ZDT debugger. Use the CP/M DDT manual supplemented by these notes

A bbbb

Enters an ASCII character into memory from the keyboard, beginning with address bbbb. Echoes the entry to the keyboard. Delete is not supported in this mode, and will enter any keystroke into the next memory location and continue until terminated by [CTRL-C], then prints the next available address before returning to M>.

B

Warm Boot (returns to CP/M). B does not disturb the programs in the TPA (0100 to the bottom of CP/M 4300 for the relocated version) so that programs may be saved with the modifications made by the disassembler. Type SAVE (etc. see CP/M information for correct syntax).

ZDT Z80 Debugging Utility

C bbbb eeee nnnn

Memory compare. Compares byte by byte a memory block starting at bbbb and ending at eeee-1 with the block starting at nnnn. Useful in verifying prom programming, or other data supposed to be the same. Prints address and data in first block then data in other block when compare fails, or nothing at all if compare succeeds.

D bbbb eeee

Dumps hex and ASCII value of memory between addresses bbbb and eeee. Does not include byte at eeee if another line would be generated. Fills all of last line, even if greater than eeee.

E bbbb eeee [ASCII STRING]

Search for ASCII string (limit 16 characters) printing occurrences of string from address bbbb to eeee, not including eeee. Requires carriage return at end of string entry.

F bbbb eeee [HEX STRING]

Search for hex string (limit 16 bytes) printing addresses of all occurrences of the specified combination from address bbbb to and not including eeee. Requires carriage return to signify end of string of hex bytes.

G

Go to next breakpoint. Loads registers as they were upon encountering the breakpoint, restores the instruction at the old breakpoint, and will continue execution to the next breakpoint. If G is used without a breakpoint, the message NO BREAKPOINTS SET,CONTINUE (Y,N) will display.

ZDT Z80 Debugging Utility

H aaaa nnnn

Hex add and subtract. Prints + and the sum, - and the difference in hex on the console.

I nn

Input. Prints the hex value at the input port nn.

J nnnn

Jump to and execute location nnnn. Puts the M> entry point on the stack, so the routine at nnnn can end in a return, if so desired. This is convenient when testing subroutines or modules of a larger program.

K

Keyboard echo. Inputs from keyboard, and puts out to console. Memory not affected.

L filename.ext nnnn

Loads a disk file to location nnnn so it can be disassembled or dumped (default is to 0100 Hex, 4300 Hex for the relocated version). For instance, L B:DISASM.COM will load the disassembler for disassembly. B: is the drive number, and can be omitted for drive A. It is possible to load a file which is too big to fit under the disassembler, and will write over it, causing a halt to the proceedings.

M bbbb eeee nnnn

Move a block of data beginning at bbbb and ending at eeee -1 to nnnn. Moves files up and down in memory, one byte or many. Does not get confused and write over itself.

N aaaa

Enters offset aaaa. This allows loading a program for disassembling at an address other than the run address, and subtracting the offset from all addresses printed on Dump, Symbolic Dump, etc. Use the H command if necessary to determine the correct offset. Enter N followed by a carriage return to reenter the default value of 0000. (normal)

O nn dd

Output. Puts byte dd out to port nn.

P bbbb

Program. Keyboard programming, input hex from keyboard, and insert into memory. Space bar advances without changing memory, and rub or delete backs up the pointer to previous byte or bytes to make corrections. Prints location and current value, a space, and accepts data, which is printed on the console as it is put into memory.

Q bbbb eeee

Prints ASCII, of any, from memory location bbbb to eeee-1.

R n ll tt ss bbbb

Read block of data from disk n of length ll (pages-256 bytes) starting at track tt and sector ss, placing the data at bbbb. The numbering is hexadecimal, values for tt from 00 to 4C, ss from 01 to 1A for standard 8" IPM diskette formats. These values will vary depending on the size of your disk. Refer to your hardware manual for details.

ZDT Z80 Debugging Utility

S bbbb eeee

Symbolic dump. Prints addresses, the hex code located there, and the Zilog/Mostec mnemonic for the instruction represented from address bbbb to eeee -1. The last instruction will be printed entirely if the end address falls within a two-or-more-byte opcode.

T bbbb eeee

Destructive memory test from location bbbb to eeee-1. If this core includes the stack (at 0A0H) or the disassembler, results will not be as expected. Prints memory addresses, the byte which was written, and the byte fetched, when errors occur.

U bbbb

Set breakpoint at address bbbb. Up to six breakpoints may be set at one time. More than six will generate an error message. Breakpoint will be reset to the original byte when found and the G command used to continue execution.

V

Displays all currently set breakpoints.

W n ll tt ss bbbb

Write on disk n the data at bbbb for a length of ll starting at track tt, sector ss. Writes sequentially until the length (in pages) is written to disk. See above under R (read).

ZDT Z80 Debugging Utility

X

Registers display. Current contents of all the Z80 registers.

Y

Not used.

Z bbbb eeee dd

Zero or set memory. Places in memory locations starting with bbbb and ending with eeee -1 the byte dd.

1026

LIFEBOAT ASSOCIATES
SOFTWARE PROBLEM REPORT

please use this form to report errors or problems in software supplied by Lifeboat Associates. This form is designed to act as a transmittal sheet, and problem details can be described on additional pages.

Date: _____

Software Product Name: _____ Version No. _____

Disk Format: _____ Serial Number: _____

Computer Type: _____ CPU Type: _____

Operating System: _____ Serial Number: _____ Version: _____

Memory Size: _____ Number of Disk Drives: _____

Please describe the problem you have encountered. Include references to the manual if appropriate. Try to reduce the problem to a simple test case. Enclose any appropriate listings. If you have discovered a patch or interim solution please describe it.

This form may also be used to suggest enhancements to our software products.

Information on product changes and current version numbers are published in Lifelines, our software newsletter.

PROBLEM DESCRIPTION:

Name: _____ Phone: _____

Address: _____

City: _____ State: _____ Zip: _____

Return to: Lifeboat Associates
1651 Third Avenue
New York, New York 10028

lspr-08/01/80