ZSTEP: A SUMMARY

by Sam Solomon

ZSTEP is a machine language monitor written for Radio Shack's Level II TRS-80 whose main function is as a debugging tool for machine language programs. The monitor, which occupies about 2.1K bytes at the end of a 4K or 16K RAM, consists of three main sections: trace, edit and support.

TRACE

ZSTEP features three separate series of commands that enable the user to trace a program in development:

Single-Step: ZSTEP accomplishes software single-stepping by simulating execution of the program being traced. CPU control is retained by the ZSTEP monitor at all times. Contents of all registers of the traced program, including program counter (PC), are stored. When the "N" (next instruction) command is given, all register values are retrieved, the next instruction is executed, new register values are stored, all registers are displayed on the video screen, and control returns to user.

The user specifies the address at which tracing is to start via the "T" command. While in single-step mode, any instruction that does not affect the program counter is executed directly. Instructions such as JP, CALL or RET are simulated by ZSTEP, allowing the monitor to retain control of the CPU.

Single-Step Continuous: This mode allows speedy execution of portions of code which do not need to be single-stepped yet through which the user wishes to retain control. By one of four methods, the user sets a breakpoint. ZSTEP proceeds to single-step through the program without displaying registers or returning control to the user until the breakpoint is encountered. The breakpoint is not inserted into the program under trace, but stored by ZSTEP.

The user may escape from this mode at any time via the shift "Z" command. Upon receiving this command ZSTEP breaks simulated execution of the monitored program and returns to single-step mode. Following are the four ways to set a breakpoint:

Command "B" followed by a hex address sets a breakpoint directly and sends ZSTEP into single-step continuous mode.

Command "G" followed by a single digit (which may be defaulted) sets a breakpoint at the sum of the current PC value and the given parameter. This allows easy continuous tracing through subroutines or loops. For example, if the PC points to a CALL instruction in the monitored program, the command "G 3" causes the entire subroutine to be simulated and control returned to user upon return from the routine. "G 1" would do the same for a RESTART instruction. "G 2" might cause a continuous trace through a DJNZ loop.

Command "Z" (not to be confused with shift "Z") restores the last set breakpoint (set only by "B" or "G" commands) and goes into single-step continuous mode. This comes in handy when tracing a loop. The user can set a breakpoint within the loop, then repeatedly use the "Z" command to examine registers and memory after each pass through the loop.

Command "Y" causes tracing to continue in single-step continuous mode until the next non-sequential instruction in the traced program is encountered. A non-sequential instruction is any instruction forcing a program branch, such as jump, call or return, or a jump, call or return on condition if the condition is met. This command allows the user to get a picture of program flow by speeding through sequential portions and stopping only at branches.

Software Interrupt: The user sets a software interrupt (breakpoint) by inserting a call to the appropriate ZSTEP routine into the monitored program. This is done through the command "S". The "J" command then transfers CPU control to the monitored program, enabling execution at CPU speed. Upon encountering the call to ZSTEP, control is returned to the monitor and current contents of all registers are displayed. The call to ZSTEP may be removed from the monitored program and replaced with the original code through the command "R".

EDIT/RELOCATE

The "E" command is a major feature of ZSTEP. This command allows a block of code to be relocated to any other address in memory. The user specifies the starting and ending addresses of the block to be moved, and the starting address of the location to which the user wishes it moved. In addition to moving the code, the command performs the following functions:

All jumps and relative jumps are adjusted. If the jump destination is within the block to be moved the effective jump address will be changed to the new location. If the destination was outside the moved block, the effective jump address will remain the same. If moving the block forces a relative jump to go out of range, the error is displayed along with the attempted "to" and "from" addresses.

All instructions in which an immediate or indirect address is used (e.g., LD HL,41FE or LD (378F),A) are displayed along with their new (post-edit) locations. This allows the user to manually adjust storage or table addresses which must be relocated along with the code.

JP (HL) and JP (IX or IY) instructions are displayed along with their new addresses.

The second editing command provided by ZSTEP is the "M" command. This functions identically to "E" except the block to be moved is treated as literal data rather than code, and is moved directly.

SUPPORT

The support section of ZSTEP includes commands that display or alter memory, perform hexadecimal calculations and save and load files.

Display/Alter Memory: The "DM" command permits the display of any byte or bytes in memory. Subcommands permit the following byte to be viewed or the displayed byte to be replaced with user input. Memory is displayed four bytes per line, permitting up to 64 bytes to be viewed on a single display.

The "DR" command displays current contents of all CPU registers as well as the last four and next eight bytes of memory addressed by the PC.

Contents of the stack may be viewed directly using the "DS" command.

Contents of CPU registers may be altered using the "DM" command or the AF, HL or SP registers may be altered directly using the "IA", "IH" or "IS" commands.

Calculate: The "C" command permits two hex bytes to be added or subtracted and the result displayed.

Save/Load: Full cassette tape save and load commands, compatible with Radio Shack's cassette storage format, are provided. Starting address, ending address, entry address and optional file name are specified by user on a "punch" command. The load command loads the next file on tape and displays its file name and entry address.

CONTROL COMMANDS

The BREAK key may be used at any time to abort a command and return control to user.

The BACKSPACE key may be used to delete a character before the end of a field. If ZSTEP expects a four-digit hex address, for example, the first three digits may be deleted.

Shift Z breaks a continuous trace and returns to single-step mode.

ENTER is used to default on certain parameters, all of which have values which may be pre-set by the user (details provided with ZSTEP).

USER COMMANDS

Space for nine primary user commands is provided (subcommands, of course, are unlimited). ZSTEP contains a 52-byte jump table which is accessed directly when an alphabetic command is entered. 17 characters are used by ZSTEP. The remaining table entries address an invalid command routine until altered by the user. Details provided with ZSTEP. (Above figures are decimal.)

HOW TO ORDER ZSTEP

ZSTEP occupies the last 2146 bytes of any 4K block of RAM. Thus, the last three digits, in hex, of its location will always be 79E to FFF. In a 4K system the first digit must be a 4, in a 16K system it could be a 4, 5, 6 or 7, etc. Specify starting location when ordering.

The instruction manual for ZSTEP includes a detailed explanation of each command, a list of storage areas used by the program, and specifications and locations of ZSTEP subroutines of possible user interest.

A source and object code listing is available at a small extra charge.

ZSTEP COMMAND SUMMARY

PC is program counter, SP is stack pointer.

Default variables are pre-set but may be re-set by user.

Capital letters entered literally by user. Blanks printed by ZSTEP. Lower case letters represent one character each. Characters enclosed in parentheses represent optional entries.

B aaaa : Continuous single-step without display. Break at PC equals aaaa.

C aa bb (c) : If c equals "S" display aa-bb. Else display aa plus bb.

DM aaaa (b) : Display b bytes of memory starting at address aaaa. b is one hex digit (1-F). Initial default parameter for b is hex 20.

DR : Display current contents of all CPU registers plus contents of last four and next 8 bytes addressed by PC.

DS (b) : Display b bytes of memory starting at current value of SP. (Identical to DM except SP value is used instead of aaaa.)

E aaaa bbbb cccc : EDIT by moving block of object code starting at aaaa and ending at bbbb to cccc. Adjust absolute and relative jumps occurring in block to be moved. Print flags and error messages.

G (a) : Continuous single-step without display. Break at PC equals current PC plus a. Initial default parameter for a is 3.

Ia bbbb : Insert the hex value bbbb into register pair a. a may be "A" (AF), "H" (HL) or "S" (SP).

J (aaaa) : Transfer program control to address aaaa (JUMP). Default parameter for aaaa is current value of PC.

K (a) : Skip next a bytes (add the hex value to PC). Initial default parameter for a is one.

L : Load next file from cassette. Print file name and entry address.

M aaaa bbbb cccc : Move block starting at aaaa and ending at bbbb to cccc.

N : Execute single instruction at address pointed to by PC (single-step). After execution, display new contents of registers (execute DR command).

P aaaa bbbb cccc (dddddd) : Punch to cassette block starting at aaaa and ending at bbbb. Assign entry address cccc and file name dddddd where dddddd is six or fewer alphanumeric characters.

R : Remove software interrupt and replace with the 3 bytes of code originally at its location.

S aaaa : Set software interrupt at address aaaa by saving 3 bytes of code at that address and inserting a call to the interrupt routine in ZSTEP.

T aaaa : Begin trace at address aaaa by loading the value aaaa into PC and executing command "DR".

Y : Single-step continuously until the next transfer of control is encountered in program being traced.

Z : Single-step continuously without display. Break at address set by last "B" or "G" command.

Shift Z : Halt continuous single-step at current instruction, execute DR command and return control to user.

BREAK key: Abort command and await new command.

BACKSPACE key : Erase last-entered character (may be used within a single field only).