## **Software Reference Manual**

ZMON Z8000 Monitor

# Central Data Corporation

Z8000 Monitor Manual

Central Data Corporation P.O. Box 2530, Station A Champaign, Il 61820 Phone: (217) 359-8010

Copyright (C) 1980, Central Data Corporation

### TABLE OF CONTENTS

<u>Section</u>		<u>P</u> a	age
1	General Information		1
2	Installation and Use		2
3	Operating Instructions		4
4	Command Reference		6
	Set or Display Breakpoints. Change Memory Locations. Clear Breakpoints. Compare Memory Blocks.  Dump Memory. Execute. Initialize Memory by Bytes. Display Input Port Value. Initialize the System Stack Pointer. Initialize Memory by Words. Load Move Memory Blocks. Memory Test. Write an Output Port. Write Process Number. Change and Display Registers. Single-Step.		8 9 10 111 112 113 114 115 116 117 118 119 221 22
5	System Calls		24
	Return to the Monitor		26 27 28 29 30

#### 1. General Information

The Central Data Z8000 Monitor (ZMON) is designed for use with the Memory Management Z8000 CPU Board. The monitor is designed to allow a user to easily debug Z8000 programs written for segmented operation. An extensive command set allows inspection and alteration of memory, I/O ports and registers; breakpoints and single-stepping; block memory moves and comparisons; loading of data to and from other systems; and a memory test.

The monitor is interrupt driven from one of three I/O boards manufactured by Central Data. The marking on the top of the PROMs will be as shown in the table below for the different I/O drivers:

ZMONQ	Quad Serial Interface	•
ZMONO	Octal Serial Interface	
ZMONI	Intelligent Octal Serial	Interface

The version number of the monitor is printed on the terminal after initialization, and this number should be referenced in any correspondence with Central Data. A complete listing of the monitor is available for an additional charge. Any user interested in such a listing should consult the factory for the current price.

The following keys are accepted in command mode of the monitor, with their hex value and function printed beside them:

CTRL-P	H10	Copies the last entry
DEL	H7F	Deletes one character
BS	80Н	Deletes one character
CR	HOD	Enters a command
ESC	H1B	Aborts a command

Any keys specific to certain commands are detailed in the command reference section.

#### 2. Installation and Use

The two PROMs supplied for the monitor should be inserted into socket locations 36 and 37 on the Z8000 CPU board. As marked on the board, the PROM marked HIGH should be inserted into position 37, with the PROM marked LOW inserted into position 36.

The user should be sure that the PROM set received corresponds to the serial I/O board which is in the system (refer to the General Information section of this manual). After this has been checked, the I/O board should be addressed and strapped to work with the monitor. The Quad Serial Interface board should be addressed at HO100, while the Octal and Intelligent Octal Interfaces should start at HO200 and HO300, respectively.

For non-intelligent I/O boards, all transmitter and receiver interrupts should be enabled (using TxRDY for the transmitter interrupt on the Ouad Serial Interface), and the Multibus interrupt should be set to level 1. Since interrupts are totally programmable on the intelligent I/O board, no interrupt level needs to be selected, although level 2 is used. Also for an intelligent I/O board, the common memory base address should be set at HFEOOOO.

The terminal should be wired to port 0 of the serial interface board being used. If another system is being hooked to the Z8000 board (see the Load command), it should be connected to port 1 of the octal serial interface boards, or port 2 of the quad serial interface board. Port 2 of the quad serial interface board is used so that independent baud rates can be set for the terminal and the external system.

At least 8K of RAM must be available starting at address 0. The monitor, when first executed after power-on or initialization, moves itself from PROM to the first 4K of RAM. It then deselects the PROM, and re-maps the RAM to start at logical address H400000. The monitor's stack starts at H402000. Various data is stored from H401000 to H401100, and should not be altered by the user. Finally, the user's system stack pointer starts from H401800.

The monitor sets up memory maps to allocate 64K for each of the 16 process numbers on the board. The first segment of each process is allocated, with the physical address of the memory being H000000 for process 0 (the monitor and system routines--logical segment H40), H010000 for process 1, etc. If the user wishes memory to appear in segment 0 of any processes, he must place memory boards at the proper physical addresses. For example, if the user wanted to have 16K of memory available for process 2 starting at its address H008000, the physical address of the 16K section should be set to be H028000. The user's program can manipulate the memory mapping circuitry as desired to allow different mapping arrangements.

Due to the usage of memory by the monitor, it is suggested that any user programs be located above H2000 in segment 40 or in segment 0 of the process that is being used.

### 3. Operating Instructions

This section of the manual describes the basic operations of the monitor and gives support to the information presented in the Command Reference section.

When the system is initialized (powered-on or reset), the RETURN key on the terminal must be pressed until the initial message is displayed. This allows the monitor to determine the baud rate of the terminal.

The monitor facilitates debugging of user programs with several features. First, the entire status of the Z8000 CPU (including the registers, program counter, and flags) is kept in RAM, and loaded into the processor during execution and single-stepping.

With this arrangement, the user has access to a complete processor, totally independent of the monitor. Program execution can be interrupted at any time, memory or registers changed, and execution resumed exactlly where it left off. The system stack pointer is kept seperate for the monitor's execution and for when the user program is executing. The user's system stack pointer is setup to be at H401800 initially, and a return address to the monitor is pushed onto this stack. The monitor uses a stack starting at H402000, keeping full independence of the monitor and the user program.

The monitor displays a stack warning message when the user's system stack goes below address H401200. This can be an indication of a runaway stack.

For systems which contain a Central Data Floppy Disk Controller board or a Cartridge Disk Controller board, two interrupt service routines are provided to clear initial interrupts which occur from these boards. Note that under normal conditions a bootstrap PROM is used with these disk controller boards, but in cases where the user desires the monitor to be entered on initialization in a disk system, these routines prevent permanent interrupts from occuring on VI6 (for floppy disks) or VI5 (for cartridge disks).

At any time, the user can press Control-X to return to command mode in the monitor (unless vectored interrupts are

disabled). If a user program was being executed at the time of the keypress, the registers and flags are saved, allowing inspection by the user.

### 4. Command Reference

This section of the manual details the commands for ZMON, showing the format for each command along with a description of its use. The other sections of the manual should be referenced for general information pertaining to all commands.

Command: Set or Display Breakpoints

Format: BR [<addr> [,<repeat>]]

Description: this command is used to set a breakpoint at a specified address or to print the addresses of all currently set breakpoints.

If no parameters are entered with the command, the monitor will display the addresses of all currently set breakpoints. If no breakpoints are set, a message to that effect is displayed.

If a single parameter is entered (the <addr>), the monitor will set a breakpoint at that address. The next time that address is executed, a return will be made to the monitor, the breakpoint will be cleared, and single-step mode will be entered.

If two parameters are entered, the first is the address of the breakpoint, while the second indicates the number of times that address should be executed before a return to the monitor is done. This allows the user to breakpoint on a specified iteration of a loop without having to repeatedly set breakpoints and execute the loop. Note that the number entered is the number of times the address will be allowed to execute, and that a value of zero is equivalent to a simple (non-repetitive) breakpoint.

For repetitive breakpoints, it should be noted that the loop will not execute at its full speed since each time the specified address is reached a routine in the monitor must be called to determine if the proper number of loops has been done.

Up to four breakpoints can be set at any one time. If the user tries to set more, the command is aborted with an error message.

When a breakpoint is set, the monitor inserts a H7FFF opcode into the program. This is a system call instruction, which causes a return to the monitor upon its execution. Obviously, breakpoints must be set on the first word of an instruction, or unpredictable results may occur.

Command: Change Memory Locations

Format: CH [<addr>]

Description: this command allows the user to inspect and change individual bytes of memory. The starting address can be specified in the command or can default to the current program counter value.

After the command is entered, the monitor displays the address on the next line of the terminal along with the data byte at that address. During the entire change routine, the user has three options. The first is to leave the current byte as displayed and go to the next byte. The next option is to store the current byte as displayed and return to command mode. The last option is to change the current byte. In all cases, the term "current byte" means the last byte printed by the monitor.

The first option is exercised by hitting the space bar, while the user can return to command mode by hitting RETURN or ESC.

To change a byte of data, the user simply enters the new data. As individual characters are entered, the old contents of the byte are shifted out to the left, with the new character replacing the right-most digit. It is important to note that the data displayed on the screen is exactly the same as the data stored in memory at any time. Therefore, there is no guesswork involved in whether a character entered was stored or not.

Every 16 characters, the change routine starts a new line with the updated address and corresponding data byte.

Command: Clear Breakpoints

Format: CL [<addr>]

Description: this command clears breakpoints previously set by the user. If no address is specified, all breakpoints are cleared, while if an address is specified, it will be the only one cleared.

For each breakpoint that is cleared, a message is displayed showing the address. If no breakpoints were cleared, no message is displayed.

Command: Compare Memory Blocks

Format: CO <start addr 1>,<end addr 1>,<start addr 2>

Description: this command compares two memory blocks and displays any differences. The comparison is done on a byte-for-byte basis, and any non-equal locations are displayed with their addresses and data.

The first memory block is specified fully in the command (both start and end addresses), while only the start address of the second block is given—its length implied to be the same as the first block's.

Command: Dump Memory

Format: DU [<start addr> [,<end addr> [,<page size>]]]

or

DU @<double-register> [.<page size>]

Description: this command allow the user to display blocks of memory on the terminal. The command operates with two formats: with the starting address specified in the command or or by a Z8000 register value.

With the first format, the user can specify the starting address or let it default to the current program counter value (if no parameters are entered with the command). The second parameter is the end address, which gives the maximum address to be dumped. If this parameter is not specified, the dump will continue until the end of memory. The last parameter gives the number of lines to dump at one time, with a pause for a keypress done at the end of each group. If the last parameter is not given, the entire dump is done without stopping.

The dump is done with the beginning address for each line displayed on the left side of the screen, the hex representation displayed next to the address, and the ASCII representation displayed on the right side of the screen. If the byte is not a valid (printable) ASCII character, a period (.) is displayed for the location.

The second format of the command allows the start address to be specified as one of the Z8000's register pairs. This feature is very useful when the user wants to display data which is referenced through a register in his program without having to diplay the register contents and dump using that value as the address. Any normal register pair may be used (RRO-RR14) as well as the system stack-pointer (SR14).

With either form of the command, if only the start address is specified, one line will be dumped at a time, and a keypress is required to display the next line.

At any time during a dump the user can return to command mode by hitting ESC, and a dump can be paused by hitting any other key on the terminal. To resume the dump, another key must be pressed. Command: Execute

Format: EX [<addr>]

Description: with this command, control is transferred to the location specified. If no address is given in the command, execution begins at the current program counter location. Command: Initialize Memory by Bytes

Format: IN <start addr>,<end addr>,<value>

Description: All of the locations between the two addresses specified (inclusive) are set to the value given as the third parameter. The initialization is done on a byte basis, so that each byte within the range will have the same value after the operation.

The Initialize Memory by Words command can be used to initialize memory on a word basis.

Command: Display Input Port Value

Format: IP <port>

Description: this command reads the Multibus input port specified and displays the data found. The command does a byte input instruction to the address, so any word I/O addresses on the bus cannot be read using the command.

The value of  $\langle port \rangle$  must be in the range of 0-7FF, since those are the limits of the I/O addresses for the Multibus supported by the Memory Management Z8000 CPU board.

Command: Initialize the System Stack Pointer

Format: IS

Description: This command changes register-pair SR14 to the address value H401800, and pushes a return address on the stack at that location. This return address allows the user to execute subroutines and return to the monitor without making special provisions.

The final value for register S14 will be  $\rm H4000$ , and S15 will be  $\rm H17FC$  after the address is pushed onto the stack.

Command: Initialize Memory by Words

Format: IW <start addr>,<end addr>,<value>

Description: this command loads the value specified into all of the memory locations between the start and end addresses (inclusive). The value is taken to be a 16-bit number, and it is written to every even address in the range.

This command differs from the Initialize Memory by Bytes command in that it allows both halves of a word to contain different values.

Command: Load

Format: LO [<baud rate>]

Description: this command allows the system to load data to or from another computer through a serial channel. The other computer specifies all addresses, block lengths, and the direction of transfer. The format of the block transferred is:

Bytes	Function
1	Segment of transfer address
2	Transfer address
2.	Length of transfer (0=last block)
1	Direction (0=into Z8000, NZ=out of Z8000)
X	(len) data bytes
•	
•	additional blocks
•	

The baud rate to be used can be entered with the command, with the available speeds being 300, 600, 1200, 2400, 4800, 9600, and 19200. If no baud rate is specified, the rate stays the same as previously set, or 4800 baud if none was ever specified.

Command: Move Memory Blocks

Format: MO <start addr 1>, <end addr 1>, <start addr 2>

Description: this command moves a block of memory to another location. The destination of the block is delimited by the first two addresses entered with the command. The address of the source is given by the third parameter.

Provisions are made in the monitor to guarantee that data will be moved properly to prevent data from being overwritten before it is moved. This means that the data is moved the start to the end in some cases, while it is moved from the end to the start in others.

Command: Memory Test

Format: MT <start addr>, <end addr>

Description: the command performs a memory test on the selected block of memory. The test consists of writing an incrementing pattern to the memory block (excluding zeros) and then reading the data back, checking each location. Any errors are printed on the terminal, with an indication of the address, the data read, and the data expected.

The memory test can be stopped at any time by hitting the ESC key.

Command: Write an Output Port

Format: OP <port>, <data>

Description: this command writes the specified byte of data to an output port of the Multibus. The port address given must be in the range of 0-7FF, since that is the extent of the Multibus I/O ports supported by the Memory Management Z8000 CPU board. Also note that only byte output operations are suported, so the data written is duplicated on the high and low Multibus data lines.

Command: Write Process Number

Format: PR <number>

Description: this command writes the number specified into the on-board process number port. After the new process number is written, the lower 64 segments (HO-H3F) will be mapped for the new process, and the memory mapped to all other processes will be inaccessable.

The process number must be in the range of 1 to 15, since the memory maps for process number 0 are used for system memory (segments H40-H7F).

Note that the monitor maps 64K of physical memory to the first segment (0) of each process. The memory for process 1 should be addressed starting at H010000, while the memory for process 2 should be at H020000, etc. It is up to the user to place memory at the addresses needed by his programs, and it should be noted that the re-mapping of any memory location is easily done by writing to the memory map I/O ports on the board.

Complete information concerning the process number and the memory mapping system can be found in the Hardware Reference Manual for the Z8000 CPU Board.

Command: Change and Display Registers

Format: RE [<reg>,<value> [<reg>,<value>...]]

Description: this command allows the user to change and display the Z8000's registers, flags, and program counter. Also displayed are the first four words located at the program counter address.

The parameter  $\langle reg \rangle$  can be one of the standard registers of the Z8000, as well as any of the following flags and registers:

Symbol	Flag/Register Represented
SEG S/N SE VIE	SEG bit of the FCW S/N bit of the FCW SE bit of the FCW VIE bit of the FCW
NVIE	NVIE bit of the FCW
C	Carry Flag
Z S	Zero Flag Sign Flag
P/V	Parity/Overflow Flag
DA H	Decimal Adjust Flag Half-Carry Flag
FCW	FCW of Z8000 (all 16-bits)
S14	R14'System Mode R14
S 15 S R 1 4	R15'System Mode R15
PC	RR14'System Mode RR14 Program Counter

Note that only byte, word, and double-word registers are alterable with the command (quad-word registers must be altered with two operations). Examples of normal register representations would be RLO, RHO, R3, and RR10.

Note that any number of registers may be altered with one command, limited only by the command buffer length.

After the registers which were specified in the command (if any) have been altered, the registers and flags are displayed. The flag bits are printed with their representative name (see above) if the flag is set, while spaces are printed if the flag is not set. Also, a stack warning message is printed if the system stack pointer falls below address H401200. This may be an indication of a runaway stack.

Command: Single-Step

Format: ST [<addr>]

Description: this command allows a user to single-step through a series of instructions, with the registers and flags displayed before each instruction is executed. When the registers are displayed, the user can press the ESC key to return to command mode. Any other keypress will cause the instruction to be executed, and the next instruction will be displayed along with the updated registers.

Since the single-step function uses the non-vectored interrupt input to the processor, the user cannot step through routines which generate non-vectored interrupts of their own (from the two timers or the 9511 on the hoard). All other interrupt types, including the more common vectored interrupt, are allowed as usual during the single-stepping process.

If a breakpoint is single-stepped, the normal breakpoint message will be displayed, and the system will stay in step mode. If a repetitive breakpoint is single-stepped, the program will be executed until the specified number of loops has been counted, and then normal breakpoint procedures will be followed.

### 5. System Calls

This section of the manual details the system calls which the monitor supports for the user. Any program can use the "SC" instruction to call one of these routines, and the user can implement additional routines as described below.

To allow user-supplied system calls, the user must change two areas of memory. The first, address H40108C should be changed to be the address (in standard Z8000 form) of the user supplied system call routine. This routine will have to inspect RL4 to determine the number of the routine that is being called (which is specified in the instruction).

The second location that the user must modify is address H40108A, with that byte changed to the maximum system call number serviced by the routine. The next location should contain the minimum number that is serviced by the user supplied routine. Note that the monitor supports system call numbers 0-6 at the present time, with expansions likely in the future. It is suggested that user-supplied system calls start with number H80.

System Call: Return to the Monitor

Number: 0

Description: this call returns the program to the monitor's command mode. All registers at the time of the call are saved in the monitor's register area.

System Call: Check Input Key Status

Number: 1

Description: this call returns status of Z if no key is waiting, and returns NZ if a key was ready. If a key was waiting from the terminal, it is returned in RLO. If no key was waiting, RLO will have unpredictable data. All other registers are saved.

System Call: Get Input Key

Number: 2

Description: this call waits for a key to be input from the terminal, and returns with the key in RLO. No other registers are altered, and the routine always returns with a status of NZ.

System Call: Editing Input Function

Number: 3

Description: this call allows the user to use the monitor's command entry routine to get data from the terminal. This allows full editing functions, including copy and backspace.

The starting location for the buffer to hold the input data should be given in RR2. The maximum number of characters should be in R1, with all registers saved by the routine. When the routine returns to the calling program, the buffer will have the data entered by the user followed by a carriage return code (HOD).

If the user hits the ESC key while in the routine, the flags return NZ, while if the user hits RETURN to finish, the flags will return Z.

System Call: Write a Character to the Terminal

Number: 4

Description: this command sends the character contained in RLO to the terminal. If the USART is busy transmitting a character, the routine waits until it can send the character. All registers are saved.

System Call: Display Hex Byte

Number: 5

Description: this call prints two hex characters on the terminal, representing the byte in RLO. All registers are saved.

System Call: Display Hex Word

Number: 6

Description: this call prints four hex characters on the terminal, representing the word value of RO. All registers

are saved.

### **Central Data Corporation**

### **Engineering Change Order**

Product Z8000 Monitor	Number_ 1008
Part Number ZMONO, ZMONO, or ZMONI	Previous No. 1000
Date 04-16-80 Approved J. Rolett	

### Parts List Changes:

N/A

### **Manufacturing Specification Changes:**

N/A

### **Testing Specification Changes:**

N/A

### **Description of Problem/Solution or Improvement:**

To allow all registers to be preserved during user system calls, the system call number is stored in location H401091 rather than in RL4. The user's system call handler should read this location to determine what action to take. All other setup requirements for user system calls remain the same.

### Schematic Changes:

N/A