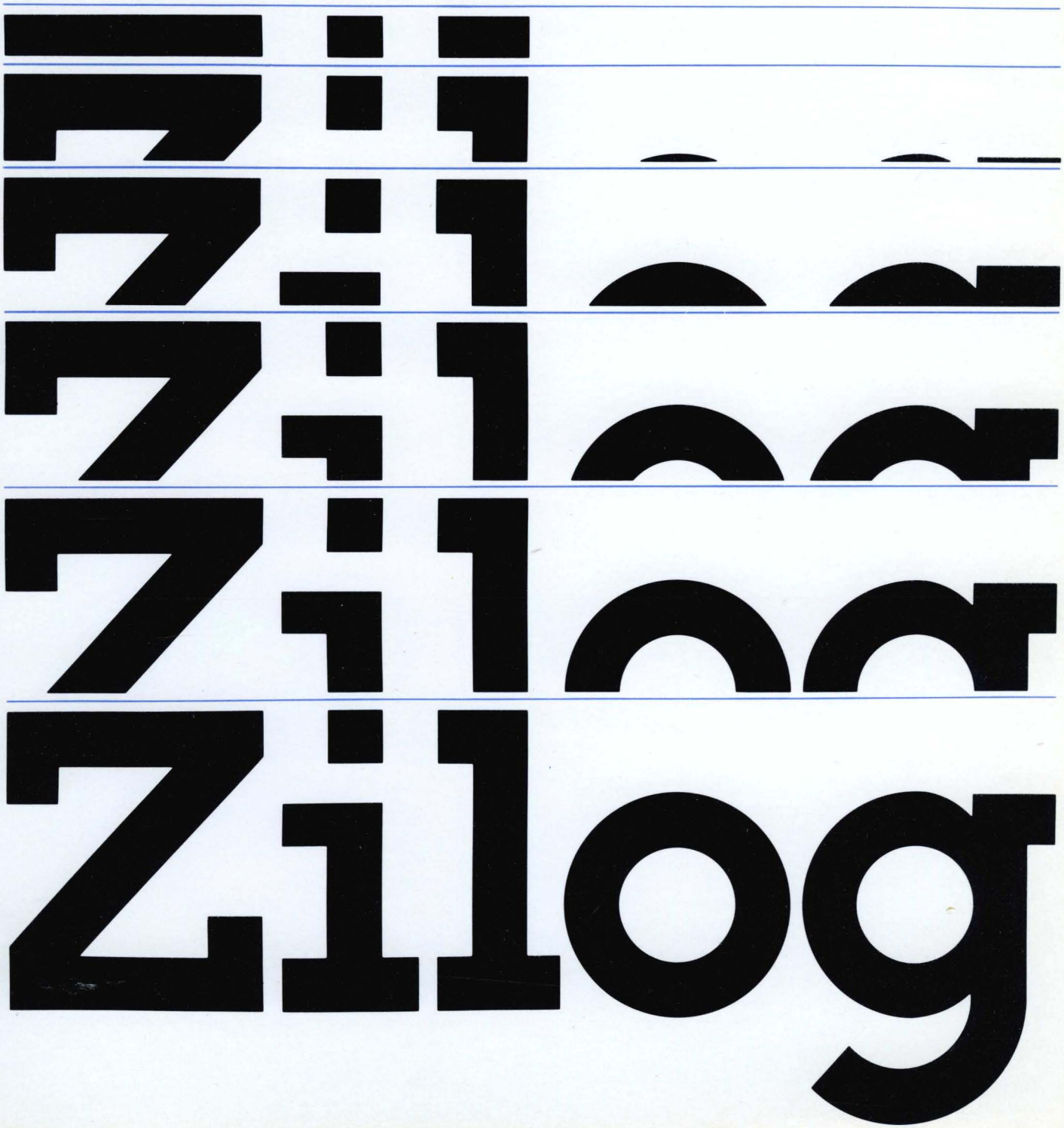
RIO FILE DEBUGGER

Zilog

Reference Manual

June 1979



03-3038-02

June 1979

RIO FILE DEBUGGER

Reference Manual

June 1979

READER COMMENTS

Your comments concerning this publication are important to us. Please take the time to complete this questionnaire and return it to Zilog.

Title of Publication: _____

Document Number: _____

Your Hardware Model and Memory Size: _____

Describe your likes/dislikes concerning this document:

Technical Information: _____

Supporting Diagrams: _____

Ease of Use: _____

Your Name: _____

Company and Address: _____

Your Position/Department: _____

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 475 CUPERTINO, CA

POSTAGE WILL BE PAID BY

Zilog
Systems Publications Department
10411 Bubb Road
Cupertino, California 95014

PREFACE

This manual is a user manual for the FILE.DEBUG and DISK.FILE.DEBUG utilities, used with the RIO Operating System on the Zilog Microcomputer System (MCZ). It provides detailed discussions of both utilities, including maintained variables and their use in commands.

Other related Zilog documentation includes:

Z-80 RIO Operating System User Manual

MCZ PROM Software Reference Manual

This manual makes use of the following conventions of notation:

- Optional portions of a command or command modifier are enclosed in square brackets, e.g., [disk address].
- The symbol for logical OR, the vertical bar (|), is used between command options when any one of the command options can be specified.
- All memory addresses and constants referring to memory allocation are given in hexadecimal.

CONTENTS

SECTION 1

THE FILE.DEBUG UTILITY

1.1 General Information

The FILE.DEBUG Utility is designed to help repair floppy-disk file structure damage resulting from "soft" (i.e., transient or software-produced) failures. The FILE.DEBUG Utility is used with system floppy disks, and can be invoked to perform the following tasks:

- Read records directly from, and write records directly to the diskette;
- Locate a file name in the directory and examine descriptor record contents;
- Trace through a file or sequence of addresses until a null Fore Pointer is reached;
- Identify all disk sectors of given contents;
- Use the PROM Debugger to do a direct edit of buffer contents prior to write operations.

WARNING: IT IS EXTREMELY EASY TO OBLITERATE AN ENTIRE FILE OR DISK THROUGH CARELESS USE OF THE COMMANDS ASSOCIATED WITH THE FILE.DEBUG AND DISK.FILE.DEBUG UTILITIES.

The FILE.DEBUG Utility maintains the following set of variables, referenced in subsequent discussion.

Variable	Description
DA	Current disk address; updated on all READ operations. All disk addresses are two bytes: the first byte is the track address, and the second byte is the drive/sector address. Drive=bits 7-5, sector address=bits 4-0. The sector addressed by DA is indicated by '(DA)'.

Variable	Description
BACK	Back Pointer of (DA); updated on all READ operations or, optionally, on a WRITE operation.
FORE	Fore Pointer of (DA); updated on all READ operations or, optionally, on a WRITE operation.
RL	Record length; set to record length of opened files, or used as an option of READ and WRITE operations.

Addresses listed by FILE.DEBUG are given without the drive specified (bits 7-5 of the second byte = 0). A drive name specified with an address as part of a command (e.g., READ or WRITE), or as part of a file name in a command (e.g., TRACE or OPEN), causes that drive name to be used in subsequent operations until it is over-ridden with a new drive name, possibly in the same manner. As a consequence, READ or WRITE operations (by sector address) done on drives other than Drive 0 should include the Unit Number in the disk address (DA).

1.2 FILE.DEBUG Commands

* USER NOTE: The link address of MCZ-1/20-type systems is 4400. The link address of MCZ-1/35 systems is 2A00.

READ [disk address [record length]]

Reads sectors of a specified record length (default record length = 128) starting at FORE (or an optional specified disk address) into a buffer starting at link address + C00.* Each READ Operation results in a display of values for DA, RL, BACK, and FORE, according to the following format:

DA=nnnn RL=nnnn BACK=nnnn FORE=nnnn

<CR>

A Carriage-Return keystroke on a null line is interpreted as a READ command with no parameters.

^

A Circumflex keystroke (^) is interpreted as a READ command with no parameters, except that the READ direction is reversed, that is, it reads starting at BACK.

WRITE [disk address [record length [back pointer [fore pointer]]]]

Writes sectors of a specified record length (default record length = 128) starting at DA (or an optional specified disk address) from a buffer starting at link address + C00.* Before each WRITE operation is begun, the system issues a prompt in the form:

DA=nnnn RL=nnnn BACK=nnnn FORE=nnnn?

A "Y" response causes a WRITE operation; anything else terminates the command.

OPEN filename

Searches the directory for the specified file and reads the descriptor record. The specified file may be qualified by a drive name (default = Drive 0). Each READ operation that occurs during the directory search causes the system to output the display line described above for READ operations. If the specified file is found, an additional display line is output in the form:

DSA=nnnn FRA=nnnn LRA=nnnn

giving the disk address of the Directory Sector Address, First Record Address, and Last Record Address, respectively. The Descriptor Record is read into the buffer starting at link address + C00.*

TRACE [filename |^]

Performs successive READ operations until a null Back Pointer, Fore Pointer or Pointer Error is found. If a specified file is added to the command, an OPEN operation is performed before the first READ operation. The "T" or "T filename" command traces forward; the "T ^" command traces backward.

LIST

Displays the current values of DA, RL, BACK, and FORE in the same form described above for READ operations.

DEBUG

Enters the PROM Debugger; entering a "Q" command while in the PROM Debugger returns control to FILE.DEBUG.

QUIT

Returns control to the RIO Operating System.

SEARCH [disk address]

Searches the current drive for sectors that match the MATCH buffer (link address + B80H) after masking with the MASK buffer (link address + B00H). The MATCH and MASK buffers are initialized to zero. Setting a bit in the MASK buffer (e.g., to 1) causes that bit in the sector read from the disk to be matched against the corresponding bit in the MATCH buffer. The disk addresses of successful matches are listed.

If a disk address is specified, the match criterion is not sector contents, but whether or not the specified disk address parameter is equal to the Fore or Back Pointers of the sector read. This type of SEARCH operation does not tell whether the match was a Fore Pointer or a Back Pointer.

NOTE: The SEARCH operation is a lengthy process, but may be prematurely terminated by an Escape (ESC) keystroke.

SECTION 2

THE DISK.FILE.DEBUG UTILITY

2.1 General Information

The DISK.FILE.DEBUG Utility is designed to help repair file structure damage resulting from "soft" (i.e., transient or software-produced) failures. The DISK.FILE.DEBUG Utility is used with cartridge disks, and can be invoked to perform the following tasks:

- Read records directly from, or write records directly to the disk;
- Locate a file name in the directory and examine descriptor record contents;
- Trace through a file or sequence of addresses until a null Fore Pointer is reached;
- Use the PROM Debugger to do a direct edit of buffer contents prior to write operations.

WARNING: IT IS EXTREMELY EASY TO OBLITERATE AN ENTIRE FILE OR DISK THROUGH CARELESS USE OF THE COMMANDS ASSOCIATED WITH THE FILE.DEBUG AND DISK.FILE.DEBUG UTILITIES.

The DISK.FILE.DEBUG Utility maintains the following set of variables, referenced in subsequent discussion.

Variable	Description
DA	Current address (all disk addresses are three bytes long; the most significant three bits specify the unit number); updated on all READ operations. '(DA)' refers to the sector addressed by DA.

Variable	Description
BACK	Back Pointer of (DA); updated on all READ operations or, optionally, on a WRITE operation.
FORE	Fore Pointer of (DA); updated on all READ operations or, optionally, on a WRITE operation.
RL	Record length; set to record length of opened files, or used as an option of READ and WRITE operations.
FILE ID	The four byte File Identification field from the Sector Header; updated on all READ operations or, optionally, on WRITE operations.

Addresses listed by DISK.FILE.DEBUG are given without the drive specified (bits 7-5 of the first byte = 0). A drive name specified with an address as part of a command (e.g., READ or WRITE), or as part of a file name in a command (e.g., TRACE or OPEN), causes that drive name to be used in subsequent operations until it is overridden with a new drive name, possibly in the same manner. As a consequence, READ or WRITE operations (by sector address) done on drives other than Drive 0 should include the Unit Number in the disk address (DA).

2.2 DISK.FILE.DEBUG Commands

* USER NOTE: The link address of MCZ-1/20-type systems is 4400. The link address of MCZ-1/35 systems is 2A00.

READ [disk address [record length]]

Reads sectors of a specified record length (default record length = 512) starting at FORE (or an optional specified disk address) into a buffer starting at the link address + F00.* The first nine bytes are the Back Pointer, the Fore Pointer, and the File ID of the sector, respectively. The 512 bytes of actual data follows. Each READ operation results in a display of values for DA, RL, BACK, and FORE, according to the following format:

DA=nnnn RL=nnnn BACK=nnnn FORE=nnnn

WRITE [disk address [record length [back pointer [fore pointer [file id]]]]]

Writes sectors of a specified record length (default record length = 512) starting at DA (or an optional specified disk address) from a buffer starting at the link address + F00.* The first ten decimal bytes are the sector header (Back Pointer, Fore Pointer, and File ID) information. Before each WRITE operation, a prompt is issued, as follows:

DA=nnnnnn RL=nnnn BACK=nnnnnn FORE=nnnnnn FILEID=nnnnnnnn?

A "Y" response causes a WRITE Operation; anything else terminates the command.

OPEN filename

Searches the directory for the specified file, and reads the descriptor record. The specified file may be qualified by a drive name (default = Drive 0). Each READ operation that occurs during the directory search causes the system to output the display line described above for READ operations. If the specified file is found, an additional display line is output in the form:

DSA=nnnnnn FRA=nnnnnn LRA=nnnnnn

giving the Directory Sector Address, First Record

Address, and Last Record Address, respectively. The Descriptor Record is read into the buffer starting at the link address + F00.*

TRACE [filename|^]

Performs successive READ operations until a null Back Pointer, Fore Pointer, or Pointer Error is found. If a specified file is added to the command, an OPEN operation is performed before the first READ Operation. The "T" or "T filename" command traces forward; the "T ^" command traces backward.

LIST

Displays DA, RL, BACK, FORE, and FILE ID values for the current disk address in the same form described above for READ operations.

<CR>

A Carriage-Return keystroke represents a null line, and is interpreted as a READ command with no parameters.

^

A circumflex keystroke (^) is interpreted as a READ command with no parameters, except that the READ direction is reversed, that is, it reads starting at BACK.

DEBUG

Enters the PROM Debugger; entering a "Q" command while in the PROM Debugger returns control to DISK.FILE.DEBUG.

QUIT

Returns control to the RIO Operating System.

Zilog, Inc. 10460 Bubb Road, Cupertino, California 95014 Telephone (408) 446-4666 TWX 910-338-7621

Printed in USA