

# Z80-RIO



## Product Specification

FEBRUARY 1978

The RIO operating system with relocatable modules and I/O management is a general purpose computing system with an architecture that is designed to facilitate the development process, provide straight-forward linkage to various system routines, and enable expansion of system features to meet the particular needs of individual users.

### Features

#### ■ OS EXECUTIVE

- Maps requests for operations on logical units to specific device handling programs
- Commands may be issued to OS from the console or by an executing program
- Any number of user defined commands may be added to the system
- Command sequences may be recorded in files and executed as a group

#### ■ ADVANCED ASSEMBLER

- Relocatable or Absolute Object Code format
- External Symbol references
- Global symbol definitions
- Macros and Conditional Assembly
- Paged Symbol table permits assembly of arbitrarily large programs in standard memory
- Include directive permits additional files to be merged into the source program at assembly time

#### ■ LINKER

- Assigns absolute addresses to program modules
- Resolves External references
- Permits overlays or memory gaps
- Produces memory map and Global address table

#### ■ TEXT EDITOR

- Paged work space permits any size of file to be edited
- Automatic backup for extra protection
- Access to other disk files during editing
- All edit operations for locating and modifying lines within a file are based on character string matching

#### ■ PROM MONITOR

- Full machine language Debug package
- Low level device handlers for system console and floppy disk
- Bootstrap loader for easy system entry

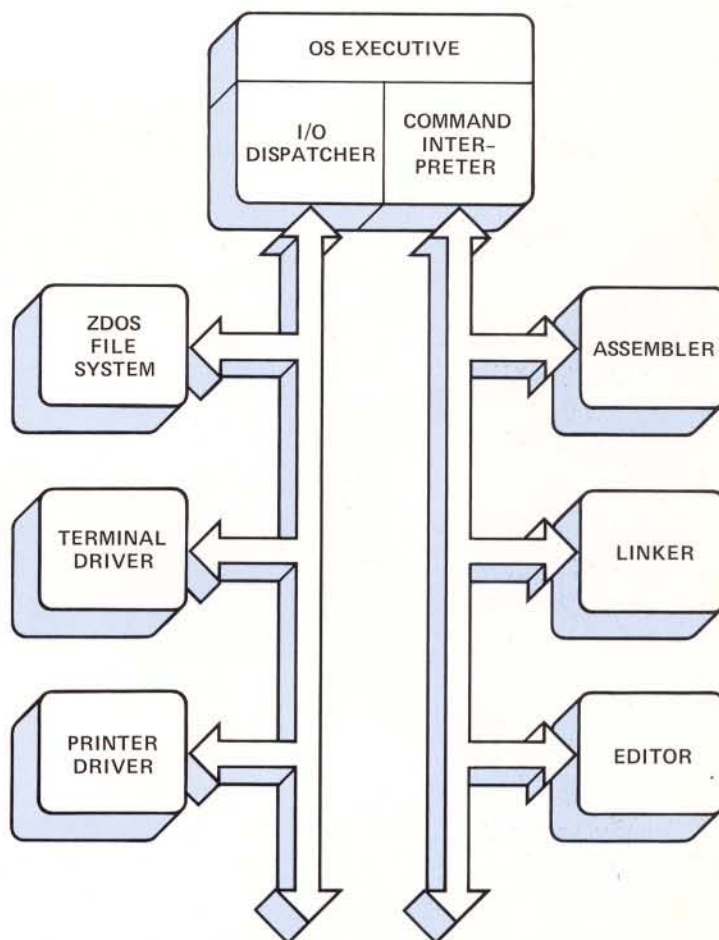
#### ■ TABBING

- Supported throughout the system
- Compact storage of tabbed text on disk

#### ■ ZDOS II DISK FILE SYSTEM

- Allocates all disk space automatically on an "as needed" basis
- Supports sequential access to disk files or direct access to any specific disk address
- Up to eight (8) floppy disk drives supported

- Entire System operates in 32K bytes of memory.



## RIO Disk-Resident OS Executive

The OS Executive serves two major functions with the system: interpretation and management of command execution, and mapping of I/O requests to device handlers.

### THE FOLLOWING ARE OS SYSTEM COMMANDS:

**ACTIVATE** makes a device known to the system by including it in the active device table.

**ALLOCATE** allocates, if possible, a memory segment consisting of one or more 128 byte blocks within a specified range.

**ASM** assembles user defined source programs.

**BRIEF MODE** commands are not echoed on the console output device as interpreted.

**CAT** prints a list of files in ZDOS directories.

**COPY** copies a source file to a destination file.

**COPY. DISK** directly copies and/or verifies the contents of one disk to another.

**DATE** displays or changes current date.

**DEACTIVATE** closes all units linked to a device and makes device unknown to system.

**DEALLOCATE** deallocates a specific memory segment of a given size.

**DEBUG** enters the PROM Debugger.

**DEFINE** creates link between logical unit and device.

**DELETE** deletes files from device directories.

**DISPLAY** prints an interpretation of the memory allocation map.

**DO** executes commands from an ASCII file.

**DUMP** provides a hexadecimal dump and the ASCII representation of a specified file's contents.

**ECHO** prints messages on console.

**EXTRACT** lists record count, record length, entry point, and memory utilization of a procedure file.

**EDIT** enters the text editor.

**FORCE** inhibits the error return which would normally be the result of attempting to load a file within previously allocated memory.

**FORMAT** formats a diskette.

**IMAGE** copies core images to a specified procedure type file.

**INITIALIZE** sends an initialize request to the master device or to a specified currently active device.

**LADT** lists the contents of active device table which includes currently active devices and their entry points.

**LINK** processes one or more object modules and generates a single program in the form of an executable procedure file.

**MASTER** makes a currently active device the default source for unqualified file names.

**MEMORY** displays allocated and unallocated memory areas.

**MOVE** The file system directories are searched and files which match the specified options are copied from one device to another.

**PAUSE** waits for console input, optionally aborting command string.

**RELEASE** deallocates space allocated as a result of loading, but not executing, an external command file.

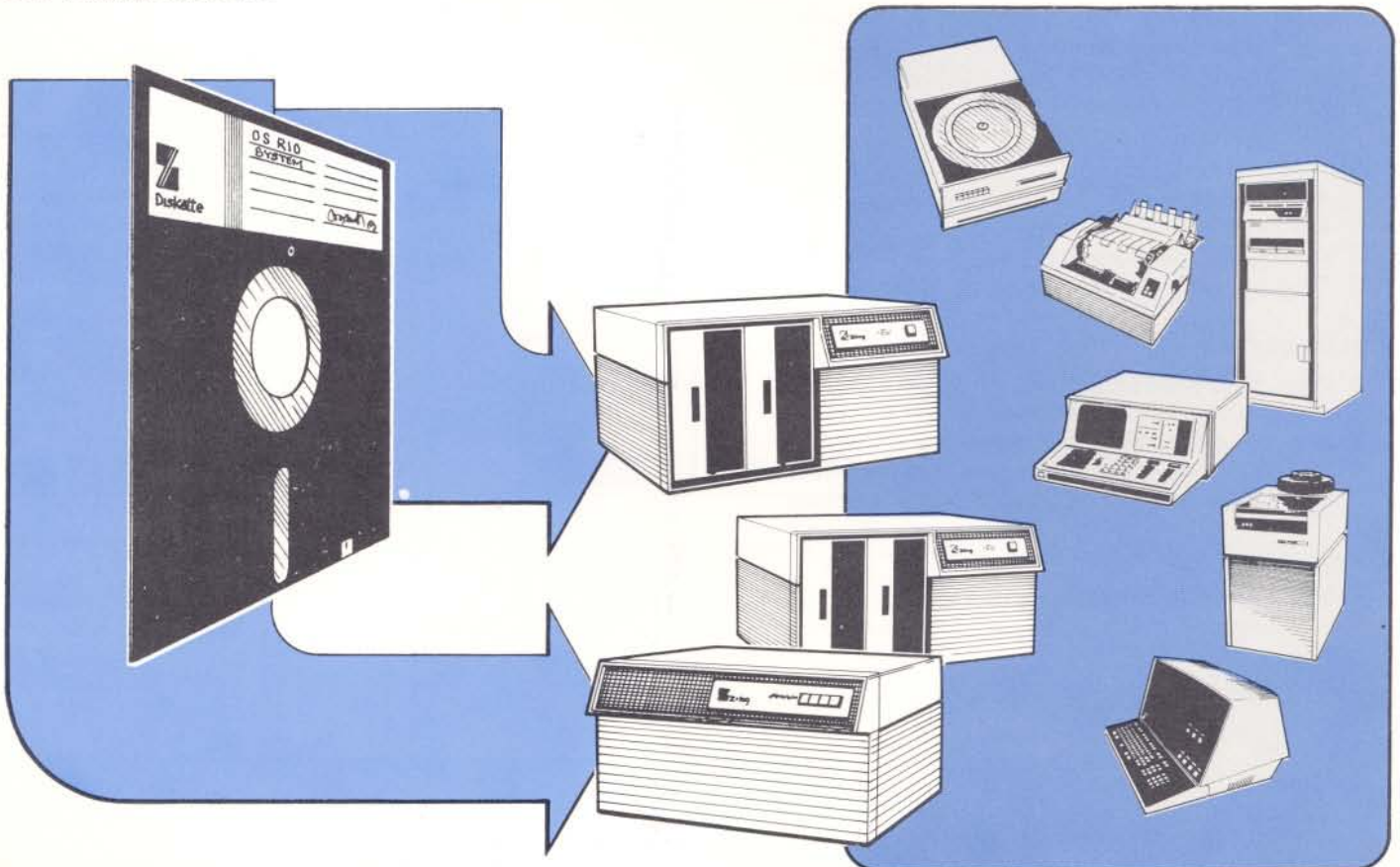
**RENAME** renames a file or a diskette.

**SET** sets various system parameters to specified values.

**STATUS** lists disk sector usage on console.

**VERBOSE MODE** echoes command strings as interpreted.

**XEQ** begins execution at specified memory locations or optionally, the starting address of last loaded file.



## RIO Relocating Assemblers

---

The relocating assembler ASM accepts a source file (a symbolic representation of a portion of a program in Z80 Assembly language) and translates it into an object module. It may also produce a listing file which contains the source, assembled code, and optional symbol cross-reference. The features of the relocating assembler include macros and conditional assembly, as well as the capability to produce either relocatable or absolute object code.

Relocation refers to the ability to bind a program module and its data to a particular memory area at a time after the assembly process. The output of the assembler is an object module which contains enough information to allow the linker to assign that module a memory area. Since many modules may be linked together to form a complete program, a need for inter-module communication arises. For example, one module may contain a call to a routine which was assembled as part of another module and may be located in some arbitrary part of memory. Therefore, the assembler must provide information in the object module which allows the linker to link inter-module references.

There are several major advantages to using the relocating assembler, as compared to the absolute assembler:

- If errors are found in one module, only that one module needs to be re-assembled and then re-linked with the other modules, thus increasing software productivity.
- Programs can be structured into independent modules, coded separately, and assembled even though the other modules may not exist yet. This separation of functionality also enhances program maintenance and verifiability.
- Libraries of commonly used modules can be built and then linked with programs without requiring their re-assembly.
- Communications between overlay segments can be achieved through methods similar to normal (non-overlay) inter-module references.
- Assignment of modules to memory areas can be handled by the linker, rather than requiring the programmer to assign fixed absolute locations via the "ORG" pseudo-op and thus modules can be relocated without requiring re-assembly.

## Linker

---

The function of the linker is to process one or more object modules created by the Assembler or any other language translator which generates relocatable object modules, and output a single program in the form of an executable procedure file. The Linker provides relocation of modules and resolves inter-module references to allow linking of separately assembled modules. In addition, a load map file and a binary symbol table file may be created.

The Linker interacts with the RIO operating system to permit the user complete control over the routing of all input and output during linkage. The Linker makes use of both the console device and disk files for I/O.

### FEATURES

- Creates a single segmented procedure type file. Code may be assigned to any location in memory, and up to 20 different segments of code can be generated which allows for discontinuous sections of memory which are not overlaid when the procedure file is loaded.
- Provides ability to link both absolute and relocatable modules together.
- Allows optional assignment of relocatable modules to specific absolute origins.
- Permits specification of the execution starting address as either a hexadecimal number or a GLOBAL symbol.
- Checks for multiply-defined GLOBALs and unresolved EXTERNALs.
- Provides the capability to specify "link-only" modules, that is, modules which are used to control assignment of relocatable module origins and resolve EXTERNAL references, but are not included as part of the generated procedure file.
- Creates a load map with the assigned origin and length of each module, followed by an alphabetically sorted list of all GLOBALs, their assigned addresses and the module which contains them. The map file also contains a list of any error messages generated during linkage.
- Allows optional creation of a binary symbol table file, structured by module, for use by a symbolic debug package.
- Provides user control over the routing of all input and output during linkage.

## RIO Editor

---

The Editor is a line editor with string handling capability and automatic interface to the disk. However, while the Editor works on one line at a time, many commands operate sequentially on several lines. The Editor automatically creates a backup of the user file.

The Editor uses a memory paging technique which allows any size text to be edited. It automatically measures the work space available, and brings segments of the user's text into this work space as edit commands are issued. The user can modify a file by any of the following commands:

**AGAIN** repeats the previous command.

**BOTTOM** establishes the last line of the text as the current line.

**BRIEF** suppresses the echoing printed line produced by most commands.

**CHANGE** locates any specified character string in the current line and replaces it with any new character string. The user can optionally specify the number of lines, from the current line, and the number of occurrences per line to be changed.

**DELETE** removes a specified number of lines from the file.

**FIND** locates the first line containing a specified string, beginning in column one, and establishes it as the current line.

**GET** inserts a block of text from a file following the current line.

**GOTO** establishes the specified line as the current line.

**INPUT** places the system in input mode where new lines of text are accepted and inserted at the current location in the text.

**JOIN** immediately executes a series of commands concatenated in a command string (similar to XECUTE when a MACRO has been defined).

**LINENO** prints out the line number of the current line.

**LOCATE** locates the first line containing a specified string and establishes it as the current line.

**MACRO** defines a set of commands, which can then be executed by a single XECUTE command.

**NEXT** moves forward in the text by a specified number of lines.

**PRINT** outputs a specified number of lines to the user console beginning from the current line.

**PUT** writes a specified block of text to a file.

**PUTD** is a variant of the PUT command which deletes from the text the block just placed in a file.

**QUIT** returns control to the OS level. The new version of the file is saved.

**REPLACE** replaces the current line of text with new text from the terminal.

**TOP** establishes the null line preceeding the first line of the text as the current line.

**UP** moves backward in the text by a specified number of lines.

**VERIFY** causes the current line to be printed at the end of certain commands.

**WINDOW** displays the first and last line numbers of the text currently in memory.

**XECUTE** executes the set of commands defined by the MACRO command.

## Debug Commands

---

### IN THE PROMs OF THE RIO SYSTEM:

A complete Debug package is included to support testing of the system under development. The command implementations vary somewhat with the specific hardware environment. The common features include:

**DISPLAY** a single address or a range of addresses. When a single address is displayed, it is also possible to modify that address' contents and/or proceed one byte at a time through memory.

**FILL** a block of memory with a specified byte value.

**GO** provides the ability to continue program execution from the address currently held in the program counter.

**I** permits examination and change of the interrupt enable flip-flop.

**JUMP** starts program execution at a specified address.

**MOVE** memory contents of a specified length from one address to another.

**NEXT** single instruction step or "N" instruction steps with all registers displayed after each instruction is executed.

**QUIT** returns control to the RIO executive.

**REGISTER** permits examination of all registers or examination and change of register contents one at a time.

**SET** permits contiguous memory locations to be set with a series of data words.

**OS** forces a bootstrap of the RIO system.

**GET** loads a memory image file and stores its entry address in the program counter.

**SAVE** records the contents of memory and a program entry address on disk.

## Debug Commands (continued)

### IN THE ZILOG MICROCOMPUTER SYSTEM:

The additional Debug commands are:

**BREAK** permits a breakpoint to be set on the "Nth" execution of an instruction.

**COMPARE** one block of memory to another.

### IN THE DEVELOPMENT SYSTEM:

The additional Debug hardware available permits the implementation of the following commands:

**BREAK** on any memory read, memory write, port read or port write operation. This is done with hardware comparators on the data, address and control busses of the system.

**HISTORY** displays the contents of the Real Time Storage Module. This is a record of the data, address and control bus states during the most recent 256 bus transactions.

**PORT** permits examination and/or modification of port data.

**PULSE** operates like **BREAK** but rather than stopping program execution, a sync pulse at a BNC connector on the back of the system is generated.

**TRACE** sets the Real Time Storage module to monitor specific types of bus transactions. The default is all transactions.

## ZDOS II (Disk File System)

The initial implementation of RIO utilizes floppy disks which may be configured incrementally from one to eight drives, allowing an on-line storage capacity of up to 2.4 megabytes of data.

ZDOS II is responsible for allocating, inserting, deleting and retrieving information. Each volume of information, or disk, contains a directory file which indicates what information is currently resident. Within the directory file are the names of individual files and a pointer to the first record of each file. The attributes associated with a file include such information as the protection associated with that file and the format of the information contained on the file. Each file record contains forward and backward pointers to adjacent records, in addition to the data. These forward and backward links enable sequential movement in either direction from any position in the file and allow insertions and deletions of records by changing the appropriate pointers.

The following are the operations available in the ZDOS II Disk Operating System:

**INITIALIZE.** Logical units map is cleared, allocation maps read and flag words indicate which disks are ready.

**ASSIGN.** Associate the given file name with the given logical unit for subsequent I/O.

**OPEN.** Prepare file for data transfer.

**CLOSE.** Terminate I/O operations for a particular file.

**REWIND.** Position to beginning of the file.

**READ BINARY.** Input from the device the given number of bytes.

**WRITE BINARY.** Output to the device the given number of bytes.

**SKIP FORWARD.** Position the file forward n records from the last record handled.

**DEACTIVATE** deallocates memory assigned to disk allocation maps.

**SKIP BACKWARD.** Position the file backward n records from the last record handled.

**DELETE.** Remove n bytes from the file.

**WRITE CURRENT RECORD.** Rewrite the last record handled. Equivalent to **DELETE**, **WRITE BINARY**.

**READ CURRENT RECORD.** Re-read the last record handled. Equivalent to **SKIP BACKWARD**, **READ BINARY**.

**READ PREVIOUS RECORD.** Position the file one record previous to the last one handled, then read it.

**READ DIRECT.** Read the record whose address is pointed to by the supplemental parameter vector word. This command can be used to implement direct access files.

**ERASE file.** Remove a file from the directory and deallocate its records.

**RENAME FILE.** Alter the directory entry.

**UPDATE DIRECTORY.** Rewrite the information to the directory and allocation map without removing the file from active status.

**DELETE REMAINING RECORDS.** All records from the current one to the end of the file are removed.

**READ AND DELETE.** Starting at the record following the current one, data is transferred from the file to memory and simultaneously removed from the file.

**SKIP TO END.** The file is positioned with the current pointer indicating the last record of the file.

**SET ATTRIBUTES.** The descriptor record is read and updated from the current information and the attributes supplied and rewritten.

**QUERY ATTRIBUTES.** The descriptor record is read and as many bytes of attribute information as have been requested are returned to the user.

## ZILOG SALES REPRESENTATIVES

### EASTERN

#### ALABAMA

Technology Marketing Assoc.  
P.O. Box 4112  
Huntsville, AL 35802  
TEL 205 883 7893

Large Packages:  
Technology Marketing Assoc.  
7910 S. Memorial Parkway  
Suite 1  
Huntsville, AL 35802

#### CONNECTICUT

Datcom Inc.  
1 Evergreen Avenue  
Hamden, CT 06418  
TEL 203 288 7005

ABC Electronic Sales Co.  
26 Sugar Hollow Road  
Danbury, CT 06810  
TEL 203 743 9441  
TWX 510 222 4232

#### DELAWARE

Dyna Rep Company  
P.O. Box 7739  
Newark, DE 19711  
TEL 703 354 1222  
TWX 710 832 0825

#### GEORGIA

Technology Marketing Assoc.  
6009 So. Orange Avenue  
Orlando, FL 32809  
TEL 305 857 3760

#### FLORIDA

\* Dyne-A-Mark Corp.  
1001 N.W. 62nd Street  
Suite 107  
Fort Lauderdale, FL 33309  
TEL 305 771 6501  
TWX 510 956 9872

\* Dyne-A-Mark Corp.  
405 S. Aurora  
Clearwater, FL 33515  
TEL 813 441 4702,3  
TWX 810 866 0438

\* Dyne-A-Mark Corp.  
303 East Semoran Blvd.  
Suite 300P  
Altamonte Springs, FL 32701  
TEL 305 831 2097  
TWX 810 853 5039

#### MASSACHUSETTS/ NH/VERMONT

\* Datcom Inc.  
55 Moody Street  
Waltham, MA 02154  
TEL 617 891 4600  
TLX 92-3462

### NEW JERSEY

ABC Electronic Sales Co.  
121 Oak Street  
Tenafly, NJ 07670  
TEL 201 568 2354  
TWX 510 222 4232

\* Vantage Sales Co.  
21 Bala Avenue  
Bala Cynwyd, PA 19004  
TEL 215 667 0990  
TWX 510 662 5846

#### NEW YORK

ABC Electronic Sales Co.  
2 Willard Avenue  
N. Tarrytown, NY 10591  
TEL 914 631 2650  
TWX 510 222 4232

\* ABC Electronic Sales Co.  
99 Hillside Avenue  
Williston Park, LI, NY 11596  
TEL 516 747 6610  
TWX 510 222 4232

\* Nycom Inc.  
10 Adler Drive  
East Syracuse, NY 13056  
TEL 315 437 8343  
TWX 710 541 1506

#### PENNSYLVANIA

\* Vantage Sales Co.  
21 Bala Avenue  
Bala Cynwyd, PA 19004  
TEL 215 667 0990  
TWX 510 662 5846

#### VIRGINIA/MARYLAND

\* Dyna Rep Co.  
6399 Little River Turnpike  
Alexandria, VA 22312  
TEL 703 354 1222  
TWX 710 832 0825

#### CANADA

\* Munro Electronic Components Ltd.  
2 Keats Dollard Des Ormeaux  
Montreal, Quebec H9H 3M6  
TEL 514 626 6723

### MIDWESTERN AREA

#### ILLINOIS/WISCONSIN

\* Mar-Con Associates, Inc.  
4836 Main Street  
Skokie, IL 60076  
TEL 312 675 6450  
TWX 910 223 3645

#### IOWA

PMA Corp.  
P.O. Box 1090  
Cedar Rapids, IA 52406  
TEL 319 362 9177

Freight to:  
406 1st Street, SW  
Cedar Rapids, IA 52406

### KANSAS

\* PMA Corp.  
P.O. Box 6264  
Overland Pk, KS 66206  
TEL 913 381 0004  
TWX 910 749 6473

Freight to:  
9602 Outlook Drive  
Overland Pk, KS 66207

\* PMA Corp.  
P.O. Box 11252  
Wichita, KS 67202  
TEL 316 264 2662  
TWX 910 741 6851

Freight to:  
230 Laura, Suite 107  
Wichita, KS 67211

#### MICHIGAN

\* Greiner & Assoc.  
15324 E. Jefferson  
Gross Point Pk, MI 48230  
TEL 313 499 0188

#### MISSOURI

\* PMA Corp.  
P.O. Box 1539  
Maryland Hts., MO 63043  
TEL 314 569 1220  
TWX 910 764 0881

Freight to:  
140 Weldon Parkway, Suite 5  
Maryland Hts., MO 63043

#### OHIO/WEST PENN/W. VA.

\* Geotronics  
8180 Brecksville Road  
Suite 202  
Cleveland, OH 44141  
TEL 216 526 2535  
TWX 810 427 9168

Geotronics  
4453 Willowbrook  
Columbus, OH 43220  
TEL 614 457 8472

#### TEXAS

The Thorson Company  
6633 Hillcroft Avenue  
Suite 105  
Houston, TX 77081  
TEL 713 771 3504  
TWX 910 881 3707

\* The Thorson Company  
4445 Alpha Road  
Dallas, TX 75240  
TEL 214 233 5744  
TWX 910 861 4423

The Thorson Company  
300 E. Huntland Drive  
Austin, TX 78752  
TEL 512 451 7527

### CANADA

\* Munro Electronic Components Ltd.  
7171 Torbram Road  
Unit C9  
Mississauga, Ontario L4T 3W4  
TEL 416 676 1042  
TWX 610 492 8970

Munro Electronic Components Ltd.  
90 Smirle  
Ottawa, Ontario K1Y 0S3  
TEL 613 729 1831

### WESTERN AREA

#### ARIZONA

Nakoma  
4425 W. Olive Avenue  
Suite 225  
Glendale, AZ 85302  
TEL 602 931 9103  
TWX 910 950 1271

\* The Thorson Company  
2505 E. Thomas Road  
Phoenix, AZ 85016  
TEL 602 956 5300  
TWX 910 951 0653

#### CALIFORNIA

The Thorson Company  
340 S. Kellogg Avenue  
Suite F  
Goleta, CA 93017  
TEL 805 964 8751  
TWX 910 334 4837

\* The Thorson Company  
520 S. Sepulveda Blvd.  
Suite 405  
Los Angeles, CA 90049  
TEL 213 476 1241  
TWX 910 342 6891

\* The Thorson Company  
625 Ellis Street  
Mountain View, CA 94043  
TEL 415 964 9300  
TWX 910 379 6598

The Thorson Company  
17775 Main Street  
Irvine, CA 92714  
TEL 714 557 4460

\* Littlefield and Smith Assoc.  
11772 Sorrento Valley Road  
Suite 235  
San Diego, CA 92121  
TEL 714 455 0055  
TWX 910 322 1730

#### COLORADO

\* The Thorson Company  
5290 Yale Circle  
Denver, CO 80222  
TEL 303 759 0809  
TWX 910 931 0429

#### OREGON

J.J. Backer Company  
2035 S.W. 58th Street  
Portland, OR 97221  
TEL 503 297 3776

#### UTAH

The Thorson Company  
3691 West 1987 South  
Salt Lake City, UT 84104  
TEL 801 973 7969

#### WASHINGTON

J.J. Backer Company  
221 West Galer  
Seattle, WA 98119  
TEL 206 285 1300  
TWX 910 444 1646

\* Designates system demonstration centers.

## ZILOG SALES OFFICES

#### EASTERN REGION

Zilog, Inc.  
76 Treble Cove Road  
No. Billerica, MA 01862  
TEL 617 667 2179  
TWX 710 347 6660

#### MIDATLANTIC REGION

Zilog, Inc.  
P.O. Box 92  
Bergenfield, NJ 07625  
TEL 201 385 9158  
TWX 710 991 9771

#### MIDWESTERN REGION

Zilog, Inc.  
1701 Woodfield Place  
Suite 417  
Schaumburg, IL 60195  
TEL 312 885 8080  
TWX 910 291 1064

#### SOUTHWESTERN REGION

Zilog, Inc.  
17982 Sky Park Circle  
Suite C  
Irvine, CA 92714  
TEL 714 549 289  
TWX 910 595 280

Supplied by

**MICROPOWER LTD**  
The U.K.'s only dedicated  
ZILOG distributor

HAMPSTEAD HOUSE  
BASINGSTOKE HAMPSHIRE  
RG21 1LG  
Tel: Basingstoke (0256) 54121  
Telex: 858572

10460 Bubb Road, Cupertino, California 95014

03-0052-01

**Zilog**

Telephone: (408) 446-4666 TWX: 910-338-7621

Printed in U.S.A.  
Copyright © 1977 by Zilog, Inc.