

Ridge Bootstrap Debugger

(RBUG)

August 30, 1982

The Ridge bootstrap debugger (RBUG) is 4K of assembly language code that is loaded into memory location 3E000H when the system is booted. If the load switch on the clock board is enabled, RBUG then loads and transfers control to the Ridge operating system (ROS). Otherwise RBUG prints a banner and prompt, ":". RBUG may be used to set breakpoints, and display and modify Ridge memory. RBUG runs in kernel mode, uses a 4K data area at location 3D000H, and neither this data area, nor its code are overlaid by ROS. RBUG may be entered at any time by depressing switch 0 (located on the clock board). Following is a description of RBUG's commands.

### Syntax Notation

In the syntax used below "[" and "]" surround optional components. All command names and parameters are separated by at least one blank. Numeric values are all in hex.

### Breakpoint Command

B codeloc	Sets a breakpoint at the absolute code location specified (virtual addresses won't work). When execution reaches the "codeloc", RBUG is entered, and a banner with the current program counter is printed. If the program was executing in user mode "V=xxxx" is displayed, where "xxxx" is the segment number. After a breakpoint is reached, it is cleared. A maximum of sixteen breakpoints may be set. When this is exceeded, "break full" is displayed and all breakpoints are cleared.
-----------	--

### Load ROS from Floppy Disc Command

CI

### Display Memory Command

```
{D      } address count
{DC     }
{DD     }
{DV segno}
```

Displays memory starting at "address" for "count" bytes. Memory may be displayed using one of four addressing modes. "D" uses real memory addresses, "DC", "DD", and "DV" use virtual addresses. "DC" uses the current user code segment as the segment number, "DD" uses the current user data segment number as a segment number, and "DV" uses "segno" as the segment number. If the virtual address specified is absent, nothing is displayed. The current user code and data segments are located in SR8 and SR9, respectively.

### Display Special Registers Command

DSR                   The sixteen special registers are displayed from left to right, R0 - R7, with R8 - R15 on the line below.

### Display Registers Command

DR                    The sixteen registers are displayed from left to right, R0 - R7, with R8 - R15 on the line below.

### Exit Breakpoint Command

E

### Fill Memory Command

F addr length value

Fills memory starting at location "addr" with the hex byte "value", for count "length".

### Load ROS from Hard Disc Command

H [boot-slot]  
[file.id ]

The RBUG code contains a boot table with a list of ROS file id's that can be used as initial operating system code. Typing "H" with no parameter causes RBUG to attempt to load from any file in the boot table. The first file that is successfully located and loaded is executed. Typing "H" followed by an index into the boot table causes RBUG to attempt to boot the corresponding file. "H file.id" causes RBUG to boot using "file.id" as a file id. File id must be typed as two hex numbers separated by a period.

### I/O Read Command

I parm

A READ instruction is executed using "parm" as the I/O address word. The data read is displayed in hex. If an I/O timeout occurs, the word "timeout" is displayed.

### I/O Interrupt Read Command

IR

This instruction causes an ITEST instruction to be executed. ITEST is called repeatedly until an interrupt occurs. When an interrupt occurs, the I/O interrupt read data is displayed, and the command ends.

### Modify Memory Command

M addr

The byte of data at the specified address is displayed in hex. A new value may then be input, followed by a carriage return. After carriage return, the next sequential address is displayed, and can then be modified. Typing carriage return leaves the value unmodified. Modify mode is ended by typing any non-hex character.

### Modify Register Command

M regno                    The selected register is displayed in hex. "regno" is specified in hex. A new value may then be input, followed by carriage return. Typing carriage with no new value leaves the register unmodified.

### I/O Write Command

O parm data                A write instruction is executed using "parm" as the I/O address word and "data" as the I/O data word. If an I/O timeout occurs, "timeout" is displayed.