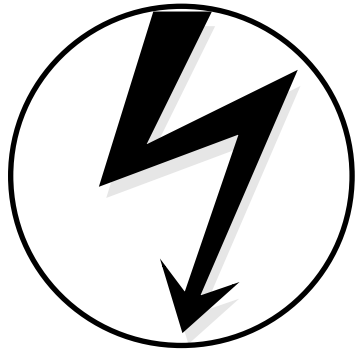
Z8ICE000ZEM
USER'S MANUAL

Electrical  Safeguards

WARNING!

Follow the precautions listed below to avoid permanent damage to the emulator.

I. Always use a grounding strap to prevent damage resulting from electrostatic discharge (ESD).

II. Power-Up Precautions.

1. If the target application board has its own power, remove the JP1 jumper on the Z8M001 board.
2. Ensure that all power to the emulator and the target application (if any) is turned OFF.
3. Connect the target pod to the target application (if any).
4. Power up the emulator, then press the RESET button.
5. Power up the target application (if any).

III Power-Down Precautions.

When powering down, follow this procedure in the precise order shown below:

1. Press the TARGET RESET button.
2. Power down the target application board (if any).
3. Remove the target pod.
4. Power down the emulator.

NOTES:

1. Refer to the “Precaution List” section of the Z8ICE00ZEM Data Sheet, DS00020300-ZX8X0998, for additional operating precautions specific to various devices.
2. Do not leave the emulator powered up with the RS-232C cable connected to a powered-down PC.
3. Before inserting target pod into target application board, refer to Chapter 2, “Set-Up and Installation,” to determine appropriate jumper selections and options.

© 1998 by ZiLOG, Inc. All rights reserved. No part of this document may be copied or reproduced in any form or by any means without the prior written consent of ZiLOG, Inc. The information in this document is subject to change without notice. Devices sold by ZiLOG, Inc. are covered by warranty and patent indemnification provisions appearing in ZiLOG, Inc. Terms and Conditions of Sale only.

ZiLOG, Inc. makes no warranty, express, statutory, implied or by description, regarding the information set forth herein or regarding the freedom of the described devices from intellectual property infringement. ZiLOG, Inc. makes no warranty of merchantability or fitness for any purpose.

The software described herein is provided on an as-is basis and without warranty. ZiLOG accepts no liability for incidental or consequential damages arising from use of the software.

ZiLOG, Inc. shall not be responsible for any errors that may appear in this document. ZiLOG, Inc. makes no commitment to update or keep current the information contained in this document.

ZiLOG's products are not authorized for use as critical components in life support devices or systems unless a specific written agreement pertaining to such intended use is executed between the customer and ZiLOG prior to use. Life support devices or systems are those which are intended for surgical implantation into the body, or which sustains life whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.

ZiLOG, Inc.
910 East Hamilton Ave., Suite 110
Campbell, CA 95008
Telephone: (408) 558-8500
FAX: (408) 558-8300
Internet: http://www.zilog.com

ABOUT THIS MANUAL

We recommend that you read and understand everything in this manual before setting up and using the product. However, we recognize that users have different styles of learning. Therefore, we have designed this manual to be used either as a how-to procedural manual or a reference guide to important data.

The following conventions have been adopted to provide clarity and ease of use:

- Courier Font For Executables

  Commands, variables, icon names, entry field names, selection buttons, code examples, and other executable items are distinguished by the use of the Courier font. Where the use of the font is not possible, like in the Index, the name of the entity is capitalized. For example, a procedure may contain an instruction which appears as: Click on `File`. However, an Index entry would appear as FILE.

- Grouping of Actions Within A Procedure Step

  Actions in a procedure step are all performed on the same window or dialog box. Actions performed on different windows or dialog boxes appear in separate steps.

- Sequencing Words Within A Procedure Step

  When an item in a procedure contains a series of actions, the second action is preceded by the word *then*, and the third and subsequent actions are preceded by the word *and*. For example: Click on `View`, then `Memory`, and `Z8 Code Memory`.

- Unavailable menu items are presented in gray.

ADDITIONAL SOURCES OF INFORMATION

In addition to this manual, you should have access to and be familiar with the following documentation:

- Z8ICE000ZEM Data Sheet, DS0020300-Z8X0998
- Z8PLUS User's Manual, UM97Z8X0300
- The new emulator Graphical User Interface (GUI) software features many enhancements, including an improved context-sensitive on-line help facility that provides brief messages on keyboard, emulator commands, and various procedures on how to use the emulator. Refer to the `README.TXT` file on the GUI diskette for detailed information.

TRADEMARKS

Windows is a registered trademark of the Microsoft Corporation.

Z8 is a registered trademark of the ZiLOG, Inc. ICEBOX is a trademark of ZiLOG, Inc.

ZiLOG

Totally Logical

TABLE OF CONTENTS

Chapter Title and Subsections **Page**

Chapter 3. Summary of Menus, Commands, and Operations

Chapter 4. Troubleshooting Guide

Chapter Title and Subsection **Page**

Totally Logical

LIST OF FIGURES

Figure Title **Page**

Chapter 2. Set-Up and Installation

Chapter 3. Summary of Menus, Commands, and Operations

Figure Title **Page**

Chapter 3. Summary of Menus, Commands, and Operations (Continued)

Totally Logical

CHAPTER 1

INTRODUCTION

OVERVIEW

Congratulations for selecting a fine development tool! The ZiLOG Z8ICE000ZEM Emulator is carefully engineered to provide the best balance between low-cost and useful features to shorten your development time for $Z8^{PLUS}$-based products. The Z8 graphical user interface (GUI) emulator software runs under MS-Windows®, supporting selected ZiLOG devices.

SUPPORTED DEVICES

The Z8ICE000ZEM is designed to support the components of the $Z8^{PLUS}$ family of products: Z8E000 and Z8E001.

The complete, up-to-date list of all supported devices is specified in the ZiLOG Data Sheet (DS) document, which is available from the ZiLOG Bulletin Board Service (ZBBS) and the home page at `www.zilog.com` (see Appendix A).

GUI-SUPPORTED COMPILER, ASSEMBLER FORMATS

The emulator GUI supports object (binary or Intel hex) code files produced by the ZiLOG Macro Cross Assembler (ZMASM), Production Languages Corporation $(PLC)_1$ assembler, 2500AD, and other third-party development tool companies. See the ZiLOG Bulletin Board Service (ZBBS) and the home page at `www.zilog.com` for additional information.

1. Production Languages Corporation (PLC) may be reached by telephone at (817) 367-3699, (800) 525-6289, or online at `info@plcorp.com`.

HARDWARE SPECIFICATIONS

Operating Conditions

Operating Temperature: 20°C ±10°C

Supply Voltage: 9.0 VDC ±10%

Maximum Emulation Speed: 10-MHz External Oscillator (Ships with a 10-MHz Oscillator)

Operating Humidity: 10-90% RH (Noncondensing)

Power Requirements

+9.0 VDC @ 0.5A Maximum

Dimensions

Width: 7.5 in. (19.0 cm)

Length: 6.75 in. (17.0 cm)

Height: 0.9 in. (2.3 cm)

Serial Interface

RS-232C @ 9600, 19200 (Default), 28800, or 57600 Baud Rates

Number of Breakpoints

Maximum: 256

HOST COMPUTER SYSTEM REQUIREMENTS

The following requirements apply when the optional ZiLOG ZMASM/ZDS assembler is implemented. If another assembler is used, refer to its requirements.

Table 1-1. Host Computer System Requirements

	ZMASM (Windows 3.1)		ZMASM (Windows 95) ZDS			
	Min.	**Recommended**	**Min.**	**Recommended**	**Min.**	**Recommended**
Processor	386	486	486	Pentium	486	Pentium
Speed (MHz)	33	66	55	100 (or higher)	66	150 (or higher)
RAM (MB)	4	8	8	16 (or more)	8	32 (or more)
Video Adapter	VGA	SVGA	VGA	SVGA	SVGA	SVGA
Hard Disk (MB) (Free Space)	2.0	4.0	3.0	4.0	10.0	16.0
3.5-Inch, HD Floppy Disk Drive	a	a	a	a	a	a
RS-232C COM Port	a	a	a	a	a	a
Color Monitor		a		a	a	a
Mouse/Pointer Device	a	a	a	a	a	a
Printer		a		a	a	a
Editor		a		a	Included	Included

KIT CONTENTS

The emulator kit contains one of each of the following items:

- Z8M001 Emulation Board (99C0603-001)
- 18-Pin Emulation Pod Cable
- 9-Pin M-F Serial cable, 6 Feet
- Z8 Graphical User Interface (GUI) Software
- ZiLOG Macro Cross Assembler (ZMASM)/ZiLOG Developer Studio (ZDS) Software
- ZiLOG Macro Cross Assembler (ZMASM) License Agreement
- ZiLOG 1998 Technical Library CD-ROM, which contains Z8 device data sheets, user manuals, application notes, and other valuable information
- Z8ICE000ZEM User's Manual

OPTIONAL ITEMS NOT SUPPLIED IN THE KIT

- Laboratory Power Supply with Supply Current of 0.5 Ampere at 9 VDC

  A laboratory-type power supply is recommended when using circuits sensitive to noise, such as analog-to-digital converters.

- SOIC-to-DIP Programming Adaptor (Z86E0700ZDP)

  – 18-Pin SOIC-to-DIP Programming Adaptor

- SSOP-to-DIP Programming Adaptor (Z8E00101ZDH)

  – 20-Pin SSOP-to-DIP Programming Adaptor for Z8M001-based emulators

- Your Z8^{PLUS}-Based Design

  Typically, this is a wire-wrapped or printed circuit prototype that includes a socket for the Z8E00x device into which you can plug the emulation cable from the emulator.

LIMITATIONS

This emulator is designed for emulation purposes only and cannot be used in a stand-alone mode.

RC oscillator emulation is not supported.

Please refer to the `P_and_L.TXT` file or icon for the latest update.

Totally Logical

CHAPTER 2

SET-UP AND INSTALLATION

SOFTWARE INSTALLATION

Software for the emulator is stored on two sets of diskettes:

1. Z8 ICE GUI

2. ZiLOG Developer Studio (ZDS) with ZiLOG Macro Cross Assembler (ZMASM)

NOTE: Refer to the `README.TXT` file on each diskette. Text (`.TXT`) files are easily accessed by using the `Notepad` program.

ZMASM ver 2.10 is the basis for the ZiLOG development environment for use with 16-bit Windows 3.1 and Windows 95.

The ZiLOG Developer Studio (ZDS) is the 32-bit development environment that runs under Windows 95 and Windows NT and includes ZMASM ver. 2.10.

ZMASM/ZDS Installation

If you are installing the ZiLOG Macro Cross Assembler (ZMASM) or ZiLOG Developer Studio (ZDS), run the installation program from the diskette before installing the GUI diskette. You may choose to use another assembler, but the sample session assumes the ZiLOG assembler has been installed.

ZDS is available for the Windows 95 and Windows NT environments. A choice is offered to the user to install either ZDS, that includes ZMASM, or to install ZMASM only.

NOTES:

1. The ZMASM/ZDS is a licensed product; it is not sold. Before opening the envelope containing the software, carefully read the Software License and Limited Warranty Agreement.

2. The installation procedure can be run before creating the installation directory.

To install ZMASM/ZDS, perform the following steps:

1. Select the `Run` command from the `File` menu under the Windows Program Manager (Windows 3.1) or under the `Start` button (Windows 95).
2. Insert the ZMASM (with ZDS) Disk 1 into drive A (or drive B, if appropriate).
3. Type `a:\setup` and press ENTER. (Type `b:\setup` if drive B is used.)

   A dialog box prompts you for the directory to install the software into. The setup program copies the files into the target directory.
4. Follow all on-screen instructions.

   In Windows 3.1, a ZMASM program group icon is placed on the desktop. In Windows 95and NT, a ZiLOG Developer Studio entry is placed on the `Programs` menu under `Start`.
5. Remove the diskette and store in a safe place when installation is complete.

Emulator GUI Installation

To install Z8 ICE GUI under Windows, perform the following steps:

1. Select the `Run` command under `Start`.
2. Insert the Z8 GUI S/W Disk 1 into drive A (or drive B, if appropriate).
3. Type `a:\setup` and press ENTER. (Type `b:\setup` if drive B is used.)

   A dialog box prompts you for the directory to install the software into.
4. Follow all on-screen instructions.

   The setup program copies the files into the target directory, creating a `Z8 ICE GUI` program group icon in the Windows environment.
5. Remove and store all diskettes in a safe place when installation is complete.

Program Uninstallation

Uninstaller facilities are created during the installation of ZMASM/ZDS and the Z8 ICE GUI. An entry is created in the submenu under `Programs`. The `Uninstall` facility should be utilized to properly restore the Windows operating environment.

HARDWARE INSTALLATION

Before installing the hardware, refer to Figure 2-1 provides a diagram for connecting the emulator to your PC and power supply; Figure 2–2 shows the jumper locations on the board.

NOTE: Proper functioning of the emulator depends upon proper installation and running of the GUI software on your PC.

Quick Installation Procedure

This section provides the quick method of installing the hardware utilizing a VDC wall-adaptor power supply.

1. If you are using an optional laboratory power supply (not included), turn the power supply ON (before connecting to the emulator), adjust it to 9.0V, and set it to at least 0.5A; or, plug the supplied wall transformer into a 120 VAC, 15 amp outlet.
2. Connect the RS-232C serial cable between the emulator and the PC.
3. If you are doing in-circuit emulation, connect the emulator to your design (if it is ready to test).
4. Set up the oscillator and option jumpers.
5. Refer to the "Electrical Safeguards" page of this manual.
6. Connect the power supply or wall transformer to the emulator.

Serial Cable Connection

Locate the serial cable. Connect the male end to the female connector, and the female end to the COM1, COM2, COM3, or COM4 connector of your PC.

Complete Hardware Installation

Connecting to a Power Supply

If your power supply allows voltage adjustment, do the following:

1. Turn the power supply on and adjust it to +9V.
2. Set the power supply for at least 0.5A, if there is a current-limiting adjustment.
3. Turn the supply OFF or ensure that a nonadjustable supply is OFF.

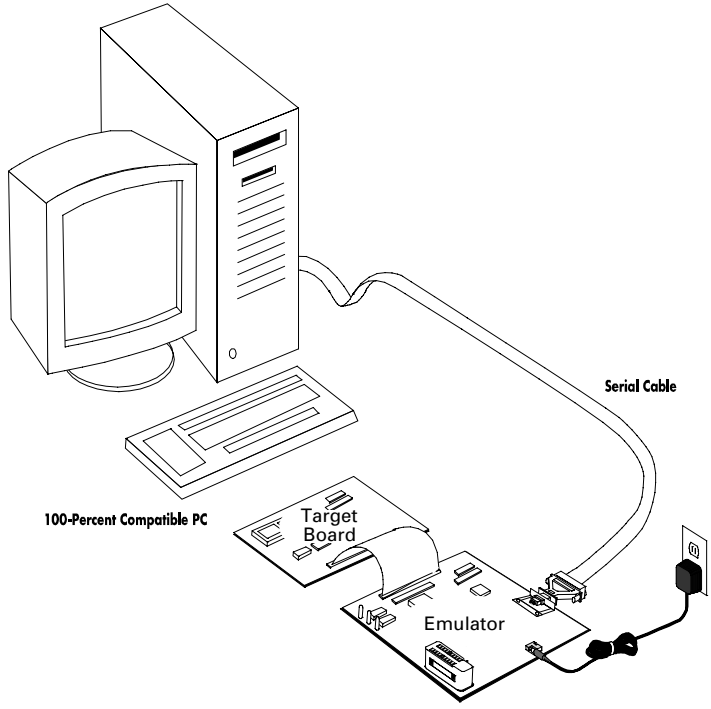


Figure 2-1. Hook-Up Diagram

4. Locate the power connectors on the emulator. Connect the connector labeled GND on the emulator to your power supply connector that's labeled COM, GND, or with a ground symbol. Connect the connector labeled 9 VDC on the emulator to your power supply connector that's labeled "+" or "+V".

Connecting the Serial Cable to the PC

5. Connect the male end of an RS-232C serial cable to the female connector on the emulator, and the female end of the serial cable to the COM1, COM2, COM3, or COM4 connector of your PC.

NOTE: If connector availability is limited to a 25-pin COM1 through COM4, you'll have to use either a different cable or a 25-pin to 9-pin converter. (ZiLOG does not provide either of these items.)

Connecting to Your Design

6. Locate the emulation cable for your $Z8^{PLUS}$-based design.

NOTE: Wear a properly grounded wrist strap or similar ESD protection before proceeding with this step.

7. Plug one end of the emulator cable into the $Z8^{PLUS}$ socket of your target hardware, being very careful to orient the pin 1 marking (as indicated by the red mark on the ribbon cable) to match the pin 1 on your

target board. Plug the other end of the cable into the matching black emulation cable connector in front of the emulator (see Figure 2-2).

NOTE: The inductance and capacitance of the emulation cable may affect the signals to and from the emulator and target board, especially if the target board has low-current drivers, pull-ups, and pull-downs.

Setting Up the Z8ICE000ZEM Oscillator and Option Jumpers

The oscillator and option jumpers are accessible on the emulator. Refer to Figure 2-2 for emulator jumper locations. Refer to Table 2-1 and Table 2-2 when setting up the jumpers. Also refer to Appendix C for the complete Z8ICE000ZEM Emulator Schematic Diagram.

The option jumpers allow you to configure things like whether the emulator provides power to your design on the V_{CC} pin of the emulated $Z8^{PLUS}$ device, and how the clock pin is connected.

The clock for the emulated device can be provided from an oscillator in the emulator or from a TTL-level clock on your design. For electrical reasons, the clock cannot be supplied from the oscillator on your design until you are done using the ICEBOX and have either programmed an EPROM or OTP $Z8^{PLUS}$ and installed it in your design.

WARNING!

If your design already has a power supply, do not power your design from the emulator V_{CC} pin on location JP1 of the Z8M001 board.

Power Up

If anything unusual (such as unexpected sounds and smells) occurs the first time you turn on the power supply, quickly turn off the power supply and check your connections. If your power supply allows voltage adjustment, ensure that adjustment is within specified range (see "Hardware Specifications" section). If your power supply has a current meter, ensure that the emulator is drawing within the rated current.

After emulator power-up, press the MASTER RESET button to reset the emulator. (Pressing the TARGET RESET button resets only the Z8M001 ICE chip.) If the emulator is not powering your design through the V_{CC} pin, turn on the power supply of the design.

Power Down

When powering down, follow the procedure described below:

1. Press the TARGET RESET button on the emulator.
2. Power down the target application board (if using its own power supply).
3. Remove the target pod.
4. Power down the emulator.

Refer to the "Electrical Safeguards" page of this manual.

Jumper Locations

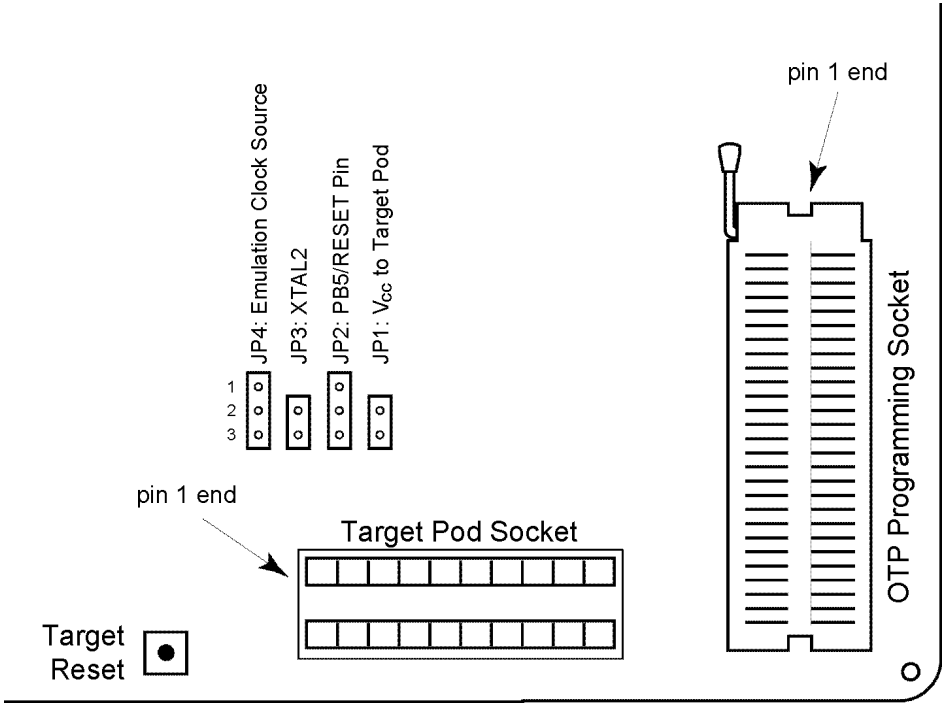


Figure 2-2. Option Jumper Locations

NOTE: Refer to the following tables for information on how to set up these jumpers for your specific emulator model before proceeding to the next installation steps.

Table 2-1. Jumper Option Settings

Jumper	Position	Description
JP1	Open* Close	Emulator power is isolated from target board. Emulator supplies power to target board via V_{CC} pin of $Z8^{PLUS}$.
JP2	1-2 2-3*	Reserved Connects the $\overline{RESET}$ pin of the Z8M001 ICE chip to pin 5 of the emulated $Z8^{PLUS}$.
Note: The * designates the default setting when shipped.		

Table 2-1. Jumper Option Settings (Continued)

Jumper	Position	Description
JP3	Open* Close	Oscillator of the emulator ICE chip is not connected to the target pod pin. Clock of the $Z8^{PLUS}$ emulator is connected to the XTAL2 pin of the $Z8^{PLUS}$ socket.
JP4	1-2 2-3*	XTAL1 pin of the $Z8^{PLUS}$ emulator is connected to the XTAL1 pin of the $Z8^{PLUS}$ socket. The target board runs the emulator from either (but not both) the emulator or the target board. XTAL1 of the emulator ICE chip is not connected to the target pod pin. The emulator crystal is driving only the ICE chip.
Note: The * designates the default setting when shipped.		

Table 2-2. Device Jumper Selection

Jumper	Z8E000/001	Reserved
JP2	2-3*	1-2

Figure 2-3 shows how Jumpers JP3 and JP4 select the clock source for emulation.

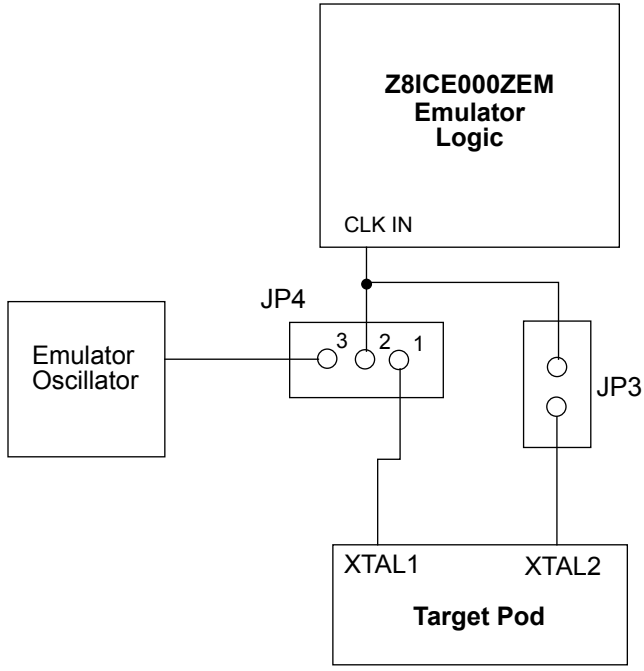


Figure 2-3. Jumper Options Diagram

Totally Logical

CHAPTER 3
SUMMARY OF MENUS, COMMANDS, AND OPERATIONS

MAIN MENU

File	View	OTP!	Configuration	Help

The Main Menu window of the emulator graphical user interface (GUI) is displayed after the GUI program is started. The following menu items can be accessed from this Main Menu:

- File
- View
- OTP!
- Configuration
- Help

Each of these items, along with all windows, subset menus, menu items, commands, and operations, are summarized in the following sections of this chapter.

NOTE: Consult *ZiLOG*'s ZBBS or Internet site at `www.zilog.com` to obtain the latest released version of Z8 GUI Emulator software.

FILE MENU

File	View	OTP!	Configuration	Help
Open Session...				
Save Session...				
Download Application...				
Download To Z8 Code Memory...				
Upload Z8 Code Memory...				
Exit Alt + F4				

Open Session

The `Open Session` menu item provides a facility for loading a previously saved session.

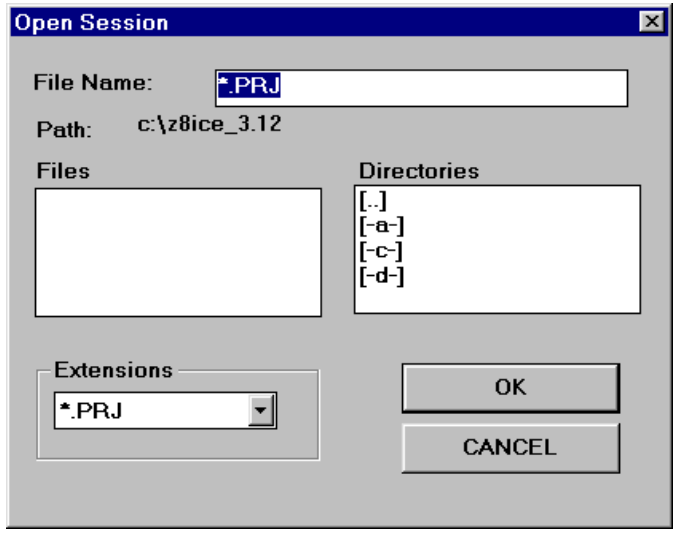


Figure 3-1. Open Session Dialog Box

Save Session

Selecting `Save session` saves the information about the position of the opened windows, downloaded code file, font size used, and debug flags such as `Trace` and `Animation`. Not all windows reappear upon reloading of a saved session.

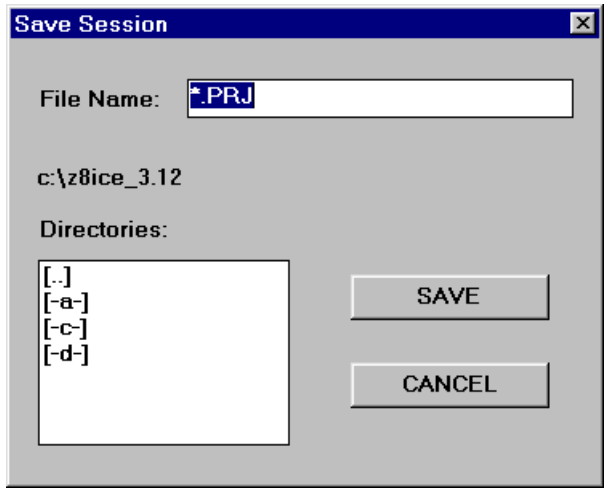


Figure 3-2. Save Session Dialog Box

Download Application

The Download Application menu item enables you to download an application object file to the Code Memory of the ICEBOX. Type the path and name or use the `Browse` button to locate and select your file. Two object formats are supported for output from ZMASM or the PLC compiler. In addition, memory can be cleared prior to downloading.

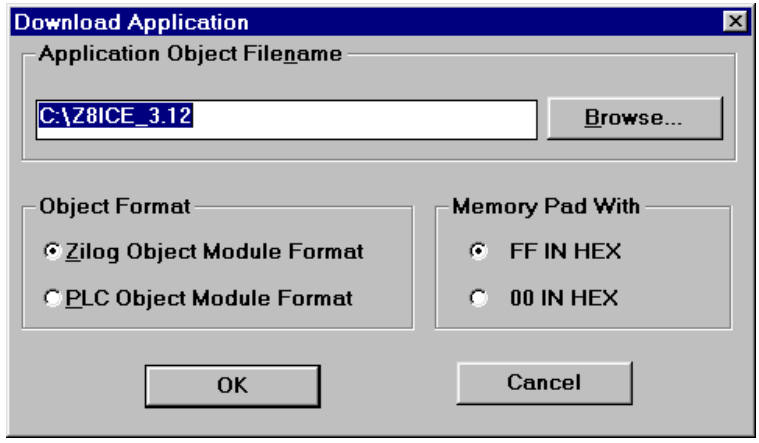


Figure 3-3. Download Application Dialog Box

Download To Z8 Code Memory

The Download To Z8 Code Memory menu item enables you to download Intel hexadecimal (hex) or binary format code to Code Memory. The Address Format section of the window designates the format of the contents of the file being downloaded. Memory can also be padded with FFFFh or 0000h.

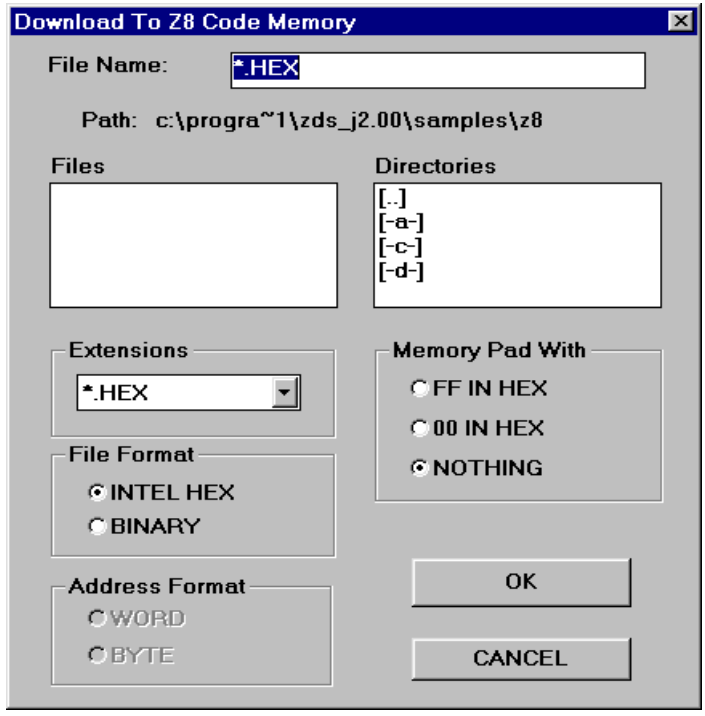


Figure 3-4. Download To Z8 Code Memory Dialog Box

Upload Z8 Code Memory

The `Upload Z8 Code Memory` menu item enables the contents of code memory to be saved in an object file. Three formats are available for storing the data: binary, Intel hex, and disassembly. The entire contents of code memory are stored when the `Save All` button is activated. A portion of the contents is stored when numbers are entered in the `Start Address` and `End Address` fields, and the `Save` button is activated.

This function may be used for sessions involving patched code.

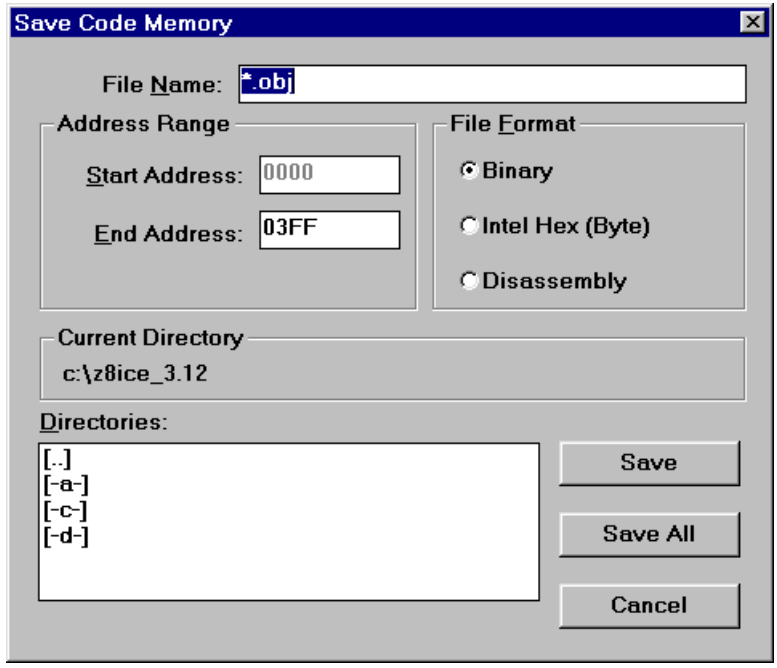


Figure 3-5. Save Code Memory Dialog Box

Exit

The Exit menu item enables you to leave the emulator GUI. The ZiLOG ICEBOX confirmation window offers the message, "Save the Current Session?", and the options `Yes`, `No`, and `Cancel`.

VIEW MENU

File	View	OTP!	Configuration	Help
	Registers			
	Memory			
	Debug			
	Source			
	Output			

Registers

The Registers menu item initiates the display of the following submenu:

File	View	OTP!	Configuration	Help
	Registers	Z8 Registers		
	Memory	Z8 Counter/Timers		
	Debug	Z8 Ports		
	Source	Z8 Status		
	Output			

Z8 Registers

The `Z8 Registers` menu item initiates the display of the Z8 Register dialog box. That window contains the contents of all the general purpose registers.

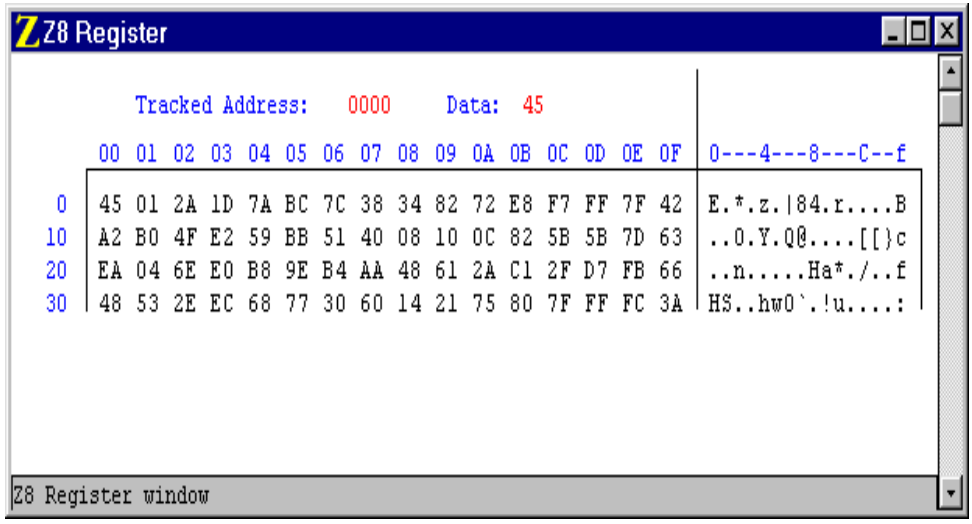


Figure 3-6. Z8 Register Window

When the Z8 Register window is displayed, the following menu bar is displayed:

File	View	OTP!	Tracking	Font Size	Window	Help

Tracking

Selecting `Tracking` from the menu, displays the `Absolute Address`, `Indirect (@Rg)`, and `Register Pointer` menu items.:

File	View	OTP!	Tracking	Font Size	Window	Help
			Absolute Address			
			Indirect (@Rg)			
			Register Pointer			

Absolute Address

You can track to a register by its absolute address by entering the absolute address value in the dialog box that appears when Absolute Address is selected from the `Tracking` menu.

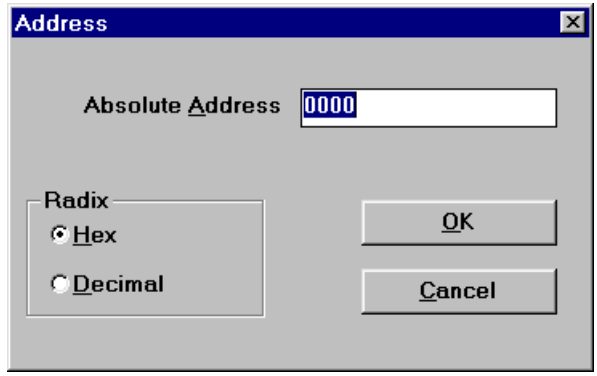


Figure 3-7. Address Dialog Box - Absolute Address

Indirect (@Rg)

You can indirectly track to a register (contents of which are the address of another register) by entering the value in the Indirect (@Rg) dialog box.

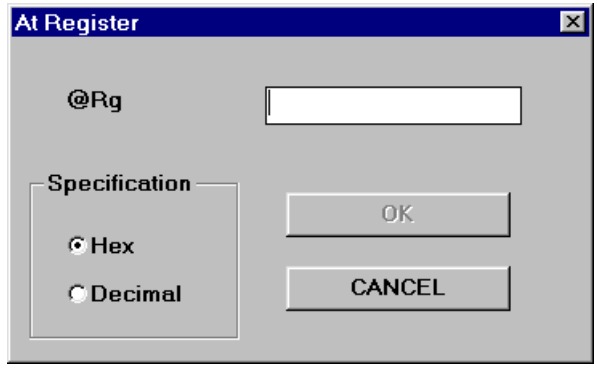


Figure 3-8. At Register Dialog Box

Register Pointer

You can retrieve the current value of the register pointer.

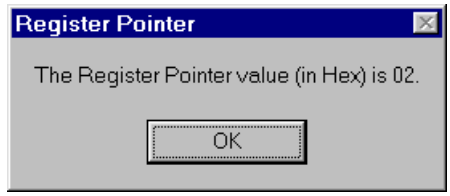


Figure 3-9. Register Pointer Window

When you right-click on your two-button mouse, the following items are displayed:

- Tracking Absolute Address...
- Tracking Indirect (@Rg)...
- Tracking Register Pointer...

Font Size

You can choose from a number of font sizes: Point 6 to Point 12.

File	View	OTP!	Tracking	FontSize	Window	Help
				Point 6		
				Point 8		
				Point 10		
				Point 12		

Window

File	View	OTP!	FontSize	Window	Help
				Cascade Shift+F5	
				Tile Shift+F4	
				Arrange Icon	
				Close All	

The `Cascade`, `Tile`, and `Arrange Icon` items perform standard Windows functions.

`Close All` displays the ZiLOG ICEBOX confirmation window which offers the message, "Save the Current Session?", and the options `Yes`, `No`, and `Cancel`.

Z8 Code Memory

The Z8 Code Memory window provides a facility for tracking and changing the contents of the code memory registers.

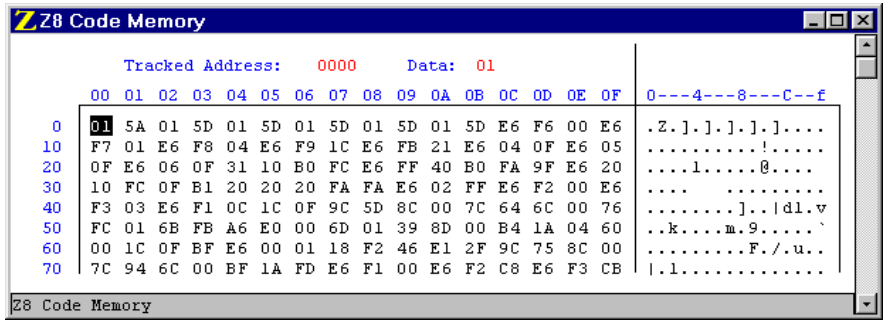


Figure 3-10. Z8 Code Memory Window

When the Z8 Code Memory dialog box is displayed, the following menu bar is displayed:

File	View	OTP!	Edit	Tracking	Font Size	Window	Help

Tracking

Selecting `Tracking` from the menu displays the following menu items:

File	View	OTP!	Edit	Tracking	Font Size	Window	Help

- **Absolute Address**
- **Indirect (@RR)**
- **Index Address**
- **Program Counter**
- **Stack Pointer**

Absolute Address

You can track to a register by its absolute address by entering the absolute address value in the dialog box that appears when Absolute Address is selected from the `Tracking` menu (see Figure 3-7).

Indirect (@RR)

You can indirectly track to a register pair by entering the value in the At Register dialog box.

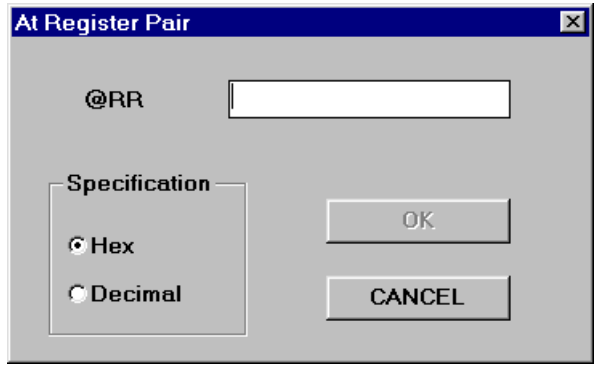


Figure 3-11. At Register Pair Dialog Box

Index Address

You can indirectly track to a value stored in an index register as a displacement from a base address.

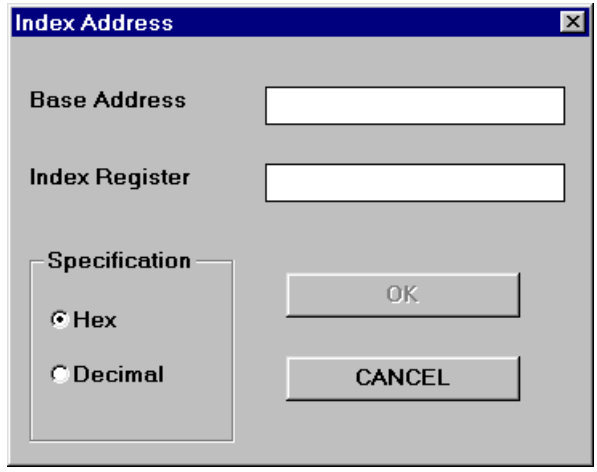


Figure 3-12. Index Address Dialog Box

Program Counter

The Program Counter message window reports the current value of the program counter.

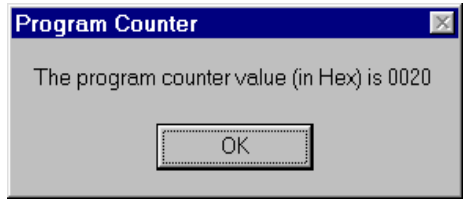


Figure 3-13. Program Counter Window

Stack Pointer

The Stack Pointer message window reports the current value of the stack pointer.

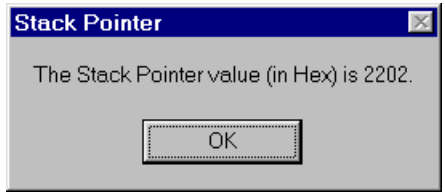


Figure 3-14. Stack Pointer Window

Edit

File	View	OTP!	Edit	Tracking	Font Size	Window	Help

- Fill...
- Clear All
- Start Address...

Fill

The `Fill` menu item enables you to assign a value to a specified range of memory addresses. The Memory Fill dialog box has entry fields for `Start Address`, `End Address`, and `Fill Value`. The `Radix` field specifies the format, hexadecimal or decimal, of the entry in the `Fill Value` field.

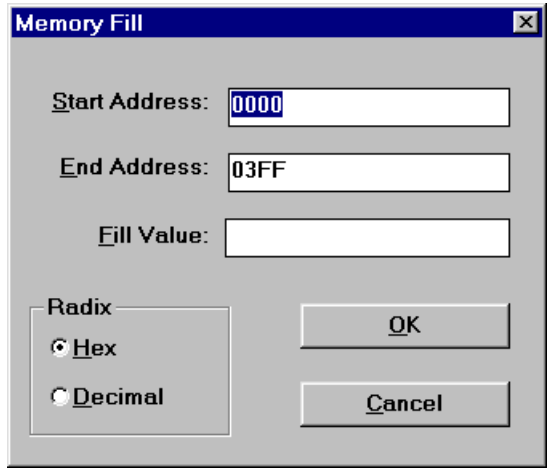


Figure 3-15. Memory Fill Dialog Box

Clear All

The `Clear All` menu item provides the option of filling Z8 Code Memory with the `0000h` or `FFFFh` value. To exit the choice list without any changes, press the Escape key.

Start Address

The `Start Address` menu item provides a facility for jumping the cursor in the Z8 Code Memory window to the location entered in the `Start Address` field. The `Radix` field specifies the format, hexadecimal or decimal, of the entry in the `Start Address` field.

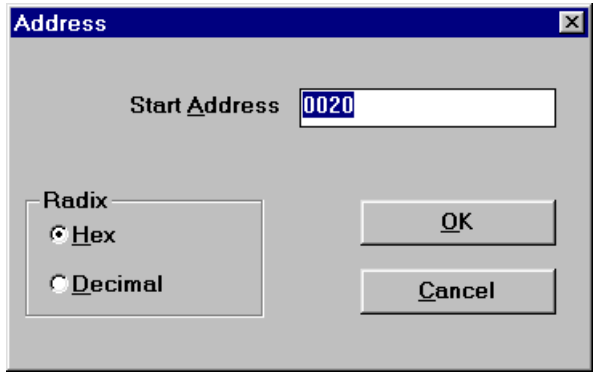


Figure 3-16. Address Dialog Box - Start Address

When you right-click on your two-button mouse, the following items are displayed:

- Clear All...
- Fill...
- Download Application...
- Download Z8 Code Memory...
- Upload Z8 Code Memory...

Debug

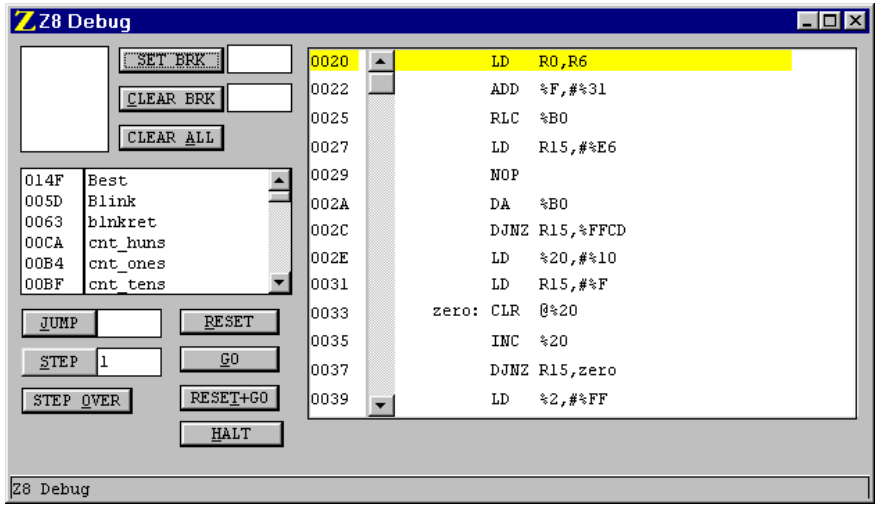


Figure 3-17. Z8 Debug Dialog Box

The Z8 Debug dialog box enables you to view the disassembled code, set break points, trace through the code, and perform other debug operations.

Display and Input Fields

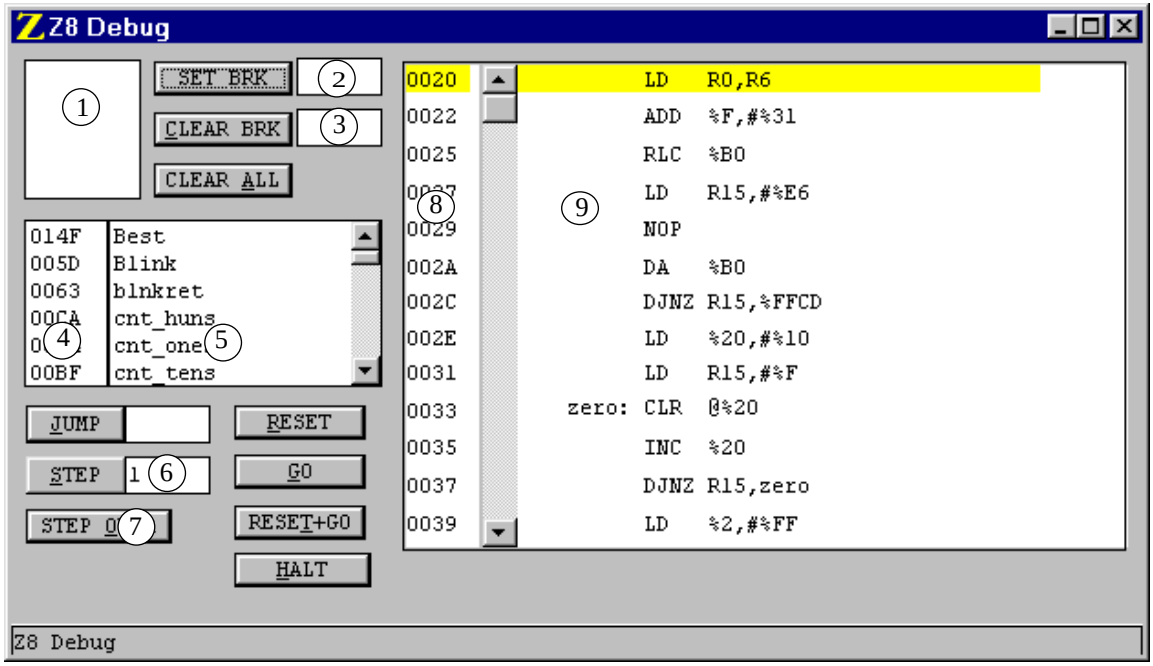


Figure 3-18. Z8 Debug Fields

The Z8 Debug dialog box contains the several fields which are numbered for reference in the preceding figure:

Table 3-1. Debug Display and Input Fields

Number	Name	Description
1	Breakpoint List	This display area contains a list of all addresses currently defined as breakpoints.
2	Set Breakpoint Input	This field is an input area for setting a breakpoint. To set a new breakpoint, type the address in this field or click on a line of code in the `Code List` field (field 9), then click on the `SET BRK` button. After the button is activated, the address continues to be displayed in the field. The address then appears in the `Breakpoint List` (field 1).

Table 3-1. Debug Display and Input Fields (Continued)

Number	Name	Description
3	Clear Breakpoint Input	This field is an input area for deleting a breakpoint. To delete a breakpoint, type the address in this field or place the cursor on the address, then click on the CLEAR BRK button. After the button is activated, the address continues to be displayed in the field. The address is removed from the Breakpoint List (field 1).
4	Symbol Address	This display area contains the address of the symbol to the right of the divider line.
5	Symbol Table	This display area contains the list of currently defined symbols. The symbol table is loaded only when both of the following conditions are fulfilled: 1) A symbol file (`.SYM`) is loaded in code memory, and 2) The Z8 Debug window is selected.
6	Jump	This field is an input area for setting the program counter to the address of a specific line of source code. To jump to an address, type the address in the field, then click on the JUMP button. The address continues to be displayed in the field. The address and line of source code are displayed in the `Trace Buffer` (field 8) and `Code List` (field 9) fields.
7	Step Editing	This field is an input area for the number of steps to advance the program counter. The default value is `1`. The maximum value is `99999`. To single-step through your code, set the value of this field to `1`, then click on the `STEP` button, or click on the `GO` button.
8	Trace Buffer	This display area contains the address of the line of code displayed on the same line in the Code List field (field 9). Use the vertical scroll bar to scroll up and down.
9	Code List	The `Code List` contains the disassembled contents of code memory. The field scrolls with the `Trace Buffer` field. Use the vertical scroll bar to scroll up and down. Selection of a line of code causes the address of that line to be inserted into the `SET BRK`, `CLEAR BRK`, and `JUMP` fields. Code Memory can be change by entering the assembly code directly into the window line by line. After you input a line of assembly code, you must press the RETURN key. The assembled code is then written to Code Memory and immediately reflected in this field. NOTE: Be careful when replacing instructions to avoid corruption of the remaining assembly code.

Buttons

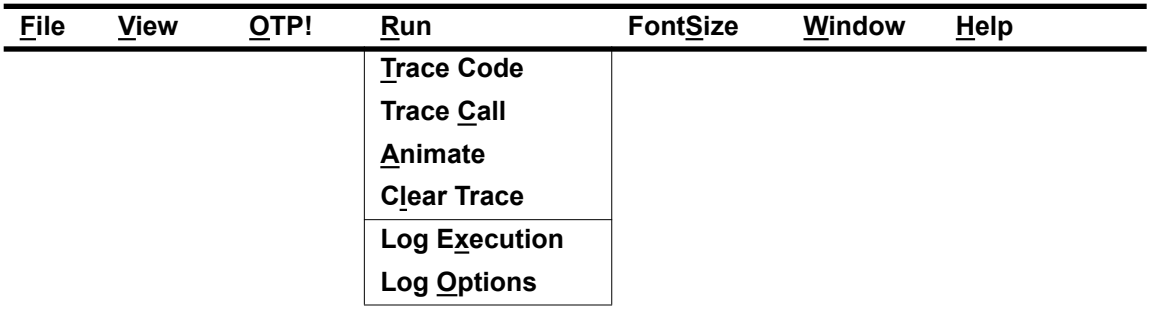
Several buttons are available in the Z8 Debug window.

Table 3-2. Debug Buttons

Name	Description
CLEAR ALL	Activation of this button clears all breakpoints. All addresses are then removed from the `Breakpoint List` (field 1).
CLEAR BRK	Activation of this button deletes a breakpoint at the address in the `Clear Breakpoint Input` field (field 3). The address is then removed from the `Breakpoint List` (field 1).
GO	Activation of this button starts execution of the $Z8^{PLUS}$ program. Execution stops when it hits a breakpoint, a `STOP` or `HALT` instruction, or when you click on the `HALT` button. After a `GO`, the user should not attempt any further operation except activating the `HALT` button to terminate execution.
HALT	Activation of this button terminates execution. If a `GO` command has been issued, the `HALT` button must be activated for the `RESET` command to take effect.
JUMP	Activation of this button sets the program counter to the address in the `Jump` field (field 6). The corresponding line of code is highlighted in the `Trace Buffer` (field 8) and `Code List` (field 9) fields.
RESET	Activation of this button causes a jump to the address located at the reset vector location in the code. If a `GO` command has been issued, the `HALT` button must be activated for the `RESET` command to take effect.
SET BRK	Activation of this button sets a breakpoint at the address in the `Set Breakpoint Input` field (field 2). The address is then added to the `Breakpoint List` (field 1).
RESET+GO	Activation of this button resets the microcontroller and sets the program counter to the beginning of the program. Execution stops at the first breakpoint setting or at activation of the HALT button.
STEP	Activation of this button advances the program counter the number of steps in the `Step Editing` field (field 7). When the value is set to `1`, the `STEP` button single-steps through the code without causing a `RESET`. NOTE: Never process a STEP command through the `HALT` or `STOP` instruction in your code; communication closes at the `HALT` or `STOP`.
STEP OVER	Activation of this button advances the program counter a single step. That is, one assembly instruction is executed. When the current instruction is a `CALL` instruction, subsequent code is executed in real time until the program returns from the subroutine.

When the Z8 Debug dialog box is displayed, the following menu is displayed:

Run



Trace Code

The Trace Code menu item provides a line-by-line trace capability for tracing all instructions while running the `Debug` program. When `Trace Code` is selected, `Animate` is also automatically selected and execution is running.

Trace Call

The Trace Call menu item provides the capability to trace only subroutine calls while running the `Debug` program. Traced code is displayed in the `Code List` field (field 9) of the Z8 Debug window. When Trace Call is selected, `Animate` is also automatically selected and execution is simulated.

Animate

Animation is a mode where the user can simulate single-step execution through the code. A `GO` command starts execution. The address and instruction of the current and last 99 executed lines of code are displayed in the `Trace Buffer` field (field 8) and the `Trace Buffer` field (field 9) of the Z8 Debug window.

NOTE: When Animation is activated, execution is not realtime. Single step execution is emulated. To execute code in real time, the `Animation` feature must be disabled.

Clear Trace

Clears the contents of the `Trace` field and ends tracing activities.

Log Execution

Enables or disables logging. When enabled, selected registers are logged to the specified file each time emulation is halted, such as at breakpoint or after single-step. See Chapter 4, "Sample Session", for an example of trace logging.

Log Options

Specifies the registers to log, the format in which to log them, and the name of the log file. Logging may be linear or circular. Linear logging treats the log file as a linear buffer with infinite capacity; circular logging treats the log file as a circular buffer with finite capacity. The circular buffer default capacity is 100 logging actions; however, the number is user-definable in the `Maximum` field.

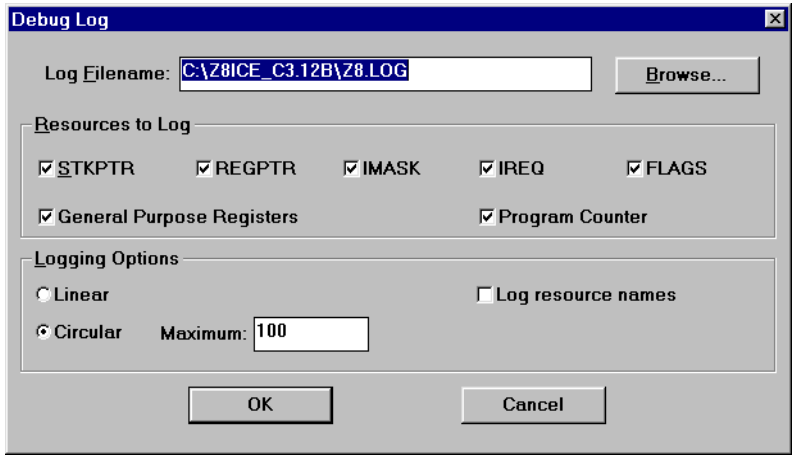


Figure 3-19. Debug Log Window

Resources to Log

The following items are available for logging:

- STKPTR - Stack Pointer
- REGPTR - Register Pointer
- IMASK - Interrupt Mask Register
- IREQ - Interrupt Request Register
- FLAGS - FCH (Flags) Register
- General Purpose Registers
- Program Counter

For detailed information about these resources and registers, refer to the Z8^{PLUS} User's Manual (UM97Z8X0300).

NOTE: If the application inadvertently enters Stop mode, the only way to halt execution through software is by doing a Stop-Mode Recovery (as defined by the user program). The application may also be reset using the target application board's RESET button.

Source

The Source dialog box displays the source code when you download a load (`.LD`) format file using the Download Application dialog box.

Figure 3-20. Source Window

When the Source window is displayed, the following menu bar is displayed:

File	View	OTP!	Options	Window	Help

Options

When the `Options` item is selected, the following item is displayed:

File	View	OTP!	Options	Window	Help
			Source Module...		

Source Module

The Source Module window displays the name of the source assembler module.

Output

The Output window displays the status at file load and at command execution, such as `Halt` and `Reset`.

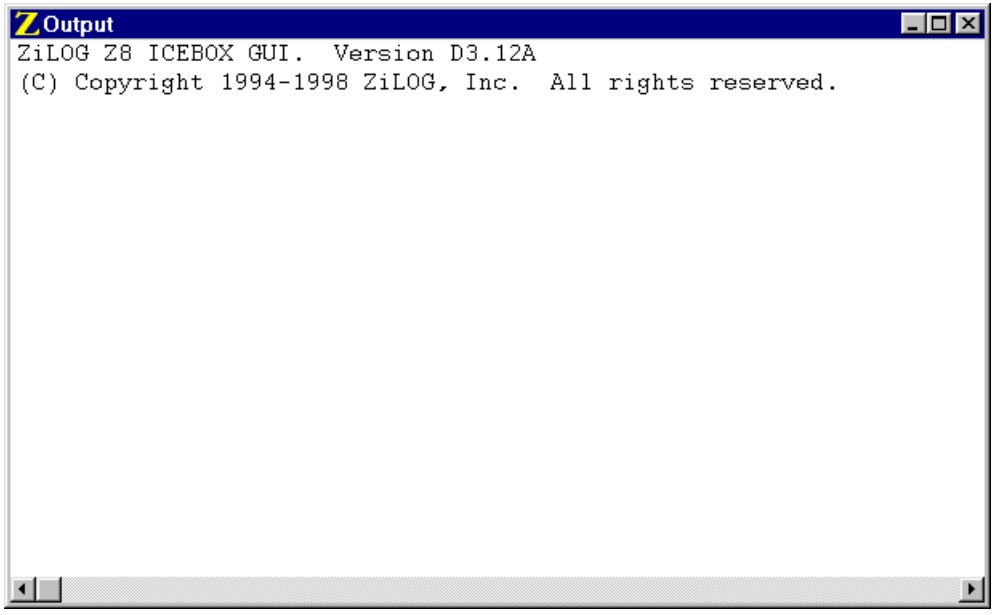


Figure 3-21. Output Window

OTP

This emulator provides OTP programming capabilities. Selecting the `OTP!` menu item displays the OTP window.

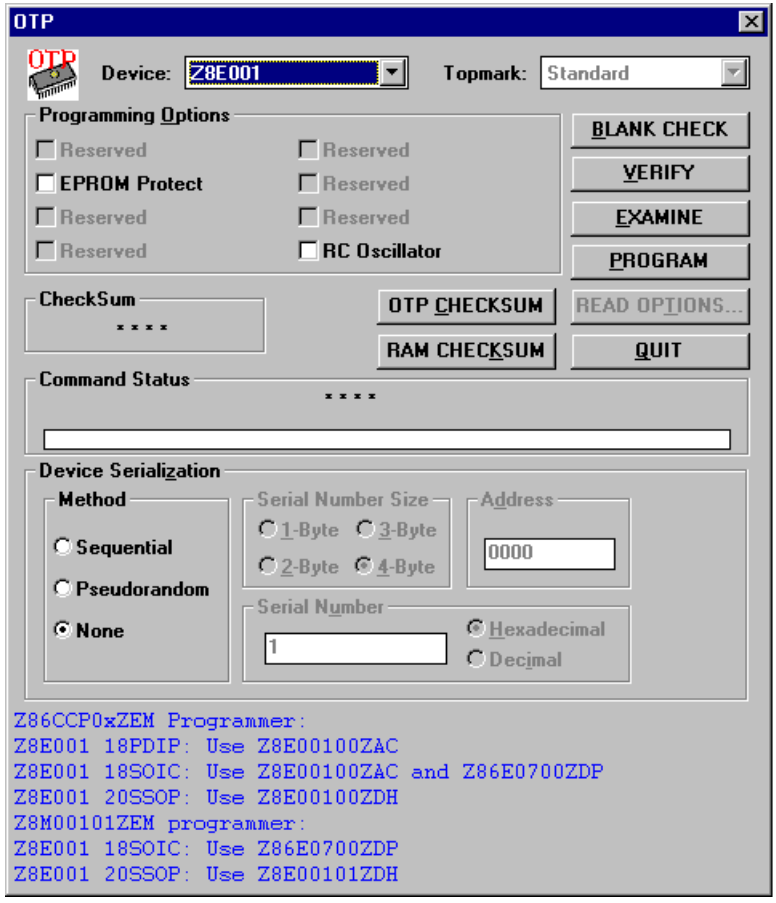


Figure 3-22. OTP Window

Programming Options

Several OTP operations can be performed:

Table 3-1. OTP Operations Functions

Operations	Functions Performed
BLANK CHECK	Checks whether the OTP device is blank.
VERIFY	Compares the contents of the Code Memory and the OTP device.
EXAMINE	Load the content of the OTP device to code memory
PROGRAM	Writes the contents of the Code Memory to the OTP device.
OTP CHECKSUM	Calculates the Checksum of the OTP device only and displays it on the screen.
RAM CHECKSUM	Calculates the Checksum of the RAM and displays it on the screen.
EPROM Protect	Disables the EPROM mode to prevent code retrieval. Also inhibits Factory Test mode.
RC Oscillator	Sets the device to operate with an RC clock source.
QUIT	Quit OTP operation.

NOTES:

1. To obtain a Data I/O (such as UniSite, 3900, or 2900) calculated Checksum, the user should first download the file with `00` padding option, then perform the OTP programming. Data I/O fills its memory with `00` after power up.

2. Checksum differences may occur. During user-program download, the memory is padded with code outside the user program memory if the user program is less than the emulation memory. The memory is padded with either `00`, `FF`, or nothing according to user selection. This does not effect the user program; however, it does effect the calculation of the Checksum during the OTP programming because the Checksum is calculated through out the whole memory size of the OTP.

Command Status

This field contains a message display area and a progress bar that is activated when one of the OTP operations are performed.

Device Serialization

A user-specified serial number can be programmed into the OTP device. The serial number can be changed at any time before it is programmed into a device. The characteristics of the serial number are specified in the Device Serialization section of the OTP window.

Method

Three methods of creating a serial number are available: sequential, pseudorandom, or none.

- Sequential Mode - The serial number increments by one after each device is programmed.
- Pseudorandom Mode - The serial number is generated according to the following algorithm:

```
r[i+1] = (a * r[i] + b) mod 2^N

    where r[i+1] is the new serial number

r[i] is the last serial number

a = 1909

b = 221 571

N = number of bits in the serial
    number
```

Example: Entering `1000` in the Address box, selecting `4-Bytes` as the Serial Number Size, and entering `12345678` in the Serial Number box outputs to the screen as follows:

```
%1000 = %12

%1001 = %34

%1002 = %56

%1003 = %78
```

- None

Serial Number Size

The Serial Number Size field specifies length in number of bytes the serial number occupies in memory. This field is not available when Device Serialization is set to `None`.

Serial Number

Enter the exact serial number in the input field. Specify the format of the number as either hexadecimal or decimal. If a decimal number is input, the number is converted to hexadecimal for storage in memory. This field is not available when Device Serialization is set to `None`.

Address

Enter a four-digit hexadecimal number to be the address where the serial number is to be stored in memory. This field is not available when Device Serialization is set to `None`.

CONFIGURATION

At start-up, the user selects the appropriate ICEBOX entry. When a configuration dialog box is shown on the screen, you can choose the microcontroller and the ROM size you want to emulate. The code file you download to Code Memory is limited to the ROM size selected. Selected ports automatically reference port availability in the selected microcontroller. Other information about the microcontroller, such as the number of ports and extended register banks, can be observed but cannot be altered. Click the `OK` or `CANCEL` button to confirm or cancel the configuration.

NOTE: To modify the configuration, you must first close all open windows; the `Configuration` command then reappears in the Main Menu.

When the `Configuration` menu item is selected, the following submenu is displayed:

File	View	OTP!	Configuration	Help
			Device...	
			Upgrade System Firmware	

Device

When Device is selected, the Configuration dialog box is displayed.

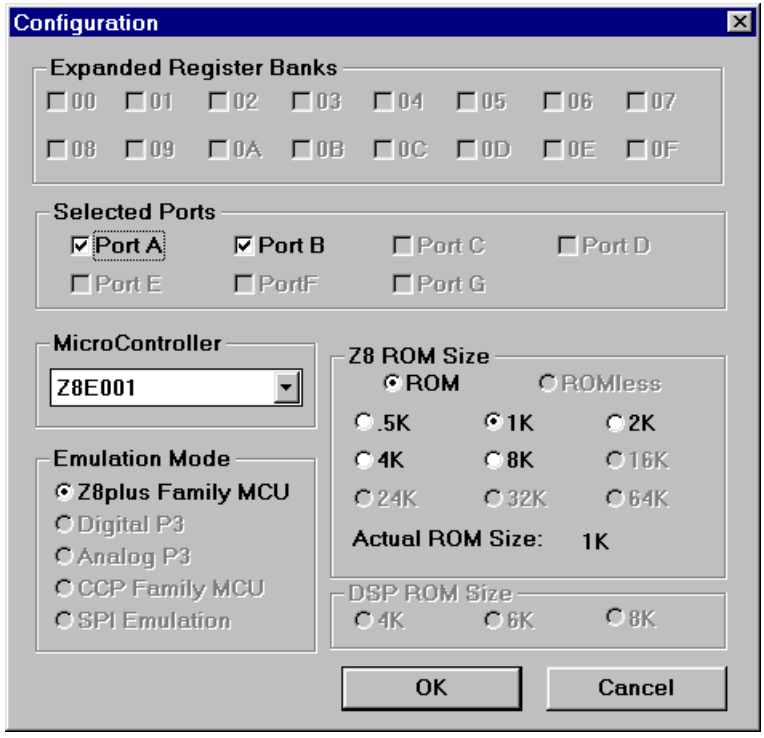


Figure 3-23. Configuration Dialog Box

Upgrade System Firmware

The `Upgrade System Firmware` menu item is not available for the Z8ICE00ZEM emulator.

HELP

File	View	OTP!	Configuration	Help

Help menu:
- Index
- Keyboard
- Commands
- Glossary
- Procedure
- Using Help
- About ICEBOX

The on-line `Help` program is available to provide brief help messages on various topics. The Help program features an index that easily references the emulator commands and procedures. Topics covered under `Help` include:

- Index of the Help Program
- Keyboard Commands
- ICEBOX GUI Commands
- Procedures for Using the Emulator
- How to Use Help
- Version and Copyright Information

WINDOW REFRESHING

When you make a change in the Code Memory window, the code view in the Debug window is updated automatically. When a program runs from the Debug window, the Status window is updated automatically at each step, `Halt`, or breakpoint. All other windows (Registers and Code Memory) are not updated automatically as the program runs. To refresh the window display to reflect the current hardware values, double click the caption bar of each window.

EMULATOR OPERATION

Hardware Reset

When you press the emulator RESET button (or power-down, then power-up), the initial screen and the ZiLOG logo are shown while the ICEBOX goes through the initialization sequence.

Pressing the RESET button resets most of the settings that you establish using the GUI. For example, the emulated ROM size is reset to 4 KB, and all 32-KB breakpoint RAM is cleared.

NOTE: After RESET, you must wait a minimum of 5 seconds (for completion of selftests) before starting the GUI software.

If you press the Hardware Reset while the GUI is running, a communication error dialog box appears with the following message: `Out of synchronization with the emulator`. This message appears when the power or RS-232C cable is removed, or whenever the ICEBOX emulator and the host PC have failed to properly communicate with each other. You have three responses to this message: `Abort`, `Retry`, or `Ignore`.

`Abort` Quits the GUI.

`Retry` Resets the GUI by reinitializing. You are returned to the initial purple screen with the *ZiLOG* logo.

`Ignore` Attempts to reestablish communication between the GUI and ICEBOX emulator without reinitializing.

ROM Size

The ICEBOX has the selectable ROM size of .5K to 8K.

Breakpoint Implementation

The emulator bases its breakpoint facility completely on addresses, rather than on inserting special TRAP instructions into the program, which enable the setting of breakpoints in Program Memory on your target board. The ICEBOX uses a 32Kx1 static RAM for each cycle emitted by the Z8 ICE chip during code execution to implement the breakpoint function in hardware.

EPROM/OTP Programmable Devices

Z8E000 and Z8E001

NOTE: Refer to the Data Sheet (DS) for exact OTP Programming requirements.

RUNNING MULTIPLE EMULATORS

The Z8 ICE GUI allows several emulators to run at the same time, if your PC has more than one communication port.

NOTE: Running multiple emulations (two or more emulators running simultaneously) requires more memory than single-emulation operation; therefore, we recommend using a PC with 8-MB RAM or 16-MB RAM for Windows 95. In addition, two or more communication ports are needed when running multiple instances. Ensure that there are no COM port interrupt conflicts if installing additional COM ports via an add-on board.

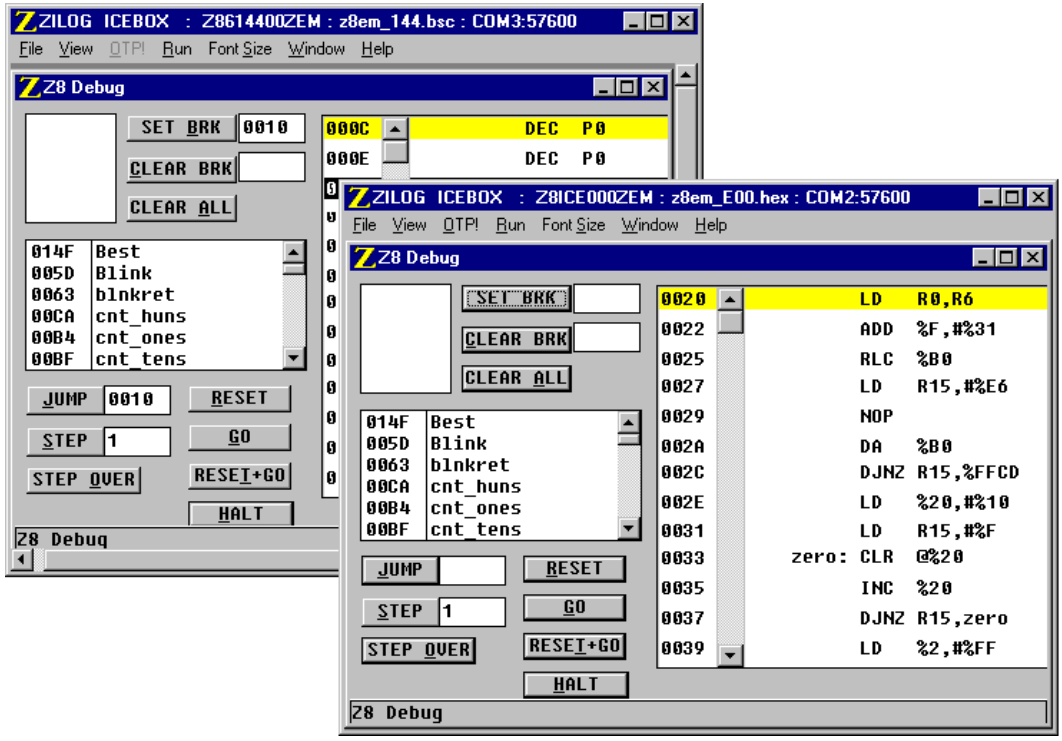


Figure 3-24. Multiple Emulator Operation Using Two COM Ports

Totally Logical

CHAPTER 4

TROUBLESHOOTING GUIDE

INTRODUCTION

Before contacting your ZiLOG representative or submitting a Problem Report, please follow these simple steps. Also, check the Precautions and Limitations sections in the Data Sheet included with your emulator to eliminate other possible known problems. If a hardware failure is suspected, contact your local ZiLOG representative for assistance.

INITIAL SCREEN DOES NOT APPEAR

The initial ZiLOG screen does not appear after selecting a COM port, and a message is displayed that begins "Time-out while reading..."

1. Check the RS-232 cable connection or try another cable that is known to be good.
2. Check if transmit/receive signals need to be swapped.

NOTE: On some DB9 connectors for the COM ports, the transmit/receive signal may be swapped and a Null Modem adapter may be required.

3. Verify that the power supply is connected and turned ON, and that power is available.
4. Verify that the power supply is set at the correct voltage.
5. Check that the power supply can supply the required 0.5 Amp current to the emulator.
6. Check selected port using another application or select another COM port.
7. After resetting the emulator, wait 5 seconds (minimum) before running the GUI software.

V_{CC} CONTENTION

To avoid V_{CC} contention when a separate V_{CC} is supplied to the target, ensure that jumper JP1 is not connected.

SELF TEST FAILURE

1. If any of the self tests fail, ensure that the target crystal labeled `Y1` and `Y2` is securely in its socket; then check `U18` for damaged, broken, or bent leads or cold solder joints.
2. If "Monitor RAM fail" appears, then check `U13` for damage or broken/bent leads/cold solder joints.
3. If "Break point RAM fail" or "User RAM fail" appears, then check `U12` or `U13` for damaged, broken, or bent leads or cold solder joints.
4. If the failure recurs after pressing the MASTER RESET button, power-down the emulator, then power it up again, and press the MASTER RESET button again.
5. The OTP chip should be removed from the socket during power up.

OTP PROGRAM FAILURE

1. Reset OTP device in socket and make sure there is good contact between socket and device leads, then reprogram the OTP device.
2. Check for a worn socket. Replace the socket if required.
3. Additional troubleshooting tips are provided in the `OTPPROG.TXT` file or icon.

XTAL CONTENTION

Do not install `JP3` and `JP4` on `1-2` at the same time.

HALT BUTTON DOES NOT STOP EMULATOR EXECUTION

If the application inadvertently goes into Stop mode, the only way to halt the emulator execution is by doing a Stop-Mode Recovery (as defined by the user program). You may also reset the application using the TARGET RESET button.

CHECKSUM DIFFERENCES

Checksum differences may occur when during download the following three conditions occur:

- User program is smaller than emulation memory.
- User-program memory outside the user program is padded with code.
- The `Nothing` item is selected in the Memory Pad With section of the Download to Z8 Code Memory dialog box.

When this situation occurs, results of the user program are not affected. However, the calculation of the Checksum during OTP programming is affected because the Checksum is calculated throughout the entire OTP memory area.

SLOW RISE AND FALL TIMES

Low-current pull-ups/pull-downs/drivers may exhibit slower rise and fall times when the emulation cable is used due to additional inductance and capacitance of the cable.

"CAN'T OPEN WINDOWS" MESSAGE

If the "Can't Open Windows" message appears while attempting to open a window using the GUI software, there may not be enough memory within the Microsoft Windows environment to properly run the GUI software. Try closing the other active applications or exit and reenter the Microsoft Windows environment.

"OUT OF SYNCHRONIZATION WITH THE EMULATOR" MESSAGE

This message appears whenever communication between the emulator and the PC is interrupted.

1. Ensure that the power cable is connected.
2. Ensure that the RS-232 cable is connected.
3. Change the baud-rate setting. the default value is `19200`. A lower setting usually improves communications reliability.
4. The Emulator hardware RESET button was pressed while the GUI was running.

   There are three responses to the message:

   `Abort` Quits the GUI altogether.

   `Retry` Resets the GUI by reinitializing. You are returned to the initial screen with the ZiLOG logo.

   `Ignore` Attempts to reestablish communication between the GUI and emulator without reinitializing.

PROGRAM COUNTER JUMPS TO AN UNEXPECTED ADDRESS

1. Any instruction other than a `DI` instruction was used to disable interrupts.
2. The stack over flowed into the general register locations.
3. Extra `POP`, `PUSH`, `IRET`, or `RET` was encountered.
4. Program keeps resetting.
5. Program counter rolled over from value `FFFF` to `0000` and proceeded back to beginning of program.
6. Watch-Dog Timer (WDT) was not refreshed.
7. Unintialized interrupt vector was activated.

Totally Logical

APPENDIX A

ACCESSING THE ZBBS/INTERNET

BULLETIN BOARD INFORMATION

The ZiLOG Bulletin Board Service (ZBBS) currently provides basic information on ZiLOG products and includes a ROM CODE upload area. In addition, the ZBBS provides valuable information on items of interest, such as ZiLOG specialty software and documentation.

How to Access the ZBBS

The ZBBS can be reached by dialing 1-408-558-8890. The ZBBS supports speeds up to 28.8K Baud with connections 8-N-1 (8 bits, No parity, 1 stop bit). We recommend that you use an ANSI/BBS terminal emulation setup.

To preview information or download files, follow the on-screen instructions.

The latest production released version of the Z8 GUI software can be downloaded from this site.

ZILOG ON THE INTERNET

ZiLOG has a Home Page on the Internet. The Home Page address is:

http://www.zilog.com

The ZiLOG Home Page includes valuable information about hardware and software development tools. The latest production released version of the Z8 GUI software can be downloaded from this site.

Totally Logical

APPENDIX B

ASCII CHARACTER SET

ASCII CHARACTER SET

Table B-1. ASCII Character Set

Graphic	Decimal	Hexadecimal	Comments
	0	0	Null
	1	1	Start Of Heading
	2	2	Start Of Text
	3	3	End Of Text
	4	4	End Or Transmission
	5	5	Enquiry
	6	6	Acknowledge
	7	7	Bell
	8	8	Backspace
	9	9	Horizontal Tabulation
	10	A	Line Feed
	11	B	Vertical Tabulation
	12	C	Form Feed
	13	D	Carriage Return
	14	E	Shift Out
	15	F	Shift In

Table B-1. ASCII Character Set (Continued)

Graphic	Decimal	Hexadecimal	Comments
	16	10	Data Link Escape
	17	11	Device Control 1
	18	12	Device Control 2
	19	13	Device Control 3
	20	14	Device Control 4
	21	15	Negative Acknowledge
	22	16	Synchronous Idle
	23	17	End Of Block
	24	18	Cancel
	25	19	End Of Medium
	26	1A	Substitute
	27	1B	Escape
	28	1C	File Separator
	29	1D	Group Separator
	30	1E	Record Separator
	31	1F	Unit Separator
	32	20	Space
!	33	21	Exclamation Point
"	34	22	Quotation Mark
#	35	23	Number Sign
$	36	24	Dollar Sign
%	37	25	Percent Sign
&	38	26	Ampersand
'	39	27	Apostrophe

Table B-1. ASCII Character Set (Continued)

Graphic	Decimal	Hexadecimal	Comments
(	40	28	Opening (Left) Parenthesis
)	41	29	Closing (Right) Parenthesis
*	42	2A	Asterisk
+	43	2B	Plus
,	44	2C	Comma
-	45	2D	Hyphen (Minus)
.	46	2E	Period
/	47	2F	Slant
0	48	30	Zero
1	49	31	One
2	50	32	Two
3	51	33	Three
4	52	34	Four
5	53	35	Five
6	54	36	Six
7	55	37	Seven
8	56	38	Eight
9	57	39	Nine
:	58	3A	Colon
;	59	3B	Semicolon
<	60	3C	Less Than
=	61	3D	Equals
>	62	3E	Greater Than
?	63	3F	Question Mark

Table B-1. ASCII Character Set (Continued)

Graphic	Decimal	Hexadecimal	Comments
@	64	40	Commercial At
A	65	41	Uppercase A
B	66	42	Uppercase B
C	67	43	Uppercase C
D	68	44	Uppercase D
E	69	45	Uppercase E
F	70	46	Uppercase F
G	71	47	Uppercase G
H	72	48	Uppercase H
I	73	49	Uppercase I
J	74	4A	Uppercase J
K	75	4B	Uppercase K
L	76	4C	Uppercase L
M	77	4D	Uppercase M
N	78	4E	Uppercase N
0	79	4F	Uppercase 0
P	80	50	Uppercase P
Q	81	51	Uppercase Q
R	82	52	Uppercase R
S	83	53	Uppercase S
T	84	54	Uppercase T
U	85	55	Uppercase U
V	86	56	Uppercase V
W	87	57	Uppercase W

Table B-1. ASCII Character Set (Continued)

Graphic	Decimal	Hexadecimal	Comments
X	88	58	Uppercase X
Y	89	59	Uppercase Y
Z	90	5A	Uppercase Z
[	91	5B	Opening (Left) Bracket
\	92	5C	Reverse Slant
]	93	5D	Closing (Right) Bracket
^	94	5E	Circumflex
_	95	SF	Underscore
`	96	60	Grave Accent
a	97	61	Lowercase A
b	98	62	Lowercase B
c	99	63	Lowercase C
d	100	64	Lowercase D
e	101	65	Lowercase E
f	102	66	Lowercase F
g	103	67	Lowercase G
h	104	68	Lowercase H
i	105	69	Lowercase I
j	106	6A	Lowercase J
k	107	6B	Lowercase K
1	108	6C	Lowercase L
m	109	6D	Lowercase M
n	110	6E	Lowercase N
o	111	6F	Lowercase O

Table B-1. ASCII Character Set (Continued)

Graphic	Decimal	Hexadecimal	Comments
p	112	70	Lowercase P
q	113	71	Lowercase Q
r	114	72	Lowercase R
s	115	73	Lowercase S
t	116	74	Lowercase T
u	117	75	Lowercase U
v	118	76	Lowercase V
w	119	77	Lowercase W
x	120	78	Lowercase X
y	121	79	Lowercase Y
z	122	7A	Lowercase Z
{	123	7B	Opening (Left) Brace
|	124	7C	Vertical Line
}	125	7D	Closing (Right) Brace
~	126	7E	Tilde
	127	7F	Delete

APPENDIX C

EMULATOR SCHEMATIC

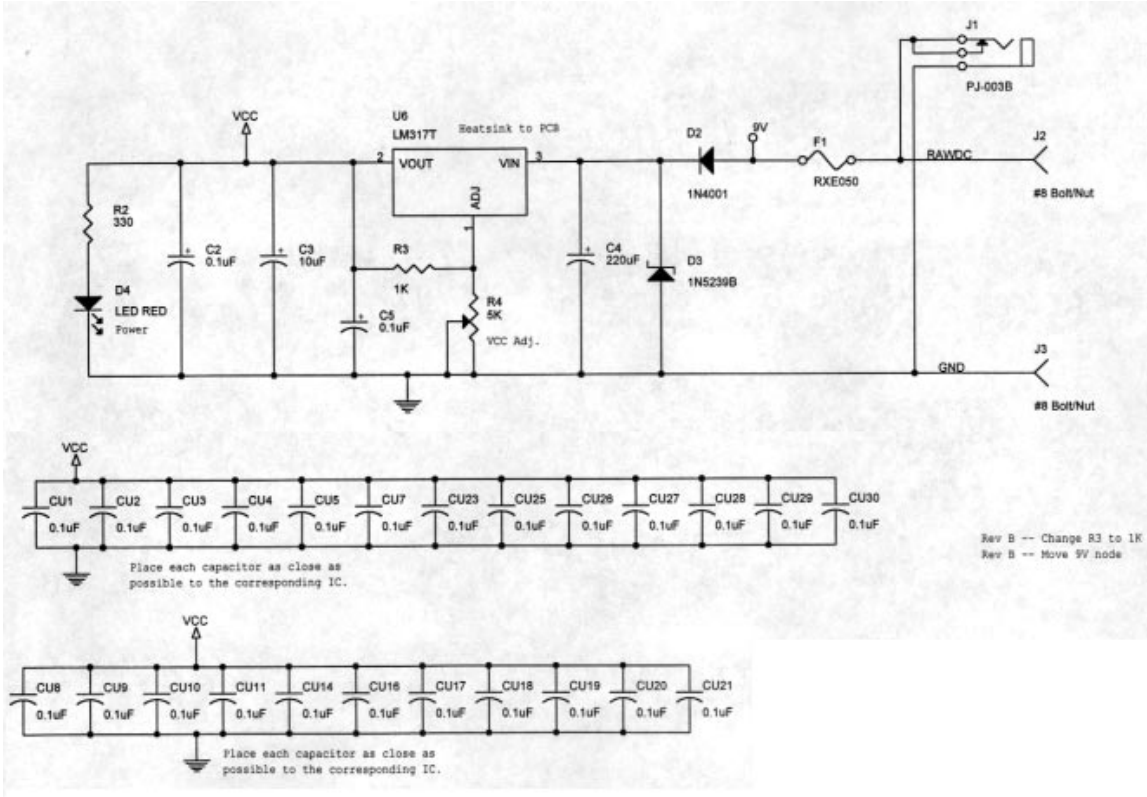


Figure C-1. Z8M001 Schematic, #1 of 10

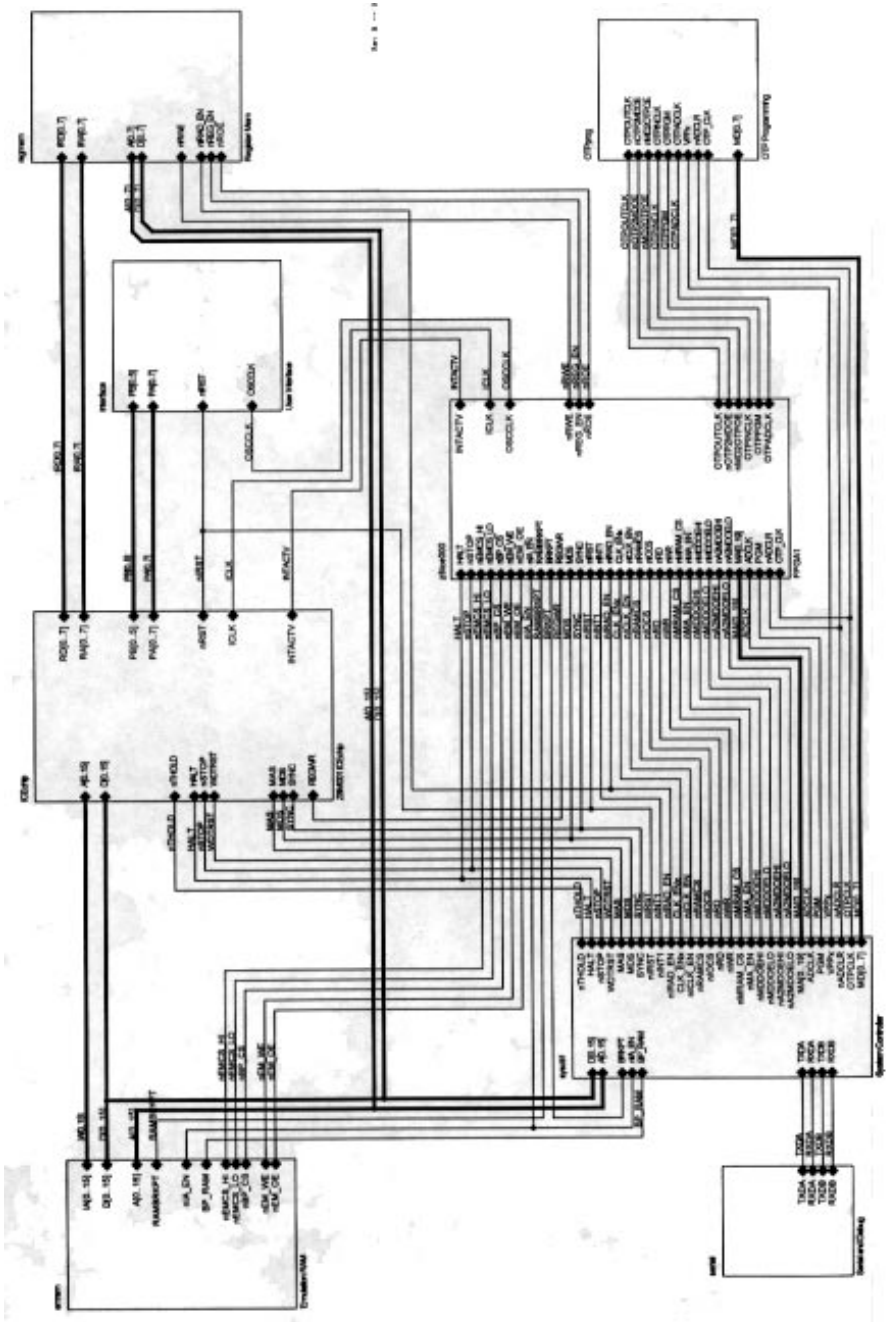


Figure C-2. Z8M001 Schematic, #2 of 10

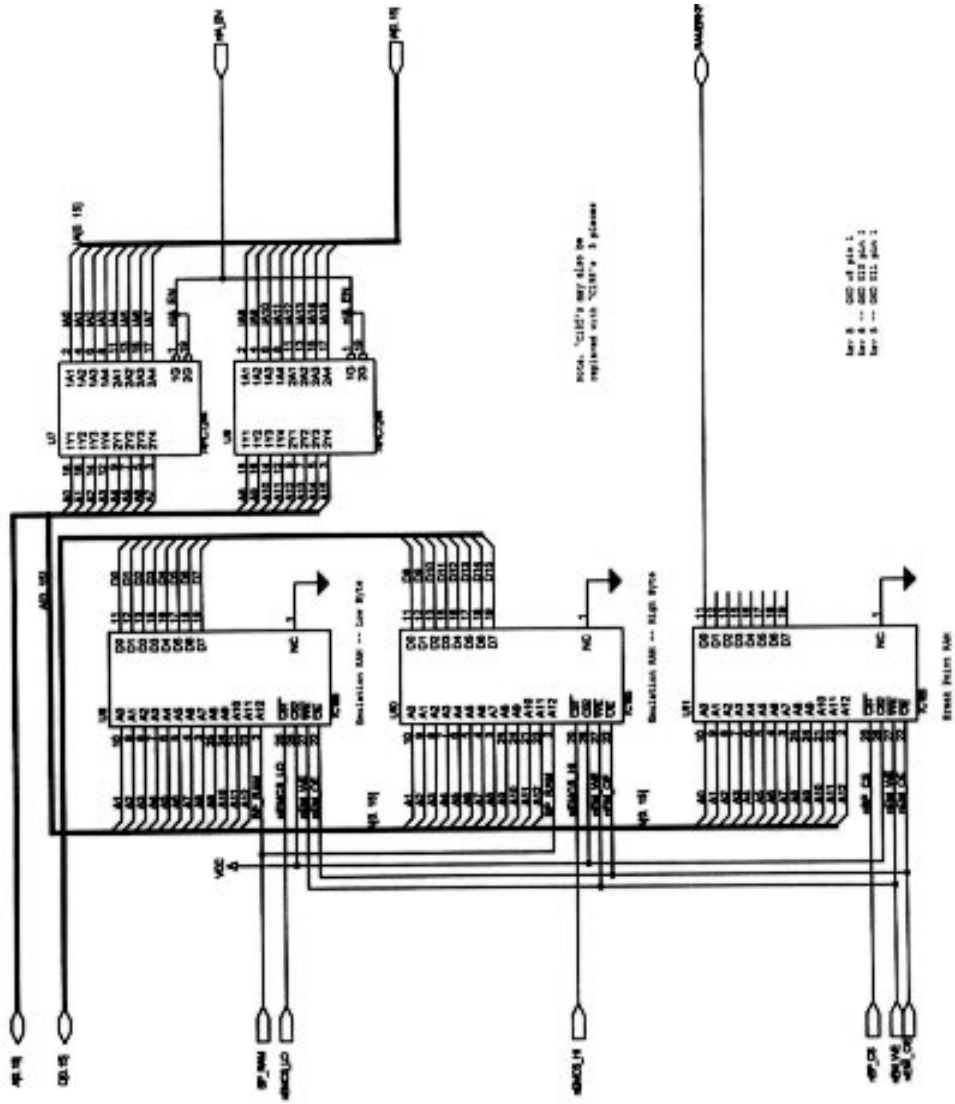


Figure C-3. Z8M001 Schematic, #3 of 10

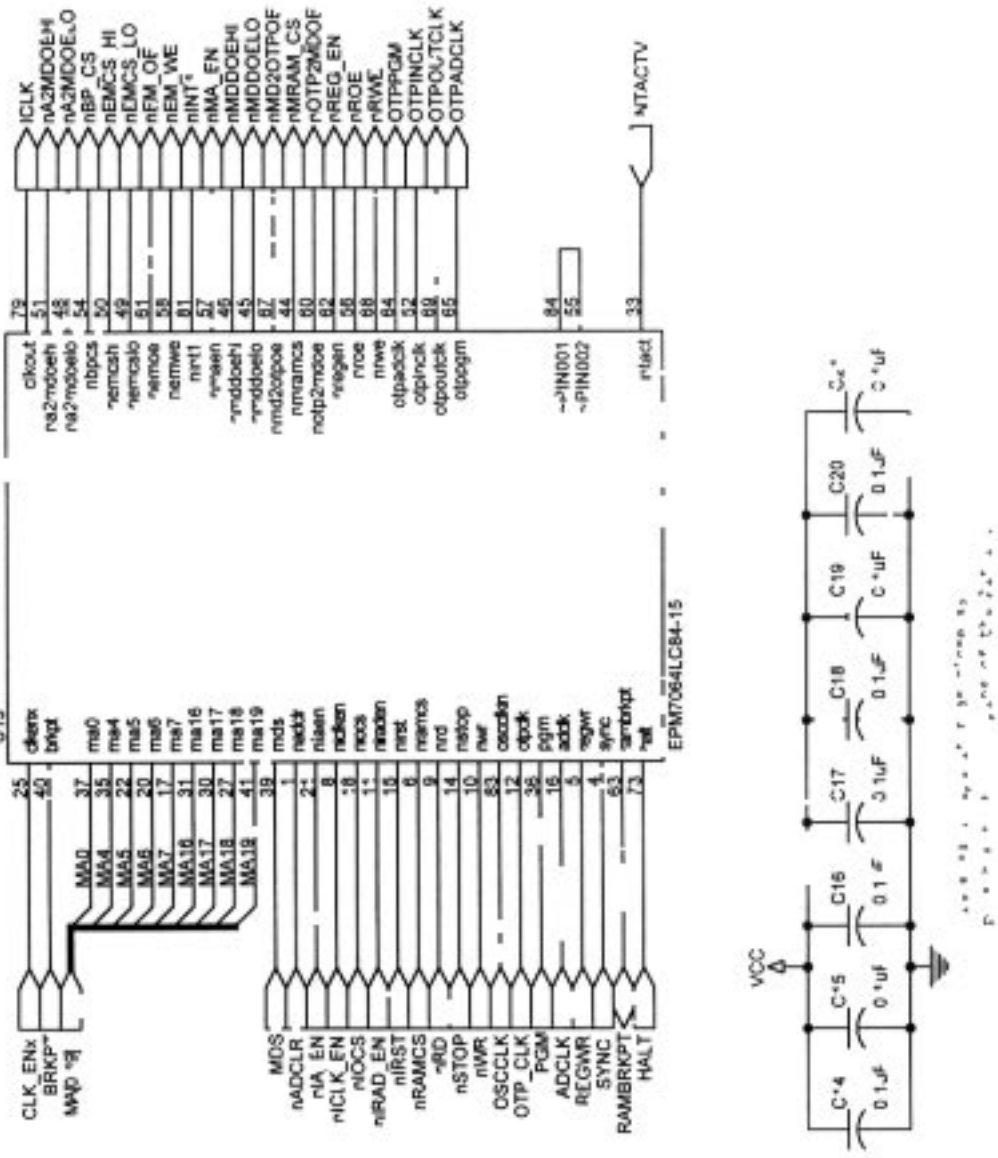


Figure C-4. Z8M001 Schematic, #4 of 10

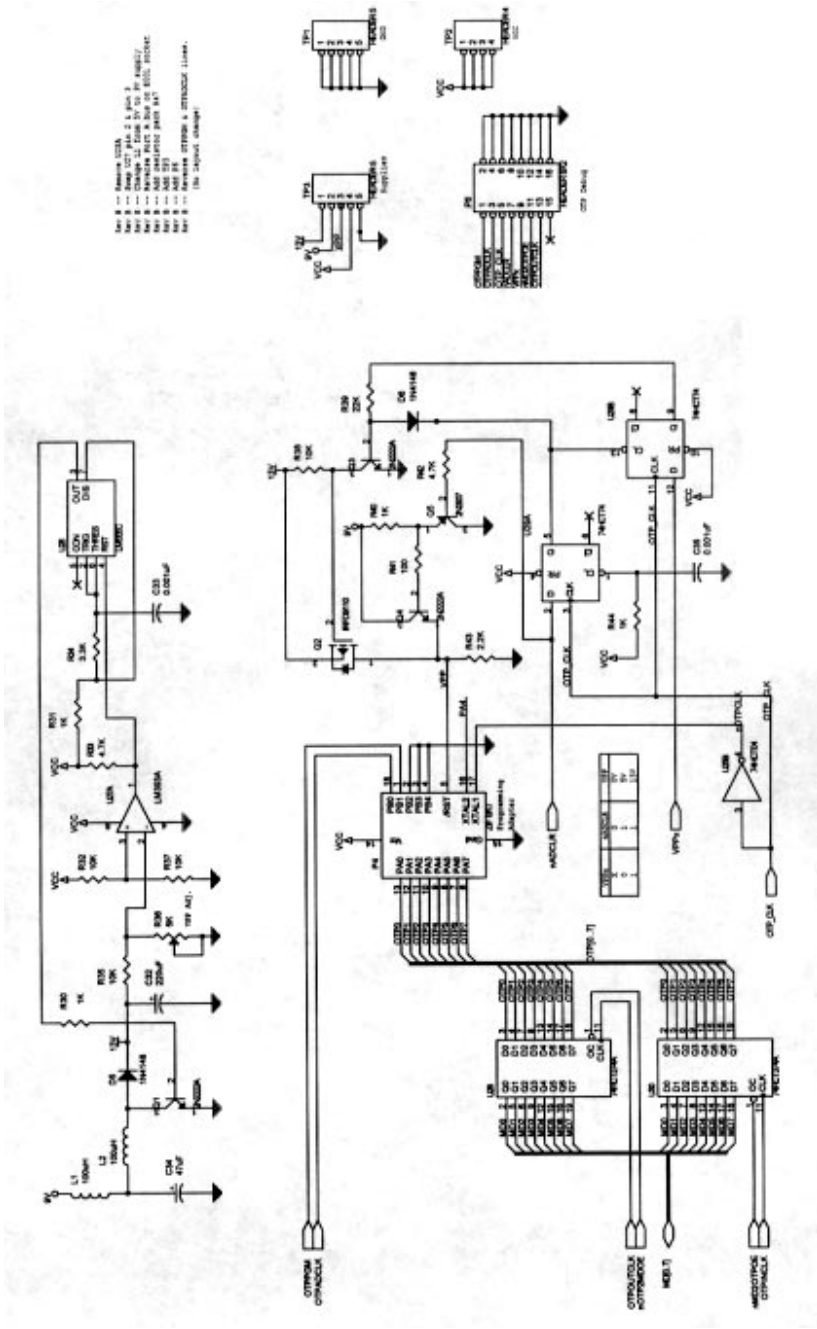


Figure C-5. Z8M001 Schematic, #5 of 10

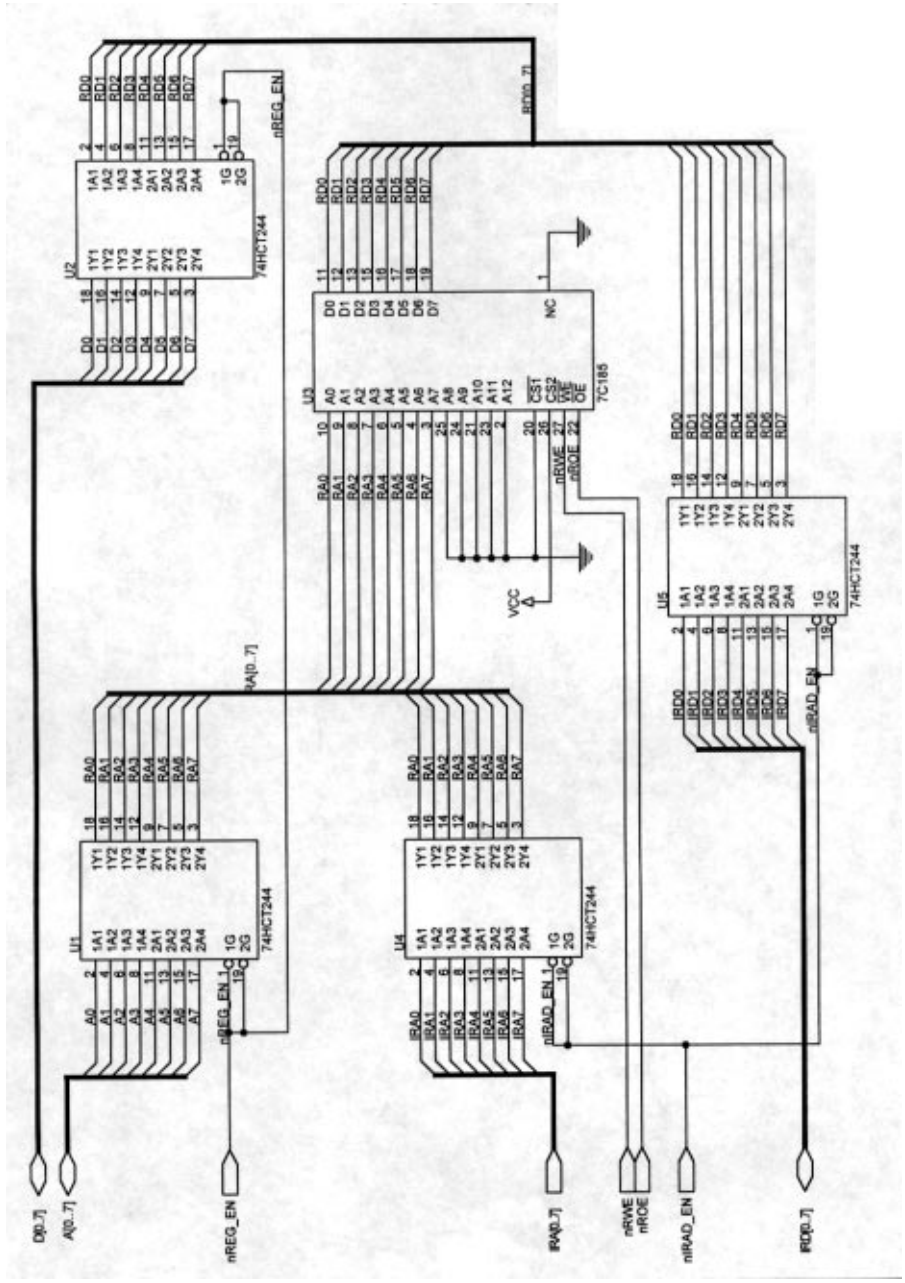


Figure C-6. Z8M001 Schematic, #6 of 10

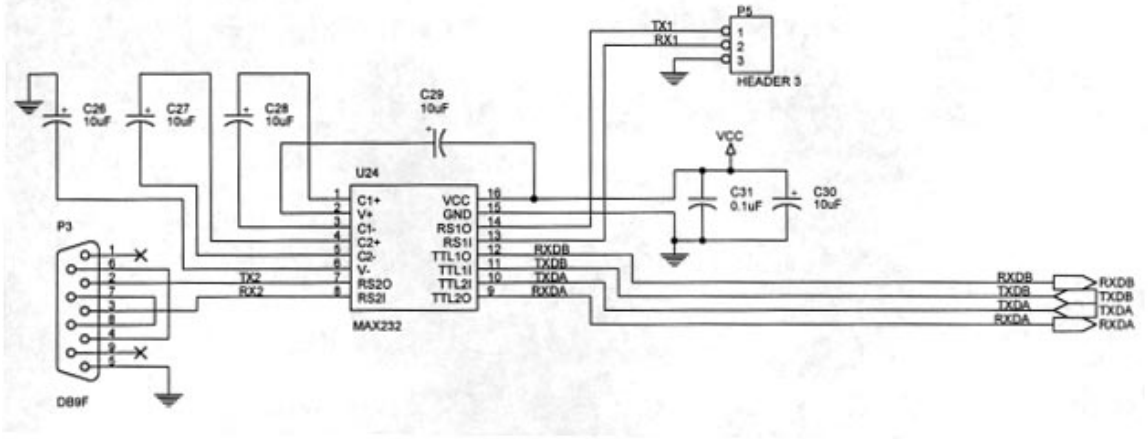


Figure C-7. Z8M001 Schematic, #7 of 10

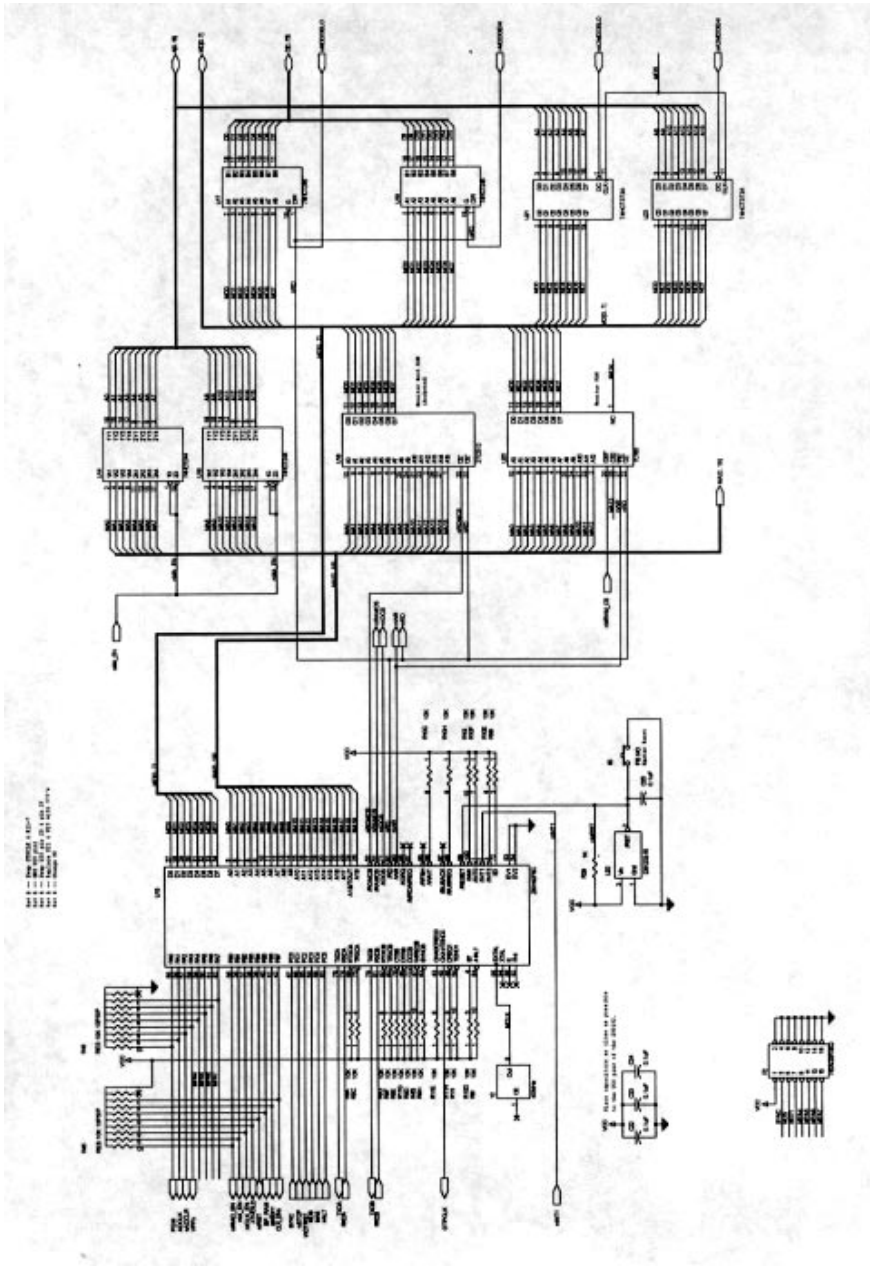


Figure C-8. Z8M001 Schematic, #8 of 10

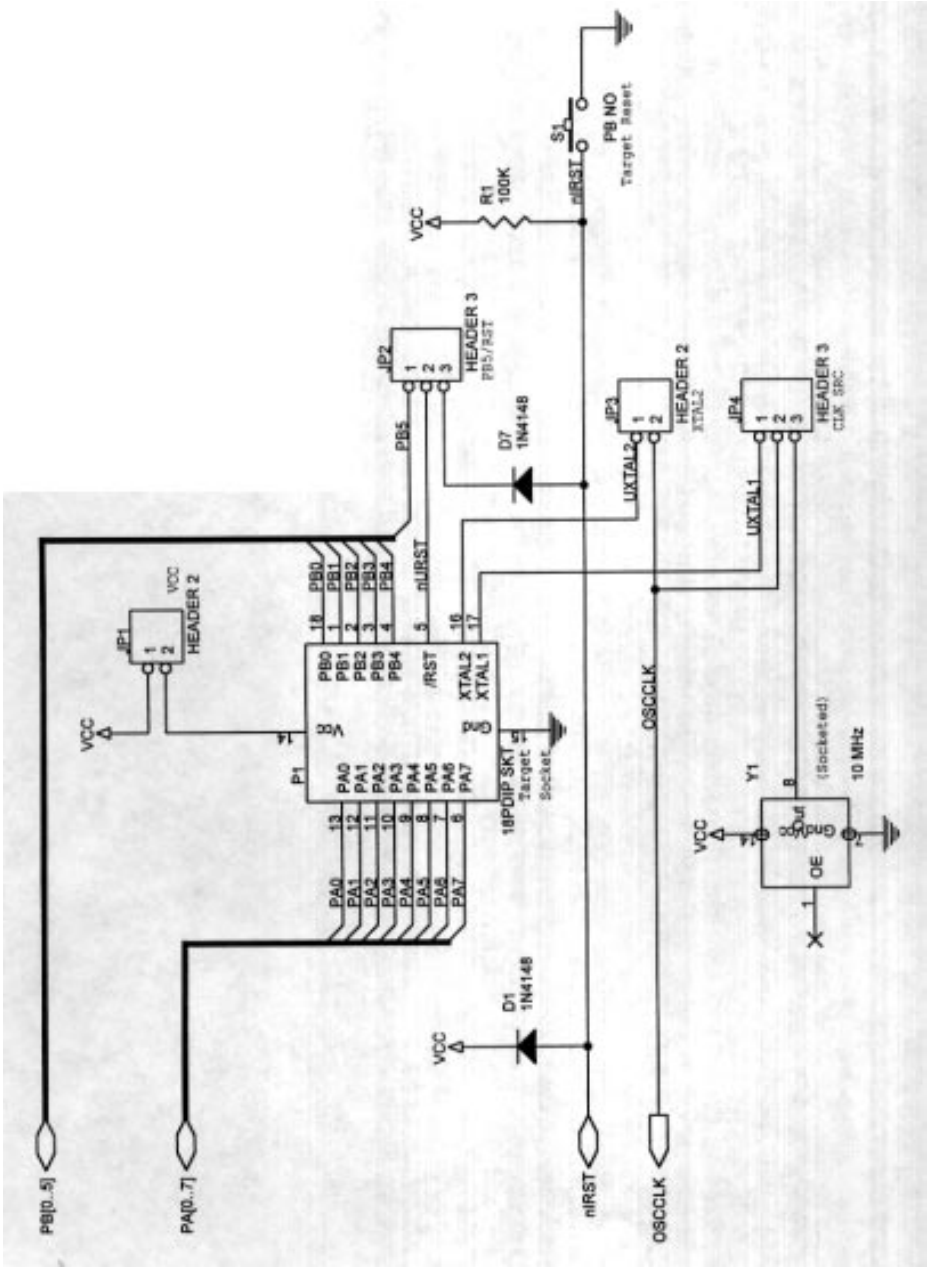


Figure C-9. Z8M001 Schematic, #9 of 10

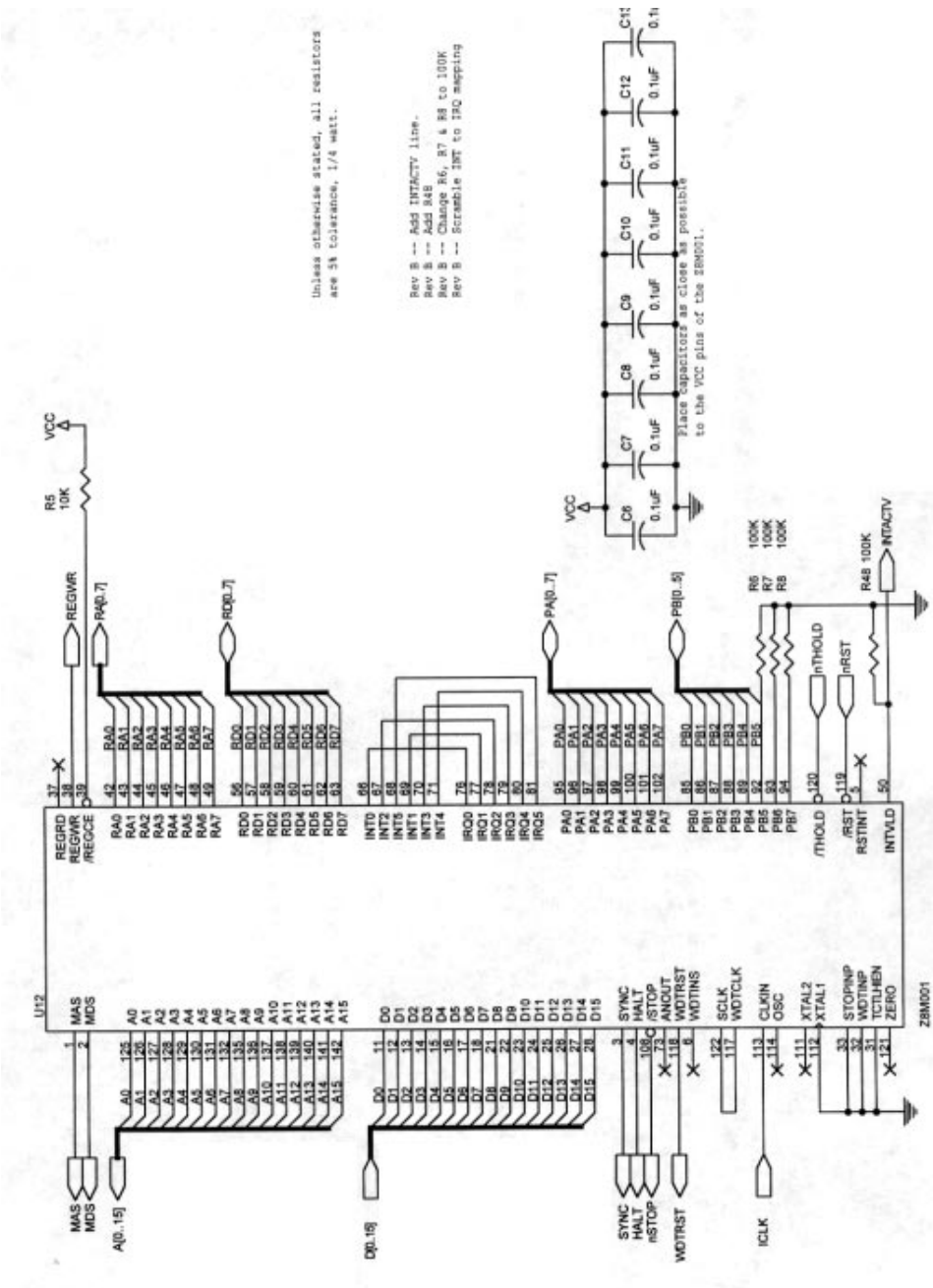


Figure C-10. Z8M001 Schematic, #10 of 10

Totally Logical

APPENDIX D

PROBLEM/SUGGESTION REPORT FORM

If you experience any problems while operating this product, or if you note any inaccuracies while reading the User's Manual, please copy this form, fill it out, then mail or fax it to *ZiLOG* (see "Return Information"). We also welcome your suggestions!

Customer Information

Name		Country	
Company		Telephone	
Address		Fax Number	
City/State/ZIP		E-Mail Address	

Product Information

Serial # or Board Fab #/Rev. #

Software Version

Manual Number

Host Computer Description/Type

Return Information

ZiLOG, Inc.

System Test/Customer Support

910 E. Hamilton Ave., Suite 110, MS 4-3

Campbell, CA 95008

Fax Number: (408) 558-8536

Email: tools@zilog.com

Problem Description or Suggestion

Provide a complete description of the problem or your suggestion. If you are reporting a specific problem, include all steps leading up to the occurrence of the problem. Attach additional pages as necessary.