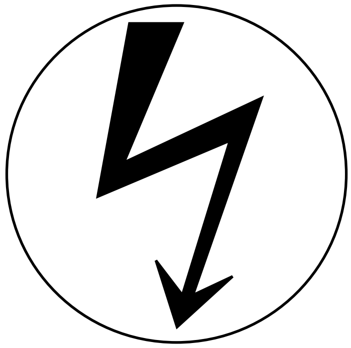
Electrical  Safeguards

WARNING:

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

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

II. Power-Up Precautions

1. Power up the PC (or dumb terminal) and ensure that it is running properly.
2. Load the Z8S180 Sample Files and Monitor Source Diskette.
3. Apply power through the "VCC" and "GND" connections on the Z8S180 Board.

III. Power-Down Precautions

1. Quit the monitor program.
2. Power down the PC (or dumb terminal).
3. Remove power from the Z8S180 Board.

Z8S180 EMULATOR USER'S MANUAL

TABLE OF CONTENTS

TITLE	PAGE

TABLE OF CONTENTS
(Continued)

Z8S180 Emulator
User's Manual

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: some will want to set up and use this emulator while they read about it; others will open these pages only as a “last resort” to check on a particular specification. Therefore, we have designed this manual to be used either as a “how to” procedural manual or a reference guide to important data.

Note that the complete Z8S180 Emulator Schematic Diagram is included at the back of this user's manual.

Please fill out and return the enclosed Zilog Registration Card as soon as possible so we can advise you of updates and improvements to your Zilog Z8S180 Emulator.

Zilog, Inc. 210 East Hacienda Ave.
Campbell, CA 95008-6600
Telephone (408) 370-8000
Telex 910-338-7621
FAX 408 370-8056
Internet: http://www.zilog.com

CHAPTER 1

INTRODUCTION

OVERVIEW

The Z8S180 Emulator (Z8S18000ZEM) is a cost-effective, in-circuit emulator for design and development using the Z8S180 Microprocessor Unit (MPU). The purpose of the Z8S180 Emulator is twofold:

1) Provide a Learning Environment for Students of the Z80® and Z180® Architecture

2) Provide a Cost-Effective Emulator for Z8S180 Design-Related Activities

The Z8S180 Emulator includes a Debug Monitor (in EPROM) that can be used with a PC (or a dumb terminal) and a software program (TZ.EXE) that provides terminal emulation facilities for a PC. Users of the Z8S180 Emulator can create software on the PC by using a Zilog assembler, linker, and other development tools included with the Z8S180 Emulator, or using Z80/Z180 software development tools from other companies. This software then can be downloaded from the PC to the Z8S180 Emulator for execution, by means of the TZ program and the Debug Monitor.

The Z8S180 Emulator does not require a target board for initial debugging and development because the 8 KB of RAM can be assigned to any 8 KB block within the first 64 KB of address space. For larger applications and advanced development, the Z8S180 Emulator can be plugged into its mating 68-Pin PLCC Emulation Adapter, which, in turn, is plugged into the Z8S180 socket of the user's target board for additional RAM and ROM. The Z8S180 also includes header pins for logic-analyzer support for hardware development and software debugging. The Z8S180 includes logic that implements switching between a “Monitor” and a “User” state, making the Z8S180 Emulator is a true in-circuit emulator.

The Z8S180 Emulator is carefully engineered to provide the best balance between reasonable cost and useful features to shorten your development time for products using the Z8S180 (see Figure 1-1 for Functional Block Diagram). The Z8S180 Emulator provides an excellent environment for hardware and software development.

KEY FEATURES

- Effective Learning Tool for Students of Z80 and Z180 Architecture
- Simple Emulator Environment for Z8S180-Based Design and Development
- Serves as a Developmental Platform for Trial Implementation of a Specific Application
- Configured to Operate as a Stand-Alone Unit

SUPPORTED ZILOG DEVICES

Device	**Packaging**
Z8S180	68-Pin PLCC

HARDWARE SPECIFICATIONS

Dimensions	4.2 in. H x 4.2 in W
Oscillator Frequency (OSC)	18.432 MHz
Host Interface	RS-232 (EIA-232)
Serial Baud Rate	1,200 to 57,600 Bits/sec.
Power Supply Voltage	+5 VDC ±5%
Power Supply Current	Less than 1A
Operating Temperature	20 degrees C, ±10 degrees C
Operating Humidity	10-90% RH (non-condensing)

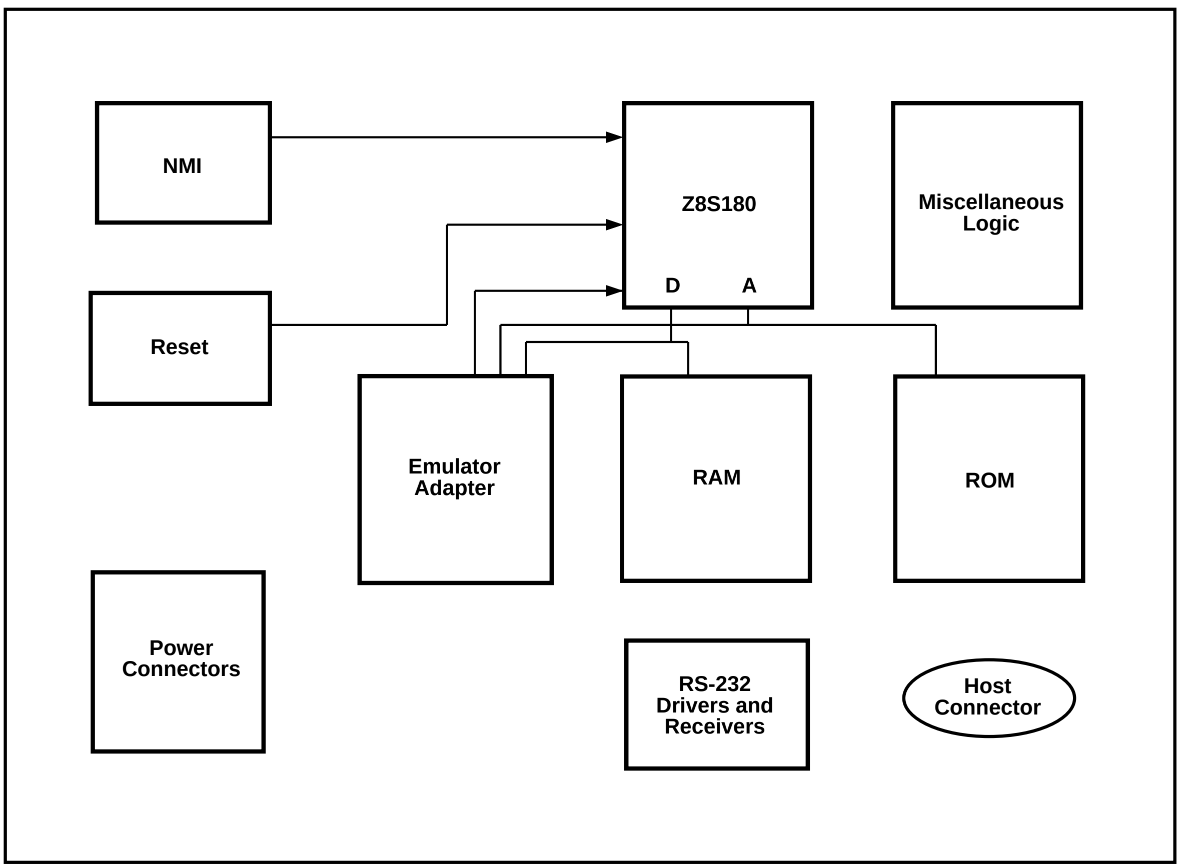


Figure 1-1. Z8S180 Emulator Functional Block Diagram

SUPPORT PRODUCTS PACKAGE CONTENTS

Hardware
Z8S180 Emulator Board, including:

Zilog Z8S180 MPU
18.432 MHz Crystal
8Kx8 EPROM (Contains Debug Monitor Program)
8Kx8 SRAM
Altera FPGA (Controls Emulation and ROM and RAM Decoding)
RS-232 Line Drivers and Receivers
Test Points Surrounding the Z8S180 Location (for all MPU signals)
NMI and RESET buttons

Z8S180 68-Pin PLCC Emulation Adapter

Software

Z8S180 Sample Files and Monitor Source Diskette
ZASM-Cross Assembler Diskette/MOBJ-Object Utility Diskette

Z8S180 Files on Source Diskette

MAKE.BAT	Assembly Macro
FIXLSTG.EXE	Listing-Fixer Executable
TZ.EXE	Terminal Emulator Executable
180MACRO.LIB	Z8S180 Include file and macros
180MON.HEX	Z8S180 Debug Monitor Source (Intel hex format)
180MON.S	Z8S180 Debug Monitor Source
DISTEST.HEX	Disassembly Test (Intel hex format)
DISTEST.S	Disassembly Test Source
TSTRST20.HEX	Test Program for Monitor Services (Intel hex format)
TSTRST20.S	Test Program for Monitor Services Source
TZ96COM1.PIF	Windows Startup for TZ on COM1
TZ96COM2.PIF	Windows Startup for TZ on COM2

Publications

Z8S180 Emulator User's Manual
asm 800, Z800 Cross Assembler User's Guide
Zilog Universal Object File Utilities User's Guide
Z80180/Z180 MPU User's Manual
Z180 Family Microprocessors and Peripherals Databook

REQUIREMENTS AND RESTRICTIONS

Minimum Hardware

The Z8S180 Emulator can be used with a dumb terminal and a power supply; however, a PC is recommended to take full advantage of the development kit's software downloading and development capabilities.

If Using a PC

Any IBM PC (or 100-percent compatible) that can run MS-DOS V.5.0. We recommend an IBM PC (or 100-percent compatible) 386-based machine at 20 MHz with 4 MB RAM, hard disk drive (with 1 MB available), and a 3.5 floppy disk drive (see “Notes” that follow).

Minimum Software Operating Systems

MS-DOS V.5.0 (see “Notes” that follow)

Additional Items Not Supplied with the Support Package

A source of power (+5 VDC ±10%) can be used in place of the PC. This can be a laboratory power supply with supply current of 1.0A.

Notes:

1. Debug Monitor with a Dumb Terminal. Two Debug Monitor commands (“L” for loading a hex file and “N” for changing the Serial Data Rate) will not properly function when running on a dumb terminal (refer to Chapter 3: Using the Debug Monitor, “TZ Program Restrictions”).

2. TZ Terminal Emulation Program. The TZ Program, which is included on the Z8S180 Sample Files and Monitor Source Diskette, was developed to run with MS-DOS V.5.0; however, the program may run on earlier versions. One (or more) copies of the TZ Program will run under Windows 3.1 (and also may run under earlier versions). The kit includes .PIF files on the Z8S180 Sample Files and Monitor Source Diskette to help start the TZ program in the Windows environment.

3. PC Models/Serial Baud Rates. The maximum serial rate that can be used between a PC and the development kit board is dependent upon which PC model and configuration that is used. If the baud rate is too fast for the PC, characters will be lost during lengthy display sequences initiated by certain commands. Also, downloading may fail if the serial rate is too fast.

General Guidelines:

PC Configuration	Maximum Baud Rate (bps)
286 or 386 (slower)	19,200
386 (faster), 486, Pentium	57,600

Note: The RS-232 drivers and receivers used on the Z8S180 Emulator are not capable of operating at 115,200 bits/second.

Target Design Restrictions

/MREQ

In order for the Z8S180 emulator to emulate the processor in a target board, the target board must enable its memories only when the /MREQ signal is low. The target board may include /MREQ in a /CS signal, or in /OE and /WE signals, of its memories.

The Z8S180 emulator avoids conflicts with target resources during cycles that access its ROM or RAM by blocking the /MREQ signal to the target board.

Clocking

A crystal or oscillator on the target board is not used by the Z8S180 emulator. In order to use a different clock rate than the standard 18.432 MHz, the following steps must be taken:

1. A different crystal, of the desired frequency, must be plugged into the crystal socket on the Z8S180 emulator.
2. The symbol “clockHz” in the Debug Monitor source (included with the emulator) must be changed to the new value.
3. The MAKE.BAT file, assembler, and object utilities included with the emulator must be used to assemble the source and create a hex file.
4. A corresponding new EPROM must be programmed and installed in the U2 socket.

Note: Steps 2–4 can be avoided if the new frequency is a simple multiple or divisor of the old. Just start the TZ program at whatever serial rate the new crystal plus the old EPROM result in. (Refer to Chapter 3, “Using the Debug Monitor.”)

Execution Restrictions

While running a program in User Mode, execution of an instruction starting at any of the following addresses makes the emulator enter Monitor Mode:

Address	Significance
00000	Reset, illegal instruction trap, RST 0, or jumping to a Null pointer.
00020	RST 20 instruction, used to provide services to user program.
00028	RST 28 instruction, used for breakpoints.
00066	Non-Maskable Interrupt.

User programs can easily work around the first problem by including a NOP at 00000, and starting execution under the Debug Monitor at 00001.

There are five other RST instructions that applications can use.

For user programs intended for eventual stand-alone use, that is, for operation without the Debug Monitor, debugging of the user’s NMI routine must wait for final system integration, when the emulator is no longer plugged into the target board.

REQUIREMENTS AND RESTRICTIONS (Continued)

Last 256 Bytes of Emulator RAM Protected

The Z8S180 Emulator includes 8 KB of RAM that you can locate at various 8 KB address boundaries, or optionally eliminate from the user memory map. When you do include this RAM, its last 256 bytes are protected/hidden from access by your software and by monitor commands to protect the monitor's internal states, variables, and tables.

CHAPTER 2
SETUP AND INSTALLATION

INTRODUCTION

This chapter describes the various steps necessary to start development using the Z8S180 Emulator. The sections covered in this chapter are as follows:

- Installing the Software
- Setting Up the Hardware
- Connecting to Power
- Initial Checkout/Sample Session

INSTALLING THE SOFTWARE

Software for the Z8S180 Emulator is stored on two diskettes:
1. Z8S180 Sample Files and Monitor Source Diskette

2. Zilog ZASM Cross Assembler/Zilog MOBJ Object File Util. diskette

Z8S180 Source Diskette Installation

1. Select the "Run" command from the "File" menu under Microsoft Windows "Program Manager".

2. Insert the diskette labeled "Z8S180 Sample Files and Monitor Source" 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 will now prompt you for the directory into which the software will be installed (default is C:\180). The setup program will copy the files into the target directory, creating an icon in the Windows environment. After the installation is finished, you can move the icon into any program group of your choice.

   Note: The icon will be created in the window that is currently selected.

4. Remove diskette and store in a safe place when installation is completed.

INSTALLING THE SOFTWARE (Continued)

Creating TZ Program (TZ.EXE) Icon Using Windows Program Manager

This kit includes a software monitor program that runs on a PC. If you choose to create a TZ icon from which you can run the TZ program, perform the following steps:

1. Select “New” from the Program Manager’s “File” menu and select “OK” or press <ENTER> on the keyboard.

2. Type the designated name (such as “tz 9600 com1”) in the Program Manager window.

3. Type the full path and filename of (one of) the .PIF files you copied from the Source Diskette, such as “C:\WINDOWS\TZ96COM1.PIF”.

4. Type the full path of the directory you created (such as “C:\180”), then press <ENTER>.

The program item icon should then be created and ready to use. If you want to run copies of TZ on both COM1 and COM2, repeat Steps 1–4.

Notes:
1. Modification of the “win.ini”, “autoexec.bat”, or “config.sys” files is not required.
2. Consult MS-Windows documentation if you need additional information about alternate install procedures.
3. Refer to the README files on diskettes. (The README files are easily accessed via the Microsoft Windows "Notepad" program.)

ZASM-Cross Assembler/MOBJ-Object Utility Installation (Optional)

If you are using the ZASM Cross Assembler/MOBJ-Object File Utility, install the appropriate diskette **before** installing the GUI diskette. (You may choose to use a different assembler.)

1. Select the "Run" command from the "File" menu under Microsoft Windows "Program Manager".

2. Insert the diskette labeled "Zilog ZASM Cross Assembler/Zilog MOBJ Object File Util." into drive A (or drive B, if appropriate).

3. Type "a:\setup" and press <ENTER>. (Type "b:\setup" if drive B is used.)
   A screen now appears listing various installation options.

4. Select the desired installation option ("Full Installation" is the default selection; however, only Z8 installation is required.)

5. Press <ENTER> and follow on-screen instructions.

6. Remove diskette and store in a safe place when installation is completed.

Note: The installation procedure can be run before creating the installation directory.

SETTING UP THE HARDWARE

Serial Connections to the Host

The Z8S180 Emulator includes two serial channels, ASCI0 and ASCI1. Header J4 connects one of these channels to the RS-232 interface chip U7, which, in turn, is connected to the DB-9 Host connector P5.

Using ASCI0 (Instead of Default ASCI1) As User Interface

1. Move shunts to the other side of J4 (as follows):

From:	**To:**
(ASCI1)	(ASCI0)
J4 2-3, 5-6, 8-9, 11-12, 14-15	J4 1-2, 4-5, 7-8, 10-11, 13-14.

2. Connect a DB-9 to DB-9 "straight-through" (EGA Extender) cable between one of the COM ports of your PC and P5 on the Z8S180 board.

RS-232 Signals

The serial interface of the Z8S180 consists of four RS-232 signals: Tx and Rx data plus one control output and one status input.

Control Output

The control output drives pins 1, 6, and 8 of the DB-9 connector P5, (DCD, DSR, and CTS respectively). When using ASCI0, the control output is controlled by the 180's /RTS0 output; when using ASCI1, the control output is always asserted.

Control Input

The control input is taken from pin 7 of the DB-9 connector P5, which is RTS from the host. When using ASCI0, it is routed to both the /CTS0 and /DCD0 pins. When using ASCI1, it is routed to the RXS/CTS1 pin.

Receiving Data Using CSI0 Facility, While Using ASCI1 for Host Communications

To receive data using the Z8S180 Emulator's CSIO facility, while using ASCI1 for host communications, remove the shunt between J4-11 and J4-12.

Using ASCI0 for Host Communications Without "Request to Send"

To use ASCI0 for host communications, but your PC does not assert Request to Send, remove the shunts between J4-10 and J4-11, and between J4-13 and J4-14, and wire J4-10 and J4-13 to Ground, which is available on J4-9, P1-2, P1-9, P2-11, P3-17, P4-8, and P4-9.

RAM Size

16 KB (or larger) RAM at U4

The Z8S180 Emulator is shipped with an 8 KB RAM in the U4 location, and a shunt between J3-1 and J3-2. If you want to put a 16 KB (or larger) RAM at U4, move the shunt to between J3-2 and J3-3.

RAM Presence/Location in User Address Space

The J5 header controls whether the RAM on the emulator appears in the user (target) memory map, and if so, at what 8 KB boundary. It includes three separate shunt positions, between pins 1-2, 3-4, and 5-6. The emulator is shipped with all three shunts installed, which locates the RAM to start at address 00000. Other options include the following:

J5-5 to J5-6	J5-3 to J5-4	J5-1 to J5-2	Emulator RAM Starts At:
In	In	In	00000
In	In	Out	02000
In	Out	In	Not in User Space
In	Out	Out	06000
Out	In	In	08000
Out	In	Out	0A000
Out	Out	In	0C000
Out	Out	Out	0E000

Note: When you locate Z8S180 Emulator RAM in user space, the emulator “hides” the last 256 bytes of the RAM from being accessed by your software and your monitor commands, in order to protect its internal variables, states, and tables. It passes accesses to these 256 bytes on to the target board, if any.

EPROM Size

32 KB (27256) or larger ROM at U3

The Z8S180 Emulator is shipped with an 8 KB ROM in the U3 location, and a shunt between J2-1 and J2-2. If you want to put a 32 KB (27256) or larger ROM at U3, move the shunt to between J2-2 and J2-3.

RESET and NMI Drive to Target

The RESET and NMI push buttons always assert /RESET and /NMI to the Z8S180 processor. The J8 and J9 shunt positions control whether or not the emulator drives these signals to the target board as well.

Insert the J8 shunt to drive /RESET to the target, if the target board drives /RESET only with open-collector (open drain) driver(s). Leave J8 open if the target board drives /RESET with a totem pole (High and Low) driver.

Insert the J9 shunt to drive /NMI to the target, if the target board does not drive /NMI, or drives it only with open-collector (open drain) driver(s). Leave J9 open if the target board drives /NMI with a totem pole (high and low) driver.

No-Emulation Option

The Emulator board is normally configured with an Altera EPLD device at U2, which implements its in-circuit emulation functions. This part can be eliminated if emulation is not desired, which converts the board to a simple “evaluation board.”

If the U2 EPLD is present, leave shunt positions J1, J6, and J7 open. In this configuration, the Emulator board operates as described in this document.

If an EPLD is not present at U2, insert shunts at J1, J6, and J7. In this case all addresses with A15 low (including 00000-07FFF) select the ROM, and all addresses with A15 High (08000-0FFFF) select the RAM. If the ROM and RAM are both 8-KB devices as shipped, each is replicated four times within each 32-KB window. (This is the same as the address map in Monitor Mode when U2 is present.)

Connecting the Serial Cable to the PC. Locate the serial cable. Connect the serial cable plug to the socket on the emulator board, and the other end of the cable to either the COM1, COM2, COM3, or COM4 connector of your PC.

Note: If connector availability is limited to a 9-pin COM1 through COM4, a different cable or a 25-pin to 9-pin converter must be used. (Zilog does not provide either of these items.)

CONNECTING TO POWER

Connecting to Power Supply. (Refer to Figure 2-1, which follows.)

1. Turn the 2.0A power supply on and adjust it to +5V (if your power supply allows adjustment).
2. Set the +5V power supply for at least 1.5A, if there’s a current-limiting adjustment.
3. Turn the supplies off, or make sure nonadjustable supplies are off.
4. Connect the Ground screw and V_{CC} screw terminals on the emulator to Ground (or COM) and power connections (usually labeled "+" or "+V" or "+5V") on the power supply, respectively.
   Note: Some manufacturers will also have black or white jacks. Refer to individual manufacturers' manuals to decide how to connect in this case.

SETTING UP FOR IN-CIRCUIT EMULATION

(Refer to Figure 2-1, Z8S180 Emulator Board and PC, and Emulation Adapter Hook-Up Diagram)
If you already have a Z80180 board, or when your new design is built, you can use the Z8S180 Emulator in place of the Z180 processor chip. (See Chapter 1 for the requirements and restrictions involved in such use.) To perform in-circuit emulation, the target board must have a 68-pin PLCC socket for the 68-pin PLCC Emulation Adapter/Z8S180 Emulator.

.

1. Locate the 68-pin PLCC Emulation Adapter, which is supplied.

2. Looking at the “top” of both boards (the sides with yellow markings), align the Z8S180 Emulator Board and the Emulation Adapter so that the P1-4 connectors match up.

3. Plug the Emulation Adapter into the connectors on the “bottom” of the emulator board. Fully seat the connectors.

4. Align the Emulator Board and Emulation Adapter with your target board, so that the beveled corner on the Emulation Adapter matches up with the beveled corner of the 68-pin PLCC socket on the target board.

5. Plug the Emulation Adapter into the PLCC socket.

Power Distribution for In-Circuit Emulation. The Z8S180’s power pin and four Ground pins are connected through the included 68-Pin PLCC Emulation Adapter. When using the Z8S180 Emulator for in-circuit emulation, there are three power-distribution options (refer to Figure 2-1):

1. Leave V_{CC} and GND screw terminals on the emulator board open. Power is drawn from the target board.

2. Connect power to the emulator board. Target board draws its power from the emulator by means of the 68-Pin PLCC Emulation Adapter.

3. Connect power to both the emulator board and the target board. Each board then has lower-impedance power distribution than the other options.

   Note: This may cause “Ground loop” in some applications.

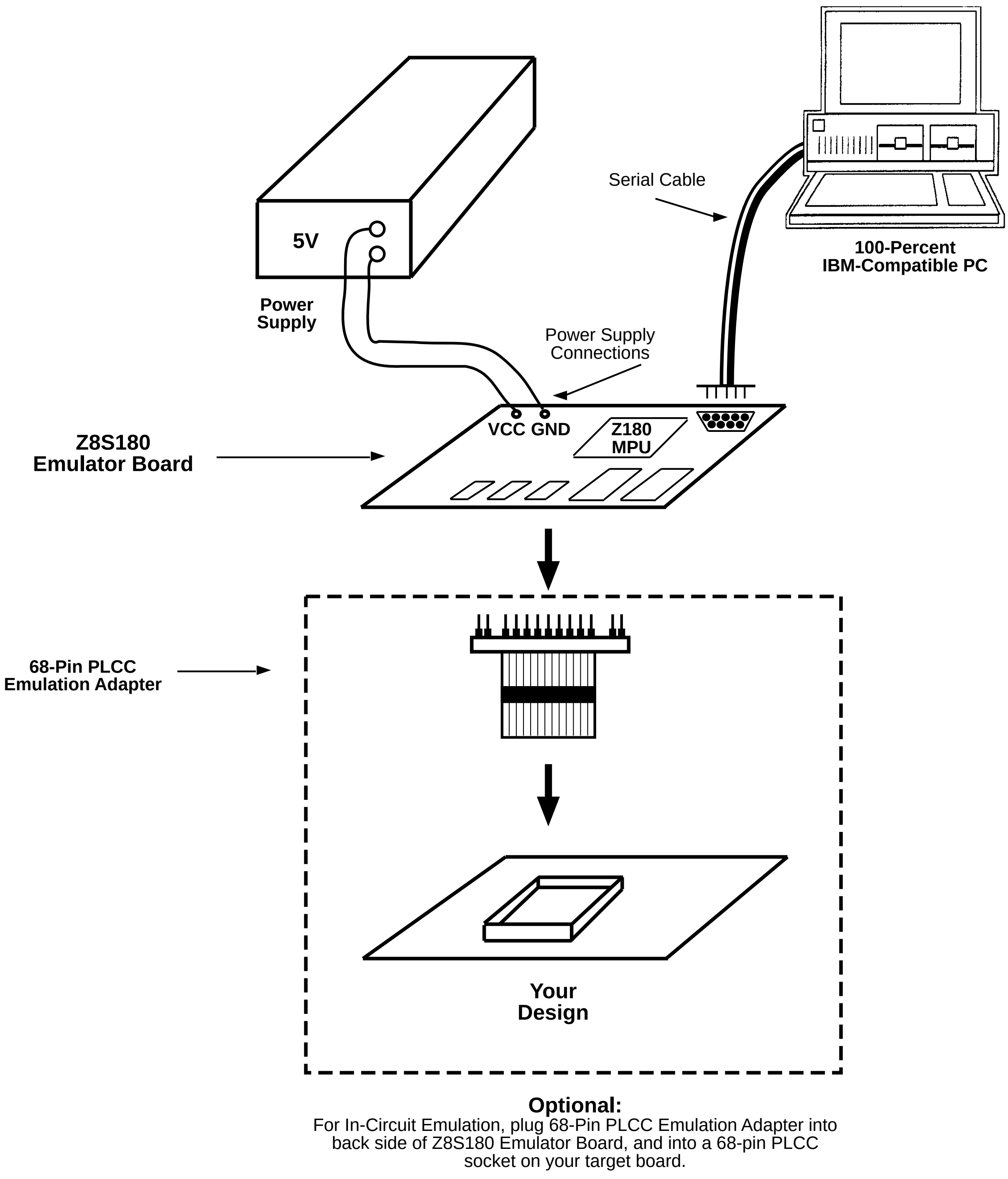


Figure 2-1. Z8S180 Emulator Board, PC, and Emulation Adapter Hook-Up Diagram

INITIAL CHECKOUT/SAMPLE SESSION

The section that follows provides an initial checkout of the hardware/software installation and setup and introduce you to some of the features of the TZ terminal emulation program.

Note: There must be RAM between addresses 00000 and 01FFF for purposes of this exercise; therefore, all three shunts must be inserted in the J5 header, unless you have already joined the emulator board to a target board that includes RAM at these addresses.

1. If necessary, power-up the PC and wait for the boot process to complete.

2. Double-click the TZ icon (if you have installed the TZ Program as an icon within Windows Program Manager).

   or

   At the DOS prompt (in the directory containing TZ.EXE, DISTEST.HEX, and TSTRST20.HEX), type "TZ COM*n*" (where *n* is the number of the COM port to which you connected the Monitor serial cable), then press <ENTER>.

   You should now see the Debug Monitor's initial screen message:
   Zilog Z80180 Monitor Version *n.m* (where *n.m* is the version number of the Debug Monitor).
   Z80180>

3. Power-up the system.
   If powering up the board from a separate power supply:
   Set the voltage to +5V.

4. Type "H" (for Help) at a Monitor command prompt to see a list of commands.

5. Type an "L". The screen should show "Enter File Name:"

   Note: Only if using the TZ emulation program only. Not available if you are using a dumb terminal or running a PC communications program other than TZ.

6. Type "DISTEST", then press <ENTER>. You should see an incrementing count. Its final value should be followed with the message "Intel hex lines—Done" and a new Monitor command prompt: "Z80180>"

   Note: Refer to Chapter 3: "Using the Debug Monitor" if you do not get the "hex done" message.

7. Type a "U". The screen should display the following message: "Disassemble Starting At (just CR = from PC):"

   Press <ENTER> to indicate that you want to disassemble from the current Program Counter value, which was set to the start of the DISTEST program as it was downloaded.

   The screen should now show a new prompt "Number of Instructions:"

8. Type “16”, then press <ENTER>.

A screen full of assembly-language instructions should appear, followed by another command-line prompt.

```
1000    00       NOP
1001    10FD     DJNZ     1000
1003    1018     DJNZ     101D
1005    20F9     JR       NZ,1000
1007    2014     JR       NZ,101D
1009    30F5     JR       NC,1000
100B    3010     JR       NC,101D
100D    08       EX       AF,AF'
100E    18F0     JR       1000
1010    180B     JR       101D
1012    28EC     JR       Z,1000
1014    2807     JR       Z,101D
1016    38E8     JR       C,1000
1018    3803     JR       C,101D
101A    013412   LD       BC,1234
101D    11DCFE   LD       DE,0FEDC
1020    215A5A   LD       HL,5A5A
1023    31A5A5   LD       SP,0A5A5
1026    09       ADD      HL,BC
1027    19       ADD      HL,DE
1028    29       ADD      HL,HL
1029    39       ADD      HL,SP
Z80180>
```

9. Press <ENTER> at the command prompt to disassemble more of the instructions in DISTEST.

Note: The file, which preceded, is not intended to be an executable program. Rather, it is the test file for the disassembly command in this Debug Monitor. DISTEST includes all the instructions that can be executed by the Z8018x family processors.

10. Type another “L” at the command prompt, then type “TSTRST20” as the file name, followed pressing <ENTER>.

11. Again you should see the final count followed by the “ Intel hex lines—Done” and a Monitor command prompt.

INITIAL CHECKOUT/SAMPLE SESSION (Continued)

12. Type “G” at the monitor command prompt. The screen should show the message “Go Starting at Address:”. Press <ENTER> to start the TSTRST20 program.

Note: Chapter describes the I/O services that the Debug Monitor provides to downloaded programs.

The first two lines displayed by TSTRST20 tests the ‘message’ service, and should appear as:
(message) Test RST 20 services (date)

A. Enter a hex value up to 4 digits (in_hl)

B. Enter a 1- to 4-digit hex value followed by pressing <ENTER>. This tests the ‘in_hl’ service. The following display tests three services called ‘out_hl’, ‘out_a’, and ‘out_char’, and should appear as follows:

16-bit value we saw (out_hl) : *XXXX*
the terminating character was (out_a) : 0D
and the ‘any digits’ CC was (out_char) : NZ
Waiting for you (test_char) :

C. Check that the value shown above as *XXXX* is the hex value you entered.

D. Type any data character on the keyboard. This tests two services called ‘test_char’ and ‘in_char’.

After you type a character, the concluding messages from TSTRST20 should appear as follows:

```
Your character was (in_char) : XX
A   F  B   C  D  E   H  L   A'  F'  B'  C'  D'  E'  H'  L'  I    IX    IY    SP   SEI
0D 20  03  B6 8E  0D  03  6A 02  F7  00  04  2E  B7  04  62 00  010F  28BF  1F00  0
024C  EF            RST     28
Z80180>
```

13. Check whether the value shown as *XX* in the preceding paragraph is the hex value of the character entered. The second through fourth lines above indicate that the TSTRST20 program ended with a Breakpoint instruction.

This completes the Initial Checkout/Sample Session.

CHAPTER 3
USING THE MONITOR PROGRAM

INTRODUCTION

This chapter begins by describing the Debug Monitor program, followed by the simple procedures necessary to start and stop the program. This chapter concludes with the full descriptions of each of the available Program Commands and RST 20H Services. This chapter is divided into the following sections:

- About the Monitor Program
- Getting Started
- Program Commands
- RST 20H Services

About the Monitor Program

The Z8S180 Emulator includes a Debug Monitor program in its EPROM chip (U3). This monitor program can be used with a character-oriented dumb terminal; however, terminal emulation facilities are available on a PC when the TZ.EXE program is used. The TZ program on the PC allows you to do the following: download a PC program, run a PC program (with or without breakpoints), display/fill memory locations, compare memory contents, display/modify registers, and read/write from/to I/O ports, including Z8S180 Registers.

Two hex files, DISTEST.HEX and TSTRST20.HEX, containing binary/absolute programs for the Z8S180 processor are provided on the Z8S180 Sample Files and Monitor Source Diskette. You can use the Zilog Z800 assembler and object utilities provided with this emulator, or third-party Z80 or Z180 software development tools to generate your own Z8S180 programs and hex files.

Debug Monitor Memory Map

The Debug Monitor occupies about 6 KB in the EPROM on the emulator, however, this EPROM is never visible in the user memory map.

The 8 KB of RAM on the emulator can be mapped to any of seven different 8 KB blocks within the first 65 KB of user memory space, or can be left out of the user memory space. The Monitor uses the last 256 bytes of this RAM for its variables, stacks, and tables. When the RAM is mapped into the user address space, the emulator hides these last 256 bytes from access by user programs or Monitor commands.

GETTING STARTED

Starting Up the TZ Program
(in DOS or at DOS Prompt in Windows)

Note: Proper installation of the monitor program assumes that the proper *hardware setup* is complete. **Before** loading the monitor program, refer to Chapter 2: "Setup and Installation" for the complete procedure for hardware setup and applying power.

1. Ensure that the PC is powered up and is functioning properly.
2. Verify that all hardware connections and settings are correct per Chapter 2 "Setup and Installation."
3. If using a PC, load the Z8S180 Sample Files and Monitor Source Diskette and copy the TZ.EXE program to the PC.
4. Enter "TZ com*n*" (where *n* is the COM serial port number [1–4] of choice); press <RETURN>.
5. Connect the emulator and power supply.
7. Power up (if power has been off) or press the RESET button on the emulator.

Starting Up the TZ Program
(TZ Icon in Windows)

Note: Proper installation of the monitor program assumes that the proper *hardware setup* is complete. **Before** loading the monitor program, refer to Chapter 2: "Setup and Installation" for the complete procedure for hardware setup, including setting up jumpers, connecting cables, and powering up the emulator.

1. Ensure that the PC is powered up and is functioning properly.
2. Verify that all hardware connections and settings are correct per Chapter 2 "Setup and Installation."
3. Double-click on the TZ icon (specifying one of the COM serial ports).
4. Connect the emulater and power supply.
5. Press the RESET button on the emulator.

Opening Screen Message

If all connections to your PC are correct, you should see the following opening message at power up or by pressing the RESET button on the board:

Zilog Z80180 Monitor Version *v.m*[1]

Z80180 >

[1] Where *v* is the current version of the debug monitor and *m* is the revision.

Exiting the TZ Program

If the TZ Program was started with DOS, enter "CTRL+C" (hold the <CONTROL> key down and enter "C", then release both). TZ will return you to the DOS prompt.

If the TZ Program was started in Windows by double-clicking the TZ icon, the window will close or the full-screen display will be replaced by the Windows environment.

TZ Program Restrictions

If the Debug Monitor is used with a dumb terminal, or with a PC running a terminal emulator other than TZ, two commands will not work as described in this chapter:

1. An "L" command (for loading a hex file) will not be followed by the "Enter File Name:" prompt.
2. An "N" command (changing the serial data rate) will either show nonsense characters, or no characters at all, after you enter the speed-selection digit.

Monitor Debug Program and Serial Baud Rates

If an N command is used to change the baud rate between the board and the PC, both the TZ Program on the PC and the Debug Monitor on the board change the rate simultaneously. Quitting, starting, or pressing RESET on the board always sets the rate used by the Monitor to 9600 bps. However, TZ can be started at any of the baud rates available by using the N command.

Type the baud rate after the COM*n* (where *n* is the COM serial port number [1–4]) argument in a DOS command, as in the following example:

TZ COM2 57600

This third argument (as in "57000" in the preceding example) can be any of the following entrees: 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, or 57600.

PC Models/Serial Baud Rates

The maximum serial rate that can be used between a PC and the emulator is dependent upon which PC model and configuration that is used. If the baud rate is too fast for the PC, characters will be lost during lengthy display sequences initiated by the Debug Monitor D, U, or F commands. Also, downloading may fail if the serial rate is too high.

PC Configuration	Maximum Baud Rate (bps)
286 or 386 (slower)	19,200
386 (faster), 486, Pentium	57,600

Note: Under Windows, you can use the Windows .PIF editor to make a new PIF file that includes such a command line, and then use a File/New command under the Program Manager to make a new icon that references the new .PIF file.

PROGRAM COMMANDS

Program Basics

1. The monitor program will prompt for all commands (A–X) with “Z80180 >”.
2. Press the <ESC> key if you make a mistake or want to return to the command prompt.
3. An audible signal is given if the monitor doesn’t recognize a command.
4. Numerical values are expected to be in hex. If a value is not entered (just <RETURN>), a zero value is assumed, except in the editing modes of the “A” and “E” commands.
5. Many commands (A, C, D, E, F, G, I, M, R, S, T, and U, for example) are repeatable from where they left off, by pressing <RETURN> at the command prompt.
6. When the first letter of the command is entered, the monitor will automatically type in the rest of the prompt.
7. To exit the program (and DOS), hold the <CONTROL> key down and enter “C”, then release both.
8. The monitor program is not case sensitive.

Table 3-1. Program Command Reference Chart

Command	Description	Page No.
A	Alter Memory	3-5
B	Set or Show Breakpoints	3-5
C	Compare Memory Data	3-6
D	Display Memory Data	3-6
E	Edit/Display I/0 Data	3-7
F	Fill Memory	3-8
G	Go To Program	3-8
H	Help	3-10
I	Input Data from I/O Address	3-10
K	Kill Breakpoint(s)	3-10
L	Load an Intel Hex File	3-11
M	Move Memory to Memory	3-12
N	Change Serial Data Rate	3-12
O	Output Byte to I/O Address	3-13
R	Display/Alter Registers	3-14
S	Step (Over Subroutine Calls)	3-15
T	Step (Into Subroutine Calls)	3-15
U	Disassemble Instructions	3-16
V	Display Version of the Program	3-16
X	EXamine the MMU	3-17

Alter Memory [A]

Use the Alter Memory command to modify bytes in memory. The display shows the address and current data. When a byte value is entered, it is written to the address shown, and the next higher or lower address and data are displayed. The monitor program does not write to the location when entering a terminating value without the hex value before it. Terminating is as follows:

<ESC>	Does not store a preceding value and returns to the command prompt.
.	Stores a preceding value if any and returns to the command prompt.
=	Stores a preceding value if any, stays at the same location, re-reads and re-displays.
RET, tab, Space, +,>	Stores a preceding value if any, and go to the next location.
-,<,^	Stores a preceding value if any, and go to previous location.

Example:

```
Z80180>Alter Memory starting at : 0100
0100 FF : 22
0101 FF : 14
0102 FF : 50^
0101 14 :
Z80180>
```

Set or Show Breakpoints [B]

A break point is a special status you can associate with a memory address in RAM, so that when you set a program running with a G command, and the execution comes to that address at the start of the instruction, it will stop running and return to the monitor command prompt. Up to eight (8) breakpoints can be set at one time. The code at each break address is replaced with the RST 28H (hex EF). This opcode will bring the control back to the monitor if execution reaches the breakpoint. The user can code RST 28H instruction right into the program, at the end of the program, and/or at any point the user needs the control to come back to the monitor. In the case of this “hard-coded” RST 28H, the monitor program simply increments the address to the following instruction before it starts execution. If a breakpoint is set at the starting address in a G command, the monitor program will set one single-steps over the first instruction, then sets all the breakpoints and goes. The monitor also checks to see whether a requested breakpoint is in fact in RAM (where it must be).

Note: Breakpoints must be set at the first byte of an instruction. (Instruction starting points can be determined by using the U command.) Breakpoints that are randomly set (or breakpoints that are set in the middle of an instruction) may cause faulty operation of the instruction.

(Example of the “Set or Show Breakpoints” command is shown on the next page.)

Set or Show Breakpoints [B] (Continued)

Example:
Z80180>Breakpoint at Address (just <ENTER> to Display All) :
No Breakpoints
Z80180>Breakpoint at Address (just <ENTER> to Display All) : 1234
Z80180>Breakpoint at Address
    (just <ENTER> to Display All) :
Breakpoints 1234
Z80180>

Compare Memory Data [C]

This command compares the contents of two areas of memory and displays any differences found. If the Monitor returns the command prompt, the two memory areas have identical contents.

Example:
```
Z80180>Move From Memory starting at : 800
                 to Memory starting at : 900
                     Number of Bytes : 100
Z80180>Alter Memory starting at : 904
0904 FF : 11
0905 FF :
0906 FF : 22
0907 FF :
Z80180>Compare Memory Data starting at : 800
             with Memory Data starting at : 900
                         Number of Bytes : 100
0804=FF : 0904=11
0806=FF : 0906=22
Z80180>
```

Display Memory [D]

When a D is entered at the command prompt, the Monitor displays the “Display Memory...” prompt and waits for you to enter a starting address and a number of bytes. The Monitor then displays the contents of that part of memory as shown in the following example. (The part on the right shows ASCII characters.)

If you <ENTER> only at the command prompt after a D command, the Monitor will display the same number of bytes again, starting at the address after the last one it displayed.

Example:

```
Z80180>Display Memory starting at : 1000
        Number of Bytes : 40
1000  CD 3A 0F E5  21   01  11  B7  28  0A  3D  F5  7E  23  B7  20   .:..!... (.=.~#.
1010  FB  F1 18  F3 CD  1C  10  E1  C9 CD  3A  0F  C5  F5  06  00   ........ ..:.....
1020  18  03 CD 60  0F  7E   23  B7  28 16   F2  22  10   E5  F5  04   ...'.~#. (..".... 
1030  21 1D 16  D6  7F  4F   0D 28   EC 7E  23  B7  20  FB  18  F6   !....O.( .~#. ...
Z80180>
```

Edit/Display I/O Data [E]

This command combines the functions of the A command and the D command, except that it applies to I/O ports rather than memory. When you key in an E command at the command prompt, the Monitor displays the "Edit/Display..." prompt and waits for you to enter an I/O address followed by an <ENTER>. Then it asks how many bytes you want to display.

If you answer this second prompt with <ENTER> only, as in the example that follows, the Monitor reads and displays the I/O address you entered in response to the first prompt, and waits to see if you want to change it, as in the A command.

If you answer the second prompt by entering a hex number of bytes followed by <ENTER>, the Monitor reads and displays the number of I/O addresses you entered, as in the D command, except that data is not displayed in ASCII form on the right side of each line.

If a hex value is entered and followed by a terminator (other than <ESC>), the Monitor writes it to the I/0 address.

The program recognizes the following terminating characters:

<ESC>	Does not store a preceding value and returns to the command prompt.
.	Stores a preceding value if any and returns to the command prompt.
=	Stores a preceding value if any, stays at the same location, re-reads and re-displays.
RET, tab, Space, +,>	Stores a preceding value if any, and go to the next location.
-,<,^	Stores a preceding value if any, and go to previous location.

Example:

```
Z80180>Edit/Display starting at I/O Register: dc
    Display How Many Bytes (just <ENTER> to Edit):
00DC 00 : 5
00DD 00 : 1F.
Z80180>Edit/Display starting at I/O Register: c0
    Display How Many Bytes (just <ENTER> to Edit): 40
00C0  78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
00D0  78 78 78 78 78 78 78 78 00 00 F8 F8 05 1F 00 5C
00E0  FF 5F FF FF FF FF FF FF C4 00 23 02 01 04 FF 78
00F0  03 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
Z80180>
```

Fill Memory [F]

When you key an F at a command prompt, the Monitors displays the three prompt messages shown in the following example, and waits for you to enter, in succession, the starting address in memory that you want to clear to a constant value, the value, and the length of the area in bytes, each terminated with <ENTER>. As at any Monitor prompt, if you change your mind above filling memory, you can press the <ESC> key to return directly to the command prompt.

After you <ENTER> the third time, the Monitor will perform the requested filling operation, and return to the command prompt. The “number of bytes” can be up to 65 KB.

On the Z8S180 Emulator, you can clear all of memory with this command, because the emulator does not allow the Monitor’s internal states, variables, and tables to be accessed by Monitor commands.

Example:

```
Z80180>Fill Memory starting at : 1108
                   with Data : 00
             Number of Bytes : 7
Z80180>Display Memory starting at : 1100
             Number of Bytes : 20
1100  22 14 50 FF FF FE FF FF  00 00 00 00 00 00 00 FF   “.P..... ........
1110  FF FF FF FF FF FF FF FF  FF FF FF FF FD FF FF FF   ........ ........
Z80180>
```

Go to Program [G]

When a G command is entered, the Monitor displays the “Go starting at...” prompt and waits for a hex address entry and an <ENTER>. The entered address is taken as a starting address for execution. The code typically has been downloaded from a Personal Computer host using an L command. If hex address is not entered, the program simply starts execution from the current Program Counter (PC) value, which can be from previous execution or the starting address from a preceding download.

In the example that follows, the user downloaded a program named “JONESTST.HEX” and then started it with a G command with no address, since JONESTST.HEX included a valid starting address. The program then displays several messages and begins waiting for the user to type another G.

Before the Monitor executes the user program, it proceeds as follows:

1. Checks to see if the opcode at the starting address is a hard-coded RST 28H. If so, it advances the starting address over the RST 28H and proceeds to Step 3.

2. If not, it checks for a breakpoint at the starting address. If so, it invokes an implicit T command, to single step over the first instruction, returning to Step 3 when this has been completed.

3. For each breakpoint, it saves the opcode at that address in a RAM table and substitutes the opcode RST 28H (EF).

4. It stacks the starting address and restores all the user register values that were saved in the last RESET, breakpoint, hard-coded RST 28, or NMI, and may have modified since then.

5. It enables or disables interrupts per the saved EI value.

6. It uses a RET instruction to begin execution at the starting address stacked in Step 4.

After a G command, there are several ways that control can return to the Monitor:

- If execution comes to an RST 28 instruction, either a breakpoint or hard-coded one
- If execution comes to an illegal instruction
- Or when you press the NMI or RESET push button

In all of these cases except RESET, the Monitor saves the program counter. In all cases, it saves the register contents, and restores the opcodes of any set breakpoints. Then, for NMI or an illegal instruction, it displays a message why execution stopped, and in any case displays the register contents, program counter, and the instruction in hexadecimal and disassembled formats.

Example:

```
Z80180>
File name: jonestst
78 Intel hex lines - Done
Z80180>Go starting at Address :
Z80180 ASCI Tx hacking: v.1.0
 Please set jumpers as follows:
ASCI0 RXA  to ESCC TXD        (Eval bd. J12-3  to J10-2)
ASCI0 TXA  to ESCC RXD        (Eval bd. J12-2  to J10-3)
ASCI0 /DCD to PIA12           (Eval bd. J12-8  to J12-20
                                    and J14-12 to J14-13)
ASCI0 /CTS to ESCC /RTS       (Eval bd. J12-5  to J10-4)
ASCI0 /RTS to ESCC /CTS       (Eval bd. J12-4  to J10-5)
Press 'G' on keyboard when ready...
```

Help [H]
Entering “H” displays the command set available from the monitor program.

A – Alter Memory
B – Set or Show Breakpoints
C – Compare Memory Data
D – Display Memory Data
E – Edit/Display I/O Data
F – Fill Memory
G – Go to Program
H – Help
I – Input Data from I/O Address
K – Kill Breakpoint(s)
L – Loading an Intel Hex Code
M – Move Memory to Memory
N – Change Serial Data Rate
O – Output Byte to I/O Address
R – Display/Alter Registers
S – Step (Over Subroutine CALLs)
T – Step (Into Subroutine CALLs)
U – Disassemble Instructions
V – Display Version of the Monitor
X – EXamine the MMU

Input Data from I/O Address [I]
When you key in an “I” at the command prompt, the Monitor displays the “Input from . . .” prompt and waits for an I/O address entry followed by <ENTER>. That address is read and displayed before returning to the command prompt. (See the E command description about 16-bit I/O addresses.)

Example:
Z80180>Input From I/O Address : 20
D020 AF
Z80180>

Kill Breakpoint(s) [K]
When a K is entered at the command prompt, the Monitor displays the “Kill Breakpoint...” prompt and waits for a hex address entry before an <ENTER>. If a hex address is not supplied, the Monitor simply clears any and all breakpoints that may be set, and returns to the command prompt. If a hex address is supplied, and the address is a breakpoint, the Monitor clears that breakpoint and returns to the command prompt. If you type an address that is not a breakpoint, the Monitor displays an error message before returning to the command prompt.

Example:
Z80180>Breakpoint at Address (just <ENTER> to Display All) :
Breakpoints 0717 0846 134E
Z80180>Kill Breakpoint at Address (just <ENTER> for All) : 846
Z80180>Breakpoint at Address (just <ENTER> to Display All) :
Breakpoints 0717 134E
Z80180>Kill Breakpoint at Address (just <ENTER> for All) :
Z80180>Breakpoint at Address (just <ENTER> to Display All) :
No Breakpoints
Z80180>

Loading an Intel Hex File [L]

When an L is entered at a command prompt, the Monitor displays the courtesy message “Load Hex File”. Then it sends a special sequence of control characters, which a current version of Zilog’s TZ terminator emulator program will recognize as a download request. TZ itself then displays the “File name” prompt and waits for you to enter a file name. Current versions of TZ will assume a “.HEX” extension.

If you happen to enter the file name wrong, TZ will display an error message. To recover from this state, press the <ESC> key or the NMI button on the board, to cancel the download. Then enter L again.

If the hex file is found, TZ then displays an increasing count of lines sent. When it has sent the whole file, TZ goes back to its normal mode of terminal emulation. The Monitor then shows how the download was executed. A complete and correct download is indicated by an “Intel hex lines - Done” message.

Error messages can include the following:
Bad Record Termination
Unknown Record Type
Bad Record Format
Checksum Error

If you get one of these messages, first check the hex file you tried to download to see that it conforms to the standard format for Intel-compatible hex files. If the file conforms, try again. If the problem persists, you might want to try a slower serial speed (using the N command), or a different PC host, or a different serial cable, or a different board, or a different hex file.

Example:
Z80180>Load Hex File
File name: joneststst
78 Intel hex lines - Done
Z80180>

Move Memory to Memory [M]

This command moves a specified number of bytes from one address to another. In order to handle overlapping “from” and “to” areas correctly, the monitor moves data differently, depending on how the addresses compare. If the “from” address is greater than the “to” address, the monitor moves the data between the starting address and higher addresses thereafter. If the “from” address is less than the “to” address, the monitor begins moving between the highest addresses implied by the number of bytes and lower address thereafter until it gets down to the starting address entered.

You can use this command anywhere in memory because the monitor’s internal states, variables, and tables cannot be accessed by monitor commands.

If you change your mind about copying data from one area of memory to another, you can <ESC> to return to command prompt. Otherwise, <ENTER> after the number of bytes moves the data as requested.

Example:

Z80180>Move From Memory starting at : 800
to Memory starting at : 900 Number of Bytes : 100
Z80180>Alter Memory starting at : 904 0904 FF : 11
0905 FF :
0906 FF : 22
0907 FF :
Z80180>Compare Memory Data starting at : 800 with Memory Data starting at : 900
Number of Bytes : 100
0804=FF : 0904=11
0806=FF : 0906=22
Z80180>

Change Serial Data Rate [N]

When you enter an N at a command prompt, the Monitor displays the “Enter 0-9...” prompt and waits for you to key a single digit. <ENTER> is not needed thereafter.

This command should be used only if you are using a current version of Zilog’s TZ terminal emulator program for communications on your host PC. If so, when you key the desired digit, the Monitor pauses for a moment and then returns to the command prompt, at the new rate. You may see a difference in performance in commands like L and D.

The baud rate remains as set until another N command, or until you press RESET, which returns the platform’s rate to 9600. Before pressing RESET when the rate is other than 9600, you may want to perform CTRL+C on the keyboard to terminate your current copy of TZ, and then enter TZ COMn again to start a new one at 9600.

The main use of this command is to speed up a subsequent L command that downloads a long file. In the example, after the download the user returns the rate to 9600, in which case RESET is pressed.

Example:

```
Z80180>Enter 0 for     1200 Bits/Second
                       1 for   2400
                       2 for   4800
                       3 for   9600
                       4 for  14400
                       5 for  19600
                       6 for  28800
                       7 for  38400
         Enter 8 for  57600 Bits/Second : 8
Z80180>Load Hex File
File name: jonestst
78 Intel hex lines - Done
Z80180>Enter 0 for     1200 Bits/Second
                       1 for   2400
                       2 for   4800
                       3 for   9600
                       4 for  14400
                       5 for  19600
                       6 for  28800
                       7 for  38400
         Enter 8 for  57600 Bits/Second : 3
Z80180>
```

Output Byte to I/O Address [O]

If an O is entered at a command prompt, the Monitor displays the two prompts shown in the following example and waits for a hex value entry followed by <ENTER> for each. The first value is the I/O address and the second value is the data to be written to it. With the second <ENTER>, the Monitor writes that data to the specified I/O address, and returns to the command prompt.

See the E command description about 16-bit I/O addresses.

Example:

```
Z80180>Output to I/O Address : dd
                          Data: f3
Z80180>
```

Display/Alter Registers [R]

If an R is entered at a command prompt, the monitor displays the “Display/Alter Register” prompt and waits for entry of the name of a register to be modified, then <ENTER>.

If the name of a register is not entered, the Monitor simply displays the contents of all of the user registers, and displays the PC and the current instruction in hexadecimal and disassembled form.

If a register name is entered, it should be one of the following: A, F, B, C, D, E, H, L, A’, F’, B’, C’, D’, E’, H’, L’, I, IX, IY, SP, EI, or PC. The Monitor then displays the current contents of that register and waits for modification.

The following terminating characters and entry procedures apply:

<ESC>	Does not store a preceding value and returns to the command prompt.
.	Stores a preceding value if any and returns to the command prompt.
=	Stores a preceding value if any, stays at the same location, re-reads and re-displays.
RET, tab, Space, +,>	Stores a preceding value if any, and go to the next location.
-,<,^	Stores a preceding value if any, and go to previous location.

Example:

```
Z80180>Display/Alter Register (just <ENTER> Displays All) : b
B       03 : 1
C       11 : 2
D       12 : 3
E       E1 : 4
H       44 :
Z80180>Display/Alter Register (just <ENTER> Displays All) :
A  F  B  C  D  E  H  L  A' F' B' C' D' E' H' L' I  IX   IY   SP   EI
02 55 01 02 03 04 44 55 00 00 00 00 00 00 00 00 00 012B 0000 1EF6 0
18DA 28F9      JR    Z,18D5
Z80180>
```

S and T Commands

Step (Over Subrouting Calls) [S]
Step (Into Subroutine Calls) [T]

If S or T is entered at the command prompt, the monitor displays the "Step How Many Instructions" prompt and waits for an optional hex value entry, followed by <ENTER>. If a value is not entered, one instruction is assumed.

After stepping over each instruction, the monitor displays the register values, the Program Counter (PC), and the next instruction in both hexadecimal and disassembled format. After the last of the specified number of instructions, the Monitor returns to the command prompt. If you just <ENTER> at this prompt, the Monitor "steps" the same number of instructions again.

The instructions to be stepped through must be in RAM. The Monitor accomplishes stepping by placing the opcode EF (RST 28) after each instruction, and/or for instructions that can transfer control to other than the next instruction, by placing the EF opcode at the destination. The Monitor then restores the registers and transfers control to the single instruction, being assured of getting control right back because of the RST 28s. Simple operations like unconditional JR, JP, and RET are handled by simply updating the user PC, without storing any RST 28s.

CALL and RST instructions are handled differently for each of the S and T commands. S implies "stepping over" subroutines: In this case the Monitor simply stores an EF opcode after the instruction, so that it will get control back after the subroutine has completed execution and returned. T implies "stepping into" subroutines: In this case the monitor stores an EF at the start of the subroutine.

In this environment, only RST 8, 10, 18, 30, and 38 instructions can be "stepped into" with a T command. This is because RST 0 transfers control to the start of the Monitor like a RESET. RST 20 is used for services provided by the Monitor and RST 28 is used for Breakpoints.

A "hard-coded" RST 28 is treated as a No-Op while stepping. The Monitor simply advances the PC to the next instruction. RST 28s are not placed at Breakpoints while stepping.

If you step into a HALT or SLP instruction, and no device interrupts occur to escape this state, you will have to press the NMI or RESET to reenter the Monitor.

Example:

```
Z80180>Display/Alter Register (just <ENTER> Displays All) :
A  F  B  C  D  E  H  L  A' F' B' C' D' E' H' L' I  IX   IY   SP   EI
21 00 1F E9 00 00 B0 B0 00 00 00 00 00 00 00 00 00 012B 0000 1F00 0
0103  7E        LD     A,  (HL)
A  F  B  C  D  E  H  L  A' F' B' C' D' E' H' L' I  IX   IY   SP   EI
ED 00 1F E9 00 00 A5 7D 00 00 00 00 00 00 00 00 00 012B 0000 1F00 0
0104  23          INC   HL
A  F  B  C  D  E  H  L  A' F' B' C' D' E' H' L' I  IX   IY   SP   EI
ED 00 1F E9 00 00 A5 7E 00 00 00 00 00 00 00 00 00 012B 0000 1F00 0
0105  FEC0      CP    A, 0C0
Z80180>
```

Disassemble Instructions [U]

This command disassembles a specified number of instructions from a specified starting address. The next instruction is always disassembled whenever the registers are shown the following: R command with no operand, breakpoint, step, and NMI. The maximum value for this command is in the range hex 15-17 (decimal 21-23) since more that this will not fit on the screen. After disassembling the indicated number of instructions, the monitor returns to the command prompt. If you respond with a <RETURN> only, the monitor disassembles the same number of instructions again, starting from where it left off. This facility is useful for “scanning through” a program looking for a particular type of code.

Example:

```
Z80180>Disassemble Starting at (just <ENTER> = From PC): 200
                                   Number of Instructions : 16
0200   214D02   LD     HL,024D
0203   CD8203   CALL   0382
0206   CD7E03   CALL   037E
0209   F5       PUSH   AF
020A   E5       PUSH   HL
020B   21AB02   LD     HL,02AB
020E   CD8203   CALL   0382
0211   6F       LD     L,A
0212   E3       EX     (SP),HL
0213   CD7A03   CALL   037A
0216   21CD02   LD     HL,02CD
0219   CD8203   CALL   0382
021C   E1       POP    HL
021D   7D       LD     A,L
021E   CD7603   CALL   0376
0221   21F902   LD     HL,02F9
0224   CD8203   CALL   0382
0227   F1       POP    AF
0228   5F       LD     E,A
0229   2805     JR     Z,0230
022B   3E4E     LD     A,4E
022D   CD6A03   CALL   036A
Z80180>
```

Display Version of the Monitor [V]

Entering a "V" displays the software version number of the Debug Monitor program.

Example:

```
Z80180 > Version
Zilog Z80180  Monitor Version 1.4

Z80180>
```

Examine the MMU [X]

This command can be used to control the mapping from logical to physical memory. When you enter an "X" at a command prompt, the monitor program reads the values in the three MMU registers (CBAR, BBR, and CBR) and translates their values into to the "current MMU map" that it displays on the next five lines. Next, it displays two lines describing the contents of the CBAR register, displays its contents, and allows you to enter a new value for CBAR. After that, if the CBAR value enables/uses the Bank area, the monitor program similarly describes BBR, displays its value, and allows you to enter a new value for it. Finally, it describes CBR, displays its value, and allows you to enter a new value.

If a new value is not entered for any of the three registers, the monitor program will return to the command prompt after you <ENTER> for CBR. Otherwise it will write the new value(s) at this point and will attempt to go back and re-display the new MMU map as before.

If a combination of register values is selected that does not let the monitor program keep running, the new MMU map will not be seen. Trial and error is the easiest route to working with the MMU using this command.

Example:

```
Z80180>eXamine the MMU

Current MMU Map:
Area          Logical Address Range       Physical Address Range
Common 0      (Not Used)
Bank          0000-EFFF                   00000-0EFFF
Common 1      F000-FFFF                   0F000-0FFFF
CBAR MS 4 = Boundary between Bank and Common 1 Areas in Logical Address Space
CBAR LS 4 = Boundary between Common 0 and Bank Areas in Logical Address Space
CBAR F0 :
BBR contains adder to Bank Area Logical Address to get Physical Address
BBR  00 :
CBR contains adder to Common 1 Area Logical Address to get Physical Address
CBR  00 :
Z80180>
```

Monitor Mode

Refer to Chapter 4: Reference, "Emulation Modes of Operation" for more information about Monitor and User Modes, including how to "debug" the Debug Monitor.

RST 20H Services

The monitor program offers a number of console I/O services using the RST 20 instruction. These services have advantages over direct I/O to the ASCI or ESCC registers—they implicitly use the console port that the user has selected.

Prior to the RST 20h, the user should set the B register to identify the services required, from the following list. No registers are changed except as indicated.

The TSTRST20. HEX program, which is on the Z8S180 Sample Files and Monitor Source Diskette, can be used to test the RST 20 instruction.

(B)	Name	Service Performed
0	out_char	Waits, if necessary, for monitor console device to be ready for output, then outputs the ASCII character in A. A is not changed.
1	in_char	Waits, if necessary, for user to press a key on the monitor console device; returns its ASCII value in A.
2	test_char	Returns NZ condition code if user has pressed a key on the monitor console device, else returns Z. A is not changed.
3	out_a	Waits, if necessary, for monitor console device to be ready for output, then outputs the 8-bit hexadecimal value in A. A is changed.
4	out_hl	Waits, if necessary, for monitor console device to be ready for outputs, then outputs the 16-bit value in HL. A is changed.
5	in_hl	Waits for user to enter a hexadecimal value on the monitor console device, and returns it in HL. Terminating characters are as in monitor hex entry; the terminating character is returned in A.
6	message	Outputs a null-terminated string—pointed to by HL—on the monitor console device. HL is advanced over the terminating null. A is not changed.

CHAPTER 4
REFERENCE

REFERENCE

About RESET

The following occurs when you apply power to the board or press RESET:

1. The Z8S180 sets the user Stack Pointer (SP) to 0000, so that the first stack locations used by a subsequent PUSH or CALL instruction would be 0FFFE and 0FFFF. When the Monitor has been given a G(o), S(tep), or T command, it checks whether the location that is one less than the current user Stack Pointer acts like RAM, because it needs a working user stack in order to transfer control to a user program. If the current SP, minus one, does not point at RAM, the Monitor searches (top down) the first 65 KB of memory space, by steps of 256 bytes, until it finds RAM, then sets the starting user SP to the address xx00, one higher than the end of RAM. (If the only RAM is the 8 KB on the Z8S180 board, and it starts at 0000, the first G, S, or T command sets SP to 1F00.)

2. The Monitor sets its user Program Counter (PC) value to 0100. (This value is changed automatically if a hex file containing a starting address is downloaded.)

3. The Monitor saves all other registers in the 180. Pressing RESET to get control back from a runaway program may help determine what is happening from the register values and data on the stack.

4. Clears any previously set breakpoints.

5. Initializes the CPU registers to use the OSC frequency directly (not divided by two), to disable refresh, and to use the appropriate number of Wait States for memory and no Wait States for I/O.

6. Sets up ASCI0 and ASCI1 for 9600 bit/second operation.

7. Checks the RAM location in which it stores whether ASCI0 or ASCI1 is used for its user interface. If this location does not contain either of two particular ASCI values, the Monitor waits until it sees an ASCII <ENTER > from ASCI0 or ASCI1, and sets the location accordingly.

8. Sends the initial "version" message and first command prompt through the selected ASCI to the host.

Note: You may find that the ASCI selection is sometimes preserved when you turn the board's power supply off, so that when you turn the power supply back on, the Monitor immediately shows its version message without waiting for you to press <ENTER>. Enough voltage leaks through the serial interface chip—from the TxD and RTS lines from the host—to maintain the contents of the RAM even though the power supply is off. If you need to change which ASCI is used for the Monitor user interface, turn off both the power supply and the PC, change the J7 jumpers, and turn power back on.

Emulation Modes of Operation

The Altera FPGA on the board has two major modes of operation:

- Monitor Mode
- User Mode

The MREQ signal is gated and controlled by the Altera as described below. This introduces about 6 ns worst-case delay in MREQ, which should not perturb user timing. All pins of the Z8S180 are direct connected to pins on the emulation headers except MREQ, XTAL1, XTAL2, RESET, and NMI.

Monitor Mode

The Altera FPGA enters Monitor Mode on RESET and when an M1 cycle occurs in User Mode at any of the addresses 00000 (RESET, Trap, RST 0), 00020 (RST 20, provides console I/O services to user programs), 00028 (RST 28, used for breakpoints and single-step), and 00066 (NMI). In Monitor Mode, the on-board ROM is mapped as addresses 0000-7FFF (four repetitions for an 8 KB ROM chip), and the RAM is mapped as 8000-FFFF (4 repetitions for an 8K RAM chip).

The monitor accesses memory on the user's target board (if any) by immediately preceding the access with a write to an address in the range 0000-1FFF, which makes the Altera FPGA treat the next non-M1 cycle access as if in the User Mode, or a write to an address in the range 2000-3FFF, which makes the Altera FPGA treat the next two non-M1 accesses as if in User Mode. Except after a write to a ROM address 0000-3FFF, the Altera FPGA blocks the MREQ signal from propagating to the user's target board while in Monitor Mode. This avoids tri-state conflicts on the data bus (between the ROM and RAM on this board) and memory-space resources on the user's target board.

"Debugging" the Debug Monitor State

After RESET and in normal operation, the Monitor "looks at" memory the way that a downloaded program would see it, that is, in User State. However, in certain situations some of the commands and functions of the Monitor will temporarily display memory as the Debug Monitor itself sees it, that is, in Monitor State. The Monitor has to operate this way, in order to operate correctly if you do such things as pressing the NMI button while a downloaded program has transferred control into the Monitor for one of the RST 20 services (refer to Chapter 3). This would happen most often for one of the input services, such as "in_a" or "in_hl".

The simplest way to enter this state is to press NMI while the Monitor is already running. In this state, the following commands and functions access memory in Monitor Mode rather than User Mode:

- Display (D) Command
- Go (G) Command without an address
- Disassembly (U) command (the next instruction in NMI, and R command with no operand)

Any of the following cancels this state, so that all Monitor functions operate normally (User state):

- Pressing the RESET button
- G command with starting address specified
- L command
- Entering a new Program Counter (PC) value via the R command
- Completion of an RST 20 service

User Mode

As part of a G, S, or T command, the monitor signals the Altera FPGA to enter User Mode by writing to an address in the range 4000-7FFF. The Altera FPGA then enters this mode at the next non M1 cycle, that is, as the starting address is beginning to be fetched from the stack by a subsequent RET instruction in the monitor.

In User Mode, the Altera FPGA never selects/enables the on-board ROM. It may, however, select/ enable the on-board RAM as controlled by J5:

Table 4-1. User Mode Selection of On-Board RAM—J5 Settings

J5 Connections			
5–6	**3–4**	**2–1**	**RAM Location**
No	No	No	E000–FEFF
No	No	Yes	C000–DEFF
No	Yes	No	A000–BEFF
No	Yes	Yes	8000–9EFF
Yes	No	No	6000–7EFF
Yes	No	Yes	No RAM
Yes	Yes	No	2000–3EFF
Yes	Yes	Yes	0000–1EFF

Regardless of J5 jumpering, the Altera FPGA never enables the on-board RAM for the last 256 bytes of the RAM, this region being private to the monitor program. The Altera FPGA enables the MREQ signal into the user's target board whenever it is not selecting/enabling the on-board RAM.

The Altera FPGA leaves User Mode upon seeing an M1 access to location 00000 (RESET), 00020, 00028 (Breakpoint), or 00066 (NMI).

NMI and RESET

These two push buttons on the board always affect the Z8S180, as do the RESET and NMI signals from the user's board via the Z8S180 Emulation Pod. Inserting the J8 and J9 jumpers make the RESET and NMI buttons on the board drive RESET and NMI low, respectively, to the user's board.

Z8S180 DEVELOPMENT BOARD SCHEMATIC

Board schematic not available at this site.

APPENDIX A

ACCESSING THE ZBBS/INTERNET

ZILOG 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 1-408-370-8024. The ZBBS supports speeds up to 28.8K Baud with connections 8-N-1. (We recommend that you use an ANSI/BBS terminal emulation setup.)

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

ZILOG ON THE INTERNET

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

http://www.zilog.com

APPENDIX B

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 ______________________________

Company ______________________________

Address ______________________________

City/State/ZIP ______________________________

Country ______________________________

Telephone ______________________________

Fax Number ______________________________

E-Mail Address ______________________________

Product Information

ICEBOX Serial # or Board Fab #/Rev. #______________

Software Version ______________________________

User's Manual Number (on back cover) ______________

Host Computer Description/Type ______________

____

Return Information

Zilog, Inc.
Support Products Director, D2–3
210 E. Hacienda Ave.
Campbell, CA 95008-6600
Fax Number: 408-370-8056

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.