Data General

Customer Documentation

Using the AViiON® System Control Monitor (SCM)

AViiON®
PRODUCT LINE

Using the AViiON®
System Control Monitor (SCM)

014–001802–04

Ordering No. 014–001802

Printed in the United States of America
Rev. 04, March 1991

Notice

Using the AViiON® System Control Monitor (SCM)
014-001802-04

Revision History:

Original Release – April 1989
First Revision – June 1989
Second Revision – September 1989
Third Revision – February 1990
Fourth Revision – March 1991

A vertical bar in the margin of a page indicates substantive technical change from the previous revision.

Preface

This manual describes the AViiON® System Control Monitor (SCM) and how to use SCM commands and menus to boot media, control program flow, set operating parameters, and debug programs on the AViiON RISC–based product line.

System programmers will use the SCM to develop, control, or debug programs. System administrators or operators will use the SCM to set or change system configuration parameters, or to control system operation when the UNIX® kernel is not running. *Anyone* using an AViiON computer system may need to use the SCM to reboot if their operating system or stand–alone program halts.

Although this manual is designed for readers with some technical background, it does not require specific knowledge about AViiON computers or about a particular operating system. To use SCM commands and configuration menus to manage your system and respond to system problems, you need to be familiar with general system operation procedures and with your system's installed devices. To use program debugging commands or menus to modify certain system environment parameters, you need enough background in assembly–level programming to understand the effect of your actions. There are no prerequisites to use the SCM to reset or to reboot your system.

Using this Manual

This manual provides a reference of the SCM functions. Where possible, we make it clear when text is directed toward particular readers, such as system programmers or system administrators and operators, and when text relates to specific models of AViiON hardware.

This manual presents information as follows:

Chapter 1 Overview of the System Control Monitor

This chapter describes the SCM, explains how and when you enter and exit from the SCM, outlines its uses, describes the command interpreter prompt, and suggests which sections of this manual might be useful for each listed function.

Chapter 2 Using SCM Menus

This chapter describes each menu item in the SCM menu interface. It shows the overall menu structure, presents default values and valid examples for each menu option, and explains how to enter and exit from each menu.

Chapter 3 Using SCM Commands

This chapter describes how to use the SCM command interpreter, including features and restrictions such as case sensitivity, radix, and screen edit control functions.

Chapter 3 contains command reference pages with the following information for each SCM command:

- Command name and minimal mnemonic
- Functional description
- Arguments and their definitions
- Related Commands
- Related Messages
- Examples

Appendix A Specifying a Boot Path

This appendix explains conventions for specifying boot devices and lists default device parameters currently supported by the SCM and by the DG/UX™ operating system.

Appendix B System Programming and Debugging Tools

This appendix lists SCM system calls and subroutines and briefly describes how to implement them. It also describes the functions of the Environment Control Word (ECW) and its use for diagnostic program development.

Related Documents

You received a comprehensive set of documents with your computer system. The manuals listed below are a subset of the complete documentation set for your AViiON computer; the manuals in this section are those books specifically mentioned within the text of this manual. Refer to the document *Read This First* (069–000519) for a complete reference of related documents.

Technical Notice for AViiON® 100 and 200 Series Stations: Programming System Control and I/O Registers (014–001880), used with *AViiON® 300 and 400 Series Stations: Programming System Control and I/O Registers* (014–001800)

Describes the system board architecture and how to program the system board, including the monochrome graphics subsystem, keyboard interface, serial and parallel interfaces, LAN interface, and SCSI.

AViiON® 300 and 400 Series Stations: Programming System Control and I/O Registers (014–001800)

Describes the system board architecture and how to program the system board, including the monochrome and color graphics subsystems, keyboard interface, serial and parallel interfaces, LAN interface, and SCSI.

Technical Notice for AViiON® 3000 and 4000 Series Computers: Programming System Control and I/O Registers (014–001878), used with *AViiON® 300 and 400 Series Stations: Programming System Control and I/O Registers* (014–001800)

Describes the system board architecture and how to program the system board, including the serial and parallel interfaces, LAN interface, and SCSI.

AViiON® 5000 and 6000 Series Systems: Programming System Control and I/O Registers (014–001805)

Describes the system board architecture and how to program the system board, including the serial and parallel interfaces, the VMEbus, and the associated I/O.

MC88100 RISC Microprocessor User's Manual (014–001809)

Describes the Motorola 88100 Central Processing Unit (CPU), including the registers, addressing modes, timing, and assembly–language instruction set.

MC88200 Cache/Memory Management Unit User's Manual (014–001808)

Describes the Motorola 88200 Cache/Memory Management Unit (CMMU), including the CMMU registers, the cache and cache coherency, memory management and user/supervisor space, the Processor bus (Pbus) and the Memory bus (Mbus).

Writing a Device Driver for the DG/UX™ System (093–701053)

Describes how to write your own device driver for a DG/UX system running on an AViiON computer. Under the AViiON architecture, drivers must be written to address either a specific device or an adapter that manages secondary bus access to specific devices. This manual address both types of driver.

Reader, Please Note:

We use the following conventions in this manual:

Within text, "press New Line" means that you should press the New Line key on a Data General DASHER® model keyboard, or that you should press the equivalent key on an industry standard IBM PC AT®–compatible keyboard, usually marked Enter, Return, or with a standard symbol like the following: ⏎.

A *multiuser server* refers to any AViiON hardware used to provide services to users via terminal lines and/or a local area network. A *workstation* refers to any single–user AViiON model that provides graphics computing either as a stand–alone station or as a network client to a server system.

A *system console* refers to the keyboard and display device that receives power–up diagnostic test messages and from which you bring up your operating system. In most cases, the system console communicates directly with the computer's system board. A workstation's system console usually consists of the graphics monitor and graphics keyboard.

We use the following for formats of command lines and menu entries:

This typeface for commands.

This typeface for required variables and parameters.

[This typeface with brackets for optional variables and parameters]. Do not type the brackets around optional variables. They only set off what's optional.

In examples we use

This typeface to show your entry.
`This typeface to show system queries and responses.`

Additionally, we use these symbols:

Symbol	**Means**
↵	Press the New Line key on a DASHER keyboard; *or*, press the Enter or Return key on an IBM PC AT–compatible keyboard.
<CR>	Press the Carriage Return key on a DASHER keyboard.
<Esc>	Press the Escape key on an IBM PC AT–compatible keyboard.
<Ctrl–x>	Press the Ctrl key and lowercase x (where x is any key) simultaneously. Do not type the brackets or the hyphen.
SCM>	The default System Control Monitor (SCM) prompt on single processor systems.
Jp#n/SCM>	The default System Control Monitor (SCM) prompt on multiple processor systems, where n is the number of the attached processor.

Contacting Data General

Data General wants to assist you in any way it can to help you use its products. Please feel free to contact the company as outlined below.

Manuals

If you require additional manuals, please use the enclosed TIPS order form (United States only) or contact your local Data General sales representative.

Telephone Assistance

If you are unable to solve a problem using any manual you received with your system, and you are within the United States or Canada, contact the Data General Service Center by calling 1–800–DG–HELPS for toll–free telephone support. The center will put you in touch with a member of Data General's telephone assistance staff who can answer your questions.

Free telephone assistance is available with your hardware warranty and with most Data General software service options. Lines are open from 8:30 a.m. to 8:30 p.m., Eastern Time, Monday through Friday.

For telephone assistance outside the United States or Canada, ask your Data General sales representative for the appropriate telephone number.

Joining Our Users Group

Please consider joining the largest independent organization of Data General users, the North American Data General Users Group (NADGUG). In addition to making valuable contacts, members receive FOCUS monthly magazine, a conference discount, access to the Software Library and Electronic Bulletin Board, an annual Member Directory, Regional and Special Interest Groups, and much more. For more information about membership in the North American Data General Users Group, call 1–800–877–4787 or 1–512–345–5316.

End of Preface

Contents

Chapter 1 Overview of the System Control Monitor

Chapter 2 Using SCM Menus

Chapter 3 Using SCM Commands

Appendix A Specifying a Boot Path

Appendix B System Programming and Debugging Tools

Tables

Table

Figures

Figure

Chapter 1
Overview of the System Control Monitor

The System Control Monitor (SCM) is your interface to AViiON® RISC–based computer hardware. The SCM is a firmware monitor program that tests and manages the system at powerup and maintains control until the operating system or other system software takes over. The SCM resumes control whenever your system software halts.

The SCM user interface consists of a command interpreter and several interactive menus. You can use SCM commands and menu items to control program flow, view or change system configuration parameters, debug programs, and boot software. The last part of this chapter describes where you will find related information for each of the tasks within this manual. The next section explains how and when you enter the SCM.

Getting to the SCM Prompt

You know you are in the SCM when you see its command interpreter prompt. You see the SCM prompt whenever all processors in your system are halted. In single–processor systems, the default prompt appears as follows:

```
SCM>
```

In multiprocessor systems, the default prompt displays the number of the attached job processor (Jp#n, where n is the number of the attached job processor):

```
Jp#n/SCM>
```

On many AViiON computers, you can change the text of your default SCM prompt. On multiprocessor computers, you can change the attached processor. For information about changing the default prompt text with the **PROMPT** command or changing the attached job processor with the **ATTACH** command, refer to the reference pages for these commands in Chapter 3.

During normal operation, you access the SCM only when system software encounters a problem it cannot handle while running. You can use the SCM to load, start, modify, control, or halt programs; but the SCM does not run in conjunction with system software.

Your system enters the SCM and displays the its command interpreter prompt on the system console screen under the following circumstances:

- The automatic boot sequence fails or is interrupted
- A user command halts the operating system
- A command break keyboard sequence halts active processors
- A user presses a hardware reset or power switch
- Operating system software encounters an unsupported program breakpoint or interrupt

The following sections describe each of these instances.

Failing the Boot Sequence

The SCM controls power–up testing and then brings up your system software through an automatic boot sequence. You enter the SCM user interface if a diagnostic test fails, or if the automatic boot sequence does not complete successfully. Refer to the manual that describes how to set up and start your computer for instructions about power–up failure. The section "Change Boot Parameters Menu" in Chapter 2 of this manual describes how you can change your automatic boot sequence.

You can intentionally enter the SCM at powerup by executing the SCM interrupt sequence <**Ctrl–C**> before your operating system software boots. If you do not have an auto–repeat keyboard, execute the sequence repeatedly until you see the SCM prompt.

CAUTION: Never use the **Ctrl–C** *sequence during power–up testing. If you want to interrupt the automatic boot sequence, wait until you see the message* `Passed`.

Halting the Operating System

You can use an operating system command line to stop all processors and display the SCM prompt.

CAUTION: Halting your system while the operating system or other software is running may result in lost or corrupted data.

If you are running the DG/UX™ system, always try the following command sequence to shut down the operating system properly:

 # cd /
 # shutdown

NOTE: You can modify the **shutdown** command to specify whether you want to bring down software immediately, or to provide a period of time for users to log out. Refer to your DG/UX documentation for information.

Then, halt the operating system to display the SCM prompt, as follows:

```
# halt -q
SCM>
```

If you don't have a DG/UX™ system, refer to your operating system documentation for information about properly shutting down your operating system before halting the system.

The following sections describe other situations which result in a system halt.

Using a Command Break Sequence

Most systems support a keyboard break sequence you can use to halt current processing and return to the SCM. Which keys you press depend on both your keyboard type and the support of your operating system or stand–alone program. For information about the proper keyboard sequence to halt your system and enter the SCM, refer to the documentation that came with your operating system or other system software.

CAUTION: *Halting your system while the operating system or other software is running may result in lost or corrupted data.*

If your DG/UX system is hung (seems frozen and you cannot continue operation), typing the following key sequence at the system console will halt the operating system and display the SCM prompt:

```
^][][][
```

Resetting the Hardware

If your operating system or stand–alone program is hung (seems frozen and you cannot continue operation), resetting the hardware may enable you to restart.

CAUTION: *Always try to shut down your operating system properly before attempting to reset the computer. Resetting the hardware while your operating system or other software is running may result in lost or corrupted data.*

You can reset your system by pressing a hardware reset switch on the computer unit (a *warm* reset), or by turning power off and then on again (a *cold* reset). You see the SCM prompt after a warm reset; the computer completes power–up testing and its automatic boot sequence after a cold reset. Refer to the manual that describes setting up and starting your system for information about your system's operator switches.

Encountering an Unsupported Breakpoint or Interrupt

Your operating system or other system software handles all exceptions (program breakpoints and interrupts); the program halts whenever it encounters an exception it does not expect or cannot handle. The SCM implements breakpoints with the trap exception; for information about inserting or removing program breakpoints, refer to the **TRAP** and **UNTRAP** command reference pages in Chapter 3. For more information about program breakpoints and interrupts, refer to the AViiON hardware programming manual for your system, listed in the Preface. For information about exception handling and vector space, refer to the Motorola documentation listed in the Preface.

An operating system (or any user program) can pass control to the SCM by executing the SCM **.HALT** system call. In a multiprocessor environment, all active processors (those started with a **.START** system call) must receive the **.HALT** system call. Refer to Appendix B for information about SCM system calls.

When Do You Use the SCM?

Your system uses SCM default boot paths to boot your operating system at every powerup and, optionally, to run a stand–alone program (extended hardware diagnostics, for example) on a routine basis *before* bringing up the operating system. Whenever there is a system fault or a problem that the operating system does not handle, you enter the SCM automatically. When this happens, you need the SCM to return control to your operating system by resuming or rebooting system software. You may also choose to enter the SCM to change system configuration parameters such as the baud rate for your system console or a default boot path.

In general, the operating system manages your operating environment when the system is running routinely; however, there are several instances when you need to use SCM commands or menus for the following:

- To respond to system errors
- To boot an operating system or stand–alone software
- To change system configuration parameters
- To control program flow
- To debug programs

The following sections describe these instances for using the SCM and tell you where to find information about each in this manual.

Responding to System Errors

You may find yourself at the SCM prompt unexpectedly if your operating system or stand–alone program halts or if your computer hardware encounters an error; you will need to use the SCM to respond. In most cases, you will record what happened and try to reboot your operating system. Refer to your hardware installation or startup manual for information about resolving power–up problems and error messages. Refer to your operating system documentation for information about resolving problems with the operating system.

Booting Software

Each time you turn on your computer, the SCM runs a basic set of power–up diagnostic programs and then attempts to boot your operating system. Once you initialize a default system boot path, your operating system (or any bootable program you specify) comes up automatically at each powerup. If you initialize a default diagnostic boot path, the SCM boots and executes separate diagnostics (or any other bootable program you specify) after power–up testing, and *then* loads your operating system.

You use the SCM to set the default boot paths, to boot your operating system from a different (nondefault) device, and to load other bootable media. For more information on these tasks, refer to the following:

- **BOOT** command reference pages in Chapter 3
- Change Boot Parameters menu in Chapter 2
- Appendix A, "Specifying a Boot Path"

Changing System Configuration Parameters

You may need to view or change your system's default configuration parameters. A system programmer may need to change the operating environment in which a program executes to observe program changes for debugging purposes.

A system administrator needs to verify or modify the configuration at the following times:

- When the system is new
- When you change or add a system console, mouse, modem, or VMEbus controller
- When you load a different operating system
- When you change the memory configuration

For more information on these tasks, refer to the following:

- **FORMAT** command reference page in Chapter 3
- Chapter 2, "Using SCM Menus"
- Appendix B, "System Programming and Debugging Tools"

Controlling Program Flow

System programmers and administrators may choose to enter the SCM to control or test program flow, either in response to system errors or to debug new programs.

For more information on these tasks, refer to the following:

- Table 3–1, "SCM Line Editing Features and Keyboard Control Sequences"
- Table 3–2, "Summary of Commands and Command Functions"
- Appendix B, "System Programming and Debugging Tools"

Debugging Programs

Several SCM commands were designed specifically for system programming, diagnostic development and program debugging. For more information on these tasks, refer to the following:

- Appendix B, "System Programming and Debugging Tools"
- Table 3–2, "Summary of Commands and Command Functions"
- Change Test Parameters menu in Chapter 2

End of Chapter

Chapter 2
Using SCM Menus

This chapter describes the menus in the System Control Monitor (SCM). You use these menus to view or change system configuration parameters.

You must be at the SCM prompt to display the primary SCM menu, View or Change System Configuration, and to access all other SCM menus. Refer to Chapter 1 for information about getting to the SCM prompt.

NOTE: On some AViiON models, the system displays the View or Change System Configuration menu automatically if the automatic boot sequence fails.

The following section contains a summary of SCM menus and information about using them. The rest of this chapter describes each menu and menu item in more detail.

Summary of Menus and Menu Conventions

You access all SCM menus from the View or Change System Configuration menu. From this primary menu you can display or modify several system configuration parameters.

To display the View or Change System Configuration menu, enter the following command line at the SCM prompt:

```
SCM> F ↵
```

The system displays the View or Change System Configuration menu. *The exact wording of your menu depends on the model of your AViiON computer.* For instance, item 3 allows you to change parameters on a serial port. On most workstations, this is your mouse port; on most multiuser or server systems, this is a modem port. Your menu text reflects this difference, as shown below:

AViiON Workstations

```
View or Change System Configuration

1   Change boot parameters
2   Change console parameters
3   Change mouse parameters
4   View memory configuration
5   Change testing parameters
5   Return to previous screen

Enter choice(s) ->
```

AViiON Server Systems

```
View or Change System Configuration

1   Change boot parameters
2   Change console parameters
3   Change modem port parameters
4   View memory configuration
5   Change testing parameters
6   Change VME A24 configuration
7   Return to previous screen

Enter choice(s) ->
```

Figure 2–1 summarizes all SCM menus and how they interconnect.

Summary of SCM Menus

The SCM displays the View or Change System Configuration menu when you enter the **FORMAT** command at the prompt. ⤵

SCM> **F**

```
View or Change System Configuration

1   Change boot parameters
2   Change console parameters
3   Change serial port parameters ¹
4   View memory configuration
5   Change testing parameters
6   Change VME A24 configuration ²
7   Return to previous screen

Enter choice(s) ->
```

Item 1

```
Change boot parameters

1   Change default boot path
2   Change power–up diagnostic boot path
3   Change data transfer mode [Block]
4   Return to previous screen

Enter choice(s) ->
```

Item 2

```
Change console parameters

1   Change baud rate [9600]
2   Change character length [8 bit, no parity]
3   Change character code set [ANSI]
4   Change flow control [Enabled]
5   Change console language [US English]
6   Return to previous screen

Enter choice(s) ->
```

Item 3

```
Change serial port parameters ¹

1   Change baud rate [1200]
2   Change character length [8 bit, no parity]
3   Return to previous screen

Enter choice(s) ->
```

Item 4

```
Memory size is 64 Mbytes
Top of memory = 3FFFFFC hex.
Memory module 1 contains 64 Mbytes
Memory module 2 not present
Memory module 3 not present
Memory module 4 not present

Press any key to continue
```

Item 6

```
Change VME A24 configuration ²

1   VME A24 Page 0  [(DTB Slave Mode (VME–to–MBUS)]
2   VME A24 Page 1  [(DTB Slave Mode (VME–to–MBUS)]
3   VME A24 Page 2  [(DTB Master Mode (MBUS–to–VME
4   VME A24 Page 3  [(DTB Master Mode (MBUS–to–VME
5   Return to previous screen

Enter choice(s)->
```

Item 5 (next page)

1 The figure text for item 3 on the View or Change System Configuration menu reads "Change serial port parameters" to reflect multiple models of AViiON computers. Your actual menu text states which serial port you can change (e.g.,"Change mouse parameters" or "Change modem port parameters").

2 Only present on computers with a VMEbus.

Figure 2–1 SCM Menus

Item 1
Change baud rate

Change baud rate

1 300
2 600
3 1200
4 2400
5 4800
6 9600
7 19200
8 Return to previous screen
Current default is [9600]

Enter choice(s) ->

Item 5
Change console language

Change console language

1 U..S English
2 German
3 U.K. English
4 French
5 Swedish
6 Spanish
7 Swiss
8 Italian
9 Japanese
10 Return to previous screen
Current default is [U.S. English]

Enter choice(s) ->

Item 1
Change baud rate

Item 5
Change testing parameters

ECW Bit	Function	State
0	Reserved	–Disabled
1	Loop on error	–Disabled
2	Output to console	+Enabled
3	Percent failure	–Disabled
4	Print pass messages	+Enabled
5	Output to printer	–Disabled
6	Disassembler	+Enabled
7	Print subtest messages	+Enabled
8	Report all	–Disabled
9	Halt on error	–Disabled
10	Enable error logging	–Disabled
11	Continue on exception	–Disabled
12	Reserved	–Disabled
13	Page mode	–Disabled
14	Reserved	–Disabled
15	Reserved	–Disabled
16	Reserved	–Disabled
17	Reserved	–Disabled
18	Reserved	–Disabled
19	Reserved	–Disabled
20–31	Reserved	–Disabled

Select bit(s) to toggle ->

Any change you make at an SCM menu becomes the new *default* parameter immediately; however, the changes are not in effect until the next time you power up or reset the computer. You can reset the computer by pressing a reset switch (some models do not have one), by using the **RESET** command at the SCM prompt, or by cycling power to the system. See Chapter 3 for information about using the **RESET** command. Refer to the manual that describes how to start your AViiON computer model for information about operator switches and the power–up process. Once you reset or power up the system, the changes you make become the current system default parameters and remain unless you change them again.

NOTE: You can restore the original system configuration defaults at any time with a keyboard control sequence. Refer to Table 3–1 in Chapter 3 for how to use the Ctrl–I sequence. Using Ctrl–I intializes the system console port to the following factory defaults: 9600 baud, 8 data bits, no parity, ANSI character set, enabled flow control. At the same time, Ctrl–I also initializes the following system defaults: U.S. English keyboard language, Block transfer mode for SCSI tape drives, and 1200 baud for the mouse or modem port. Sections in this chapter describe each of these defaults and their alternate settings.

At menus that allow you to perform several different functions, you can enter multiple item numbers at the Enter choice(s)–> prompt. The SCM executes the items in sequence without returning to the menu screen until it completes all selected items. You can use a space or a comma to separate item numbers.

You can exit from any menu by selecting the last item on the menu or by pressing New Line at the prompt without selecting an item. You return to the previous menu, *except* when you select the last item on the View or Change System Configuration menu; when you exit from the View or Change System Configuration menu you return to the SCM prompt.

The rest of this chapter contains sections that describe how to use each menu and menu item accessible from the View or Change System Configuration menu.

Change Boot Parameters Menu

At powerup, the SCM completes an automatic boot sequence. You can specify one or two automatic boot devices by setting the default system boot path and, optionally, a powerup diagnostic boot path. The default boot path is usually your operating system. The powerup diagnostic boot path can be a diagnostic program, extended power–up code, or any stand–alone program that you want to run routinely at powerup. If you set the powerup diagnostic boot path, the SCM boots that program first at each powerup; then, when the program successfully completes, the SCM automatically boots whatever is in your default boot path. If you do not set the diagnostic boot path, the SCM boots only the program you specify as the default boot path.

NOTE: Data General initializes the default boot path for most AViiON models at the factory to boot the DG/UX Starter operating system from the root disk.

If you do not set a valid default boot path (or if you leave the boot path empty), the default boot paths are not initialized; the SCM then searches for a boot device according to a hardwired sequence which varies according to AViiON model.

For example, if you power on an AViiON 300 series workstation without initialized boot paths, the SCM looks for a bootstrap file on the first SCSI (Small Computer System Interface) disk; if it cannot boot from that disk, it probes the integrated Ethernet LAN and boots from the first server that responds. For specific information about your system's automatic boot sequence, refer to your computer startup manual. For more information about boot devices, refer to Appendix A.

Finally, if the SCM cannot boot automatically, it displays an `Unable to Boot` message and then the "Change boot parameters" menu. Use the Change Boot Parameters menu to initialize, change, or view (without changing) the default boot paths. You can also use the Change Boot Parameters menu to change the data transfer method for SCSI tape devices. Subsections below describe each item on the Change Boot Parameters menu.

To view or change items on the Change Boot Parameters menu, complete the following steps:

1. While in the View or Change System Configuration menu, type **1** and press New Line to select item 1, "Change boot parameters." The system displays the following menu:

```
Change boot parameters

1     Change default boot path
2     Change powerup diagnostic boot path
3     Change data transfer mode [Block]
4     Return to previous screen

Enter choice(s)->
```

2. Select the item you want to change by entering the item number and pressing New Line. Complete the instructions in the appropriate section below.

Changing the Default Boot Path

The SCM uses the default system boot path to find the boot device at powerup. It uses the same device whenever you use the **BOOT** command without an argument. The path you specify will be the default boot device for every subsequent powerup.

To change, initialize, or boot the default boot device, follow these steps:

1. While in the Change Boot Parameters menu, type 1 and press New Line to select item 1, "Change default boot path."

The system displays the current default system boot path in brackets, followed by a prompt to change the current boot path, as follows:

```
Default boot path = [ ]
Do you want to modify the default boot path? [N]
```

NOTE: The boot path is *not initialized* when the brackets are empty.

2. To keep the current boot path, press New Line at the prompt, and then skip to step 6.

```
Do you want to modify the default boot path? [N] ↵
```

3. To set or change the system boot path, type **Y** and press New Line at the prompt.

```
Do you want to modify the default boot path? [N]  Y ↵
```

4. The system displays the following prompt.

```
Enter new default boot path ->
```

Type a valid boot path (maximum of 80 characters) and press New Line. Refer to Appendix A, "Specifying a Boot Path," for information about the format for boot path arguments.

For example, type the following to set the default boot path to boot the DG/UX operating system from the first SCSI disk managed by a Ciprico SCSI adapter, boot path **sd(cisc(0),0)** :

```
Enter new default boot path -> sd(cisc(0),0)root:/dgux ↵
```

The SCM will now automatically boot DG/UX from this disk at every powerup.

To initialize the default boot path to boot from your Ciprico SCSI tape drive, enter boot path **st(cisc(0),4)** ,as follows :

```
Enter new default boot path -> st(cisc(0),4)↵
```

5. After you specify a default boot path, the SCM displays the new boot path in brackets, followed by a prompt giving you a chance to make further changes, as follows:

```
Default boot path = [sd(cisc(0),1)root:/dgux]
Do you want to modify the default boot path? [N]
```

Press New Line to accept the new boot path. Repeat steps 3 through 5 to change the boot path again.

6. The system prompts you to boot from the new default boot path.

```
Do you want to boot? [N]
```

To return to the Change Boot Parameters menu without rebooting, press New Line. The next time you do reboot, the SCM will use the new default.

To boot from the device, type **Y** and press New Line .

```
Do you want to boot? [N]  Y ↵
```

The next message you see is from the program you booted.

Changing the Powerup Diagnostic Boot Path

The path you specify in the Change Powerup Diagnostic Boot Path menu will be the first thing that boots at every powerup until you change it again. The SCM boots this device before it boots the default system boot path device. You can specify a boot path to any bootable program you want to run first at each powerup.

To change the default diagnostic boot path, follow these steps:

1. While in the Change Boot Parameters menu, type **2** and press New Line to select item 2, "Change diagnostic boot path."

   The system displays the current default diagnostic boot path in brackets, followed by a prompt to change the current boot path, as follows:

   ```
   Diagnostic boot path = [st(insc(),4)]
   Do you want to modify the diagnostic boot path? [N]
   ```

   NOTE: The diagnostic boot path is *not initialized* if the brackets are empty.

2. To keep the current diagnostic boot path, press New Line.

   ```
   Do you want to modify the diagnostic boot path? [N] ↵
   ```

   You return to the Change Boot Parameters menu.

3. To change the diagnostic boot path, type **Y** and press New Line.

   ```
   Do you want to modify the diagnostic boot path? [N] Y ↵
   ```

4. The system displays the following prompt.

   ```
   Enter new diagnostic boot path ->
   ```

   Type a valid boot path and press New Line. Refer to Appendix A, "Specifying a Boot Path," for information about the format for boot path arguments.

   For example, type the following to change the diagnostic boot path to a file called **diags** in the **stand** directory of the logical disk **usr** on the first SCSI disk managed by an integrated SCSI controller:

   sd(insc(),0)usr:/stand/diags↵

   The SCM will boot the file **diags** (AViiON System Diagnostics) from this disk at every powerup until you change the diagnostic boot path again.

5. After you specify a diagnostic boot path, the system displays the new default diagnostic boot path in brackets, followed by a prompt to change the path again, as follows:

   ```
   Diagnostic boot path = [sd(insc(),0)usr:/stand/diags]
   Do you want to modify the diagnostic boot path? [N]
   ```

   Press New Line to accept the new boot path and return to the Change Boot Parameters menu.

   Repeat steps 3 through 5 to change the new boot path.

Changing Data Transfer for SCSI Tapes

Use item 3, "Change data transfer mode" from the Change Boot Parameters menu to toggle the default data transfer method for the SCSI tape drives connected to your computer. Under most circumstances, the SCM uses Block mode, or DMA (Direct Memory Access) to load data from a SCSI tape drive. The alternate mode is Character mode, or PIO (Programmed I/O). Do not change the default Block mode setting unless you find that your system cannot boot a SCSI tape, since Block mode provides better performance than Character mode (PIO). If you have trouble booting a SCSI tape, you can try to boot a diagnostic tape using the Character mode setting to determine if there is a problem with the DMA circuitry; note that it will take longer than usual to boot the tape.

To change the data transfer mode, type **3** and press New Line to select item 3, "Change data transfer mode," while in the Change Boot Parameters menu.

NOTE: Selecting item 3 changes the data transfer mode immediately. There are no further prompts or menus. Select item 3 again to toggle it to its previous setting.

If the SCSI transfer is currently Block mode, you will change it to Character mode. If the transfer mode default is currently Character, you will change it to Block. The new data transfer mode will be the default for SCSI tape drives until you change it again. The system displays the current value for item 3 in brackets on the Change Boot Parameters menu.

Change Console Parameters Menu

Use the Change Console Parameters menu to set the operating parameters for your system console or to view the default values for these parameters. This section describes each item on the Change Console Parameters menu.

The *system console* refers to the keyboard and display device that receives power–up diagnostic test messages and from which you bring up your operating system. In most cases, the system console communicates directly with the computer's system board. On AViiON systems, the system console port is the first RS–232 port on the computer unit rear panel; your installation manual describes its rear panel location. When you change console parameters, you change the characteristics of this first serial port. Make sure that characteristic settings on the device correspond to the console parameters set with this menu.

NOTE: A workstation's system console usually consists of the graphics monitor and graphics keyboard. If you have a graphics workstation and use a display monitor as your system console, this section does not apply. Table 3–1 in Chapter 3 describes how to use the Ctrl–V key sequence to restore default video parameters for your graphics monitor.

You can use a keyboard control sequence to restore the parameters for the system console port to the following factory defaults: 9600 baud, 8 data bits, no parity, ANSI character set, enabled flow control, U.S. English keyboard language. Table 3–1 describes how to use the Ctrl–I key sequence. This section describes how to view or change each default individually, and describes available alternate settings.

To view or change the default parameters for your console port, complete the following steps:

1. While in the View or Change System Configuration menu, type **2** and press New Line to select item 2 "Change console parameters." The system displays the Change Console Parameters menu, as follows:

```
Change console parameters


1     Change baud rate [9600]
2     Change character length [8 bit, no parity]
3     Change character code set [ANSI]
4     Change flow control [Enabled]
5     Change console language [US English]
6     Return to previous screen

Enter choice(s)->
```

   The system displays the current default for each item in brackets. Items 1, 2, and 3 only apply if you use an asynchronous terminal as system console; you cannot change these parameters on a graphics monitor.

2. Select the parameter that you want to view or change by entering the item number and pressing New Line. The system prompts you with a menu. Follow the instructions in the appropriate section below.

Changing the System Console Baud Rate

You use item 1, "Change baud rate" to select the baud rate for your system console port. The system displays the current baud rate in brackets on the Change Console Parameters menu.

The default baud rate is 9600 baud; the terminal you use to power up the first time must be set to 9600 baud. Refer to the documentation that came with the terminal to determine supported baud rates, and for information about changing the baud rate setting on the terminal. Refer to your operating system documentation for information about the correct baud rate for your system console.

To view or change the system console baud rate, complete the following steps:

1. While in the Change Console Parameters menu, type **1** and press New Line to select item 1, "Change baud rate." The system displays the Change Baud Rate menu.

```
Change baud rate

1     300
2     600
3     1200
4     2400
5     4800
6     9600
7     19200
8     Return to previous screen
Current baud rate [9600]

Enter choice(s)->
```

2. Type the item number of the baud rate you want, and press New Line. The baud rate you select will be the default baud rate for your system console the next time you power up and for every subsequent powerup or system reset, until you change it again.

3. Complete any other changes to your system console parameters you need to make, following the steps for other menu items in this section.

   The next time you reset or restart your computer, the system will conform to the new configuration parameters.

4. Power down the computer if you are connecting a new terminal or need to change the terminal characteristics. Refer to the startup manual for your hardware model (listed in the Preface) for information about connecting an asynchronous terminal.

   NOTE: If you have a graphics monitor connected to your computer system, you must disconnect the graphics keyboard before powering up using the terminal as system console.

Changing the System Console Character Length

You use item 2, "Change character length" to toggle the default character length setting for your system console port. The system displays the current value in brackets on the Change Console Parameters menu. The default value is 8 data bits, no parity. Refer to the documentation that came with the terminal for information about supported character length and parity settings.

To change the character length for your system console, type **2** and press New Line to select item 2, "Change character length," while in the Change Console Parameters menu.

NOTE: Selecting item 2 changes the character length setting for your system console; there are no further prompts or menus. Select item 2 again to toggle it to its previous setting.

If the character length is currently specified as 8 bits, no parity, you will change the specification to 7 bits, even parity. If the character length is currently 7 bits, even parity, you will change it to 8 bits, no parity.

The next time you reset or restart your computer, the system will conform to the new configuration parameters.

Changing the System Console Character Code Set

You use item 3, "Change character code set" to toggle the default character code setting for the system console port. The system displays the current value in brackets on the Change Console Parameters menu. The default character code set is ANSI. Refer to the documentation that came with the terminal for information about supported character code sets.

To change the character code set for your system console, type **3** and press New Line to select item 3, "Change character code set," while in the Change Console Parameters menu.

If the character set is currently ANSI, you will change the specification to DG mode. If the character set is currently DG mode, you will change it to ANSI.

NOTE: Selecting item 3 changes the character code set for your system console; there are no further prompts or menus. Select item 3 again to toggle it to its previous setting.

The next time you reset or restart your computer, the system will conform to the new configuration parameter.

Enabling or Disabling Flow Control

You use item 4, "Change flow control" to enable or disable the default flow control setting for your terminal or workstation monitor while in the SCM. The system displays the current value for item 4 in brackets on the Change Console Parameters menu. Flow control is enabled by default. When flow control is enabled, you can use the Ctrl–S command sequence to suspend screen display and Ctrl–Q to resume screen display.

NOTE: This item only enables or disables flow control *within the SCM*; your selection at this menu does not affect your operating system or stand–alone programs. You can, however, use the SCM to enable or disable flow control in any program using the SCM **.STDIO** system call.

To change the flow control setting, type **4** and press New Line to select item 4, "Change flow control," while in the Change Console Parameters menu. If flow control (XON/XOFF protocol) is currently enabled, you will disable it. If flow control is currently disabled, you will enable it. The next time you reset or restart your computer, the system will conform to the new configuration parameter.

Changing the System Console Language

You use item 5, "Change console language" to select the international font to use with an international keyboard. The system displays the current value in brackets on the Change Console Parameters menu. U.S. English is the default language in the SCM.

The current default language appears on the Change Console Parameters menu, next to the "Change console language" item. To change the language font for your system console keyboard, complete the following steps.

1. While in the Change Console Parameters menu, type **5** and press New Line to select item 5, "Change console language." The system displays the Change Console Language menu, as follows:

```
Change console language

1     U.S. English
2     German
3     U.K. English
4     French
5     Swedish
6     Spanish
7     Swiss
8     Italian
9     Japanese
10    Other
11    Return to previous screen

Current language is  [U.S. English]

Enter choice(s)->
```

NOTE: Only some AViiON models have item 10, "Other."

2. Select the language that is compatible with your keyboard and system software by entering the item number and pressing New Line. The next time you reset or restart your computer, the system will use the new console language.

   If your keyboard language is not listed in items 1 through 9 and you want to initialize a language type not shown on the menu, select item 10, "Other." The system will prompt you for a keyboard type, showing the item number of the current type in brackets, as follows:

   ```
   Enter language type [1]:
   ```

   Contact Data General as described in the Preface for information about alternate language types supported by the DG/UX operating system.

   NOTE: When you select item 10, "Other," the console language remains U.S. English *within the SCM*; however, it passes the alternate language type to your operating system or other system software with the **.KBLAN** system call. For more information about SCM system calls, refer to Appendix B.

Change Mouse Parameters Menu

Use the Change Mouse Parameters menu to specify the proper configuration for your workstation's mouse, or to view the default values.

NOTE: If your menu item reads "Change modem port parameters," skip to the next section.

To change the default parameters for your mouse, complete the following:

1. While in the View or Change System Configuration menu, type **3** and press New Line to select item 3, "Change mouse parameters." The system displays the following menu:

```
Change mouse parameters


1     Change baud rate [1200]
2     Return to previous screen

Enter choice->
```

   The system displays the default mouse baud rate in brackets.

2. Select item 1 to change the current baud rate for your mouse device port, type **1** and press New Line to select item 1, "Change baud rate." The system displays the Change Baud Rate menu, as follows:

```
Change baud rate

1     300
2     600
3     1200
4     2400
5     4800
6     9600
7     19200
8     Return to previous screen
Current baud rate [1200]

Enter choice(s)->
```

3. Type the item number of the baud rate you want and press New Line. The default baud rate for a serial mouse device is 1200 baud. Refer to the documentation that came with your mouse for information about the correct baud rate.

   The next time you reset or restart your computer, the system will conform to the new configuration parameter.

Change Modem Port Parameters Menu

Use the Change Modem Port Parameters menu to specify the proper configuration for the modem port.

NOTE: If your menu reads "Change mouse parameters," refer to the previous section.

On many AViiON systems, the modem port is the second RS–232 port on the computer unit rear panel; your installation manual describes its rear panel location. When you change modem port parameters, you change the characteristics of this second serial port. If you do not have a modem, you can use the modem port for any asynchronous device. Make sure that characteristic settings on the device you have connected to that port correspond to the parameters set with this menu.

To view or change the default parameters for your modem port, complete these steps:

1. While in the View or Change System Configuration menu, type **3** and press New Line to select item 3, "Change modem port parameters." The system displays the Change Modem Port Parameters menu, as follows:

```
Change modem port parameters

1     Change baud rate [1200]
2     Change character length [8 bit, no parity]
3     Return to previous screen

Enter choice(s)->
```

   The system displays the default values in brackets.

2. Select the item you want to change (baud rate or character length) by entering the item number and pressing New Line. The system prompts you with a menu. Proceed with the instructions in the appropriate section below.

   The next time you reset or restart your computer, the system will conform to the new configuration parameters.

Changing the Modem Port Baud Rate

The default baud rate for the modem port is 1200 baud. The system displays the current baud rate in brackets. Refer to the documentation that came with your modem for information about the correct baud rate.

To change the baud rate of a your modem port, follow these steps:

1. While in the Change Modem Port Parameters menu, type **1** and press New Line to select Item 1, "Change baud rate." The system displays the Change Baud Rate menu.

```
Change baud rate

1     300
2     600
3     1200
4     2400
5     4800
6     9600
7     19200
8     Return to previous screen
Current baud rate [9600]

Enter choice(s)->
```

2. Type the item number of the baud rate you want, and press New Line. The baud rate you select will be the default baud rate for your modem port the next time you reset or restart the computer.

Changing the Modem Port Character Length

Use item 2, "Change character length" to toggle the default character length setting for your modem port. The system displays the current value in brackets. The default value is 8 data bits, no parity. The alternative is 7 bits, even parity. Refer to the documentation that came with your modem for information about setting the correct character length and parity.

NOTE: Selecting item 2 changes the character length setting for your modem port; there are no further prompts or menus. Select item 2 again to toggle it to its previous setting.

While in the Change Modem Port Parameters menu, type **2**, and press New Line to select item 2, "Change character length."

If the character length is currently specified as 7 bits, even parity, you will change the specification to 8 bits, no parity. If the character length is currently 8 bits, no parity, you will change it to 7 bits, even parity.

The next time you reset or restart your computer, the system will conform to the character length last set.

View Memory Configuration Selection

While in the View or Change System Configuration menu, type **5**, and press New Line to select item 5, "View memory configuration." The system displays current system memory information.

The memory configuration display shows memory module numbers for each available slot ; this numbering differs among AViiON models. For example, some list modules 0 through 7; some 0 through 6. Others show only four available modules, numbered 1 through 4.

The following example shows the memory configuration display for a sample AViiON 400 series workstation.

```
Memory size is 16 Mbytes
Top of memory = fffffc hex
Memory module 0 contains 4 Mbytes
Memory module 1 contains 4 Mbytes
Memory module 2 contains 4 Mbytes
Memory module 3 contains 4 Mbytes
Memory module 4 not present
Memory module 5 not present
Memory module 6 not present
Memory module 7 not present

Press any key to continue
```

The example below displays the memory configuration for a sample AViiON 5000 series system.

```
Memory size is 64 Mbytes
Top of memory = 3ffffffc hex
Memory module 1 contains 64 Mbytes
Memory module 2 not present
Memory module 3 not present
Memory module 4 not present

Press any key to continue
```

Change Test Parameters Menu

You can modify test parameters for program debugging or for diagnostic program development by enabling or disabling individual bits in the Environment Control Word (ECW). Each bit of the ECW sets a particular test parameter. The SCM displays the state of each ECW bit and allows you to toggle the setting.

The changes you make by toggling bits in the Change Testing Parameters menu do not affect your operating system or stand–alone programs; the ECW determines the testing environment *in the SCM only* unless your program reads or makes changes to the ECW using the **.TECW** system call. This section describes how to view or change ECW bits using the Change Testing Parameters menu. For more information about the ECW and other SCM programming tools, refer to Appendix B.

CAUTION: *Do not enable or disable ECW bits unless you are familiar with the test parameters.*

1. While in the View or Change System Configuration menu, type **6** and press New Line to select item 6, "Change testing parameters."

   The system displays the following screen. This example shows the state of the ECW at powerup.

```
ECW Bit        Function                      State
---------------------------------------------------


0              Reserved                    - Disabled
1              Loop on error               + Enabled
2              Output to console           + Enabled
3              Percent failure             - Disabled
4              Print pass messages         + Enabled
5              Output to printer           - Disabled
6              Disassembler                + Enabled
7              Print subtest message       + Enabled
8              Report all                  - Disabled
9              Halt on error               - Disabled
10             Enable error logging        - Disabled
11             Continue on exception       - Disabled
12             Reserved                    - Disabled
13             Page mode                   - Disabled
14             Reserved                    - Disabled
15             Reserved                    - Disabled
16             Run with D_cache            - Disabled
17             Run with I_cache            - Disabled
18             Run with D_MMU              - Disabled
19             Run with I_MMU              - Disabled
20-31          Reserved                    - Disabled


Select bit(s) to toggle ->
```

NOTE: Bit 10 functions only when used via the **.TECW** system call. You cannot set an error log directly with the Change Testing Parameters menu.

2. To change test parameters, enter the bit number(s) you want to enable or disable and then press New Line at the Select bit(s) to toggle–> prompt. Separate each bit number by either a space or a comma.

   You can specify any combination of bit numbers at the prompt. The system enables those specified bits that were previously disabled, and disables those specified bits that were previously enabled.

   Exit by pressing New Line *without* entering a bit number at the prompt.

   The next time you reset or restart your computer, the system will conform to the new ECW state.

Change VME Configuration Menu

Use item 6, “Change VME A24 configuration” to alter the way default system address mapping allows access to VME A24 space.

CAUTION: *Do not make changes in the VME A24 configuration menu unless you have one or more A24–type controllers installed and are familiar with VME programming.*

Logic within the system board controls access to portions of system address space. Address decoders, in conjunction with programmable address maps, regulate accesses to and from the system board. The Mbus Address Decoder (MAD) enables Mbus access, and the VMEbus Address Decoder (VAD) enables access from a VME controller to a location or resource in system memory. MAD and VAD mapping determine, in part, how the 16–megabyte A24 address space is accessed. AViiON 5000 and 6000 series PROM code loads and verifies the MAD and VAD to default values at power up. The “Change VME A24 configuration” menu allows you to change the default A24 space mapping.

An A24 board that directs data transfers between itself and other VMEbus boards contains a *MASTER module*. If the same board contains memory accessible from the VMEbus, it also contains a *SLAVE module*. When a VME A24 Location Module is in SLAVE mode, it detects Data Transfer Bus (DTB) cycles initiated by a MASTER and can transfer data between itself and the MASTER. When A24 is in MASTER mode, it initiates DTB cycles in order to transfer data between itself and a SLAVE module. Since A24 space is partitioned into four 4–megabyte pages, you can specify which pages of A24 Address space function in SLAVE mode and which are in MASTER mode using this menu.

The combination, type, and use of VME A24 controllers in your system determines how you should configure A24 address space.

Refer to *Programming System Control and I/O Registers: AViiON 5000 and 6000 Series Systems* for more information about memory maps and address decoding. Refer to *The VMEbus Specification* for more information about DTB master and slave functionality.

To change the default VME A24 configuration, follow these steps:

1. While in the View or Change System Configuration menu, type **6** and press New Line to select item 6, “Change VME A24 configuration.” The system displays the following menu:

```
Change VME A24 configuration

1 VME A24 Page 0  [(DTB Slave Mode (VME-to-MBUS)]
2 VME A24 Page 1  [(DTB Slave Mode (VME-to-MBUS)]
3 VME A24 Page 2  [(DTB Master Mode (MBUS-to-VME)]
4 VME A24 Page 3  [(DTB Master Mode (MBUS-to-VME)]
5 Return to previous screen

Enter choice(s)->
```

   NOTE: Only AViiON 5000 and 6000 series systems display this menu.

2. Type the item number of the page of VME A24 space you want to change and press New Line.

   The default is shown above; the system displays the current value in brackets. If an A24 page is currently configured in SLAVE mode, you will change the default to MASTER mode by selecting the corresponding menu item. If the default is currently MASTER, you will change it to SLAVE.

   The next time you reset or restart your computer, the system will conform to the new configuration parameters.

End of Chapter

Chapter 3
Using SCM Commands

The System Control Monitor (SCM) provides commands for controlling and debugging programs, booting media, and changing system configuration parameters. This chapter describes how to use the command interpreter and defines each SCM command. Refer to Chapter 1 for an overview of the SCM and information about accessing the command interpreter.

The following section explains how to execute commands and describes command interpreter conventions. The last section of this chapter provides command reference pages for each SCM command, in alphabetical order by command name.

Using the Command Interpreter

A command line consists of one valid SCM command and, in many cases, one or more arguments (required or optional) that you enter at the SCM prompt. Refer to Chapter 1 for a description of the SCM prompt.

Whenever you use a command incorrectly, the SCM displays a general message and then returns the prompt. For example, if you enter a command that does not exist, or type too many or invalid characters in a command line, you see the message, Invalid command. If you use a valid command without a required argument, you see the message, Requires argument(s). If you use an argument improperly, you see the response, Invalid argument(s). The reference pages later in this chapter list more specific system messages that are unique to particular commands.

Line Editing Conventions

You can type a maximum of 80 characters in one command line. You do not have to type the entire command name when entering commands. The SCM always accepts the first letter of a command as its minimal mnemonic.

SCM commands and arguments are *not* case–sensitive, with the exception of device specification arguments to the **BOOT** command, which must be lowercase.

The SCM supports several keyboard control characters. Table 3–1 describes keyboard control sequences you can use to edit command lines, to interrupt and exit from several SCM commands, and to restore configuration parameters. Refer to the Preface for additional information about keyboard character symbols and how they are used in this manual.

Table 3–1 SCM Line Editing Features and Keyboard Control Sequences

Keyboard Entry	Function
↵ [1]	Completes the current input line, begins execution of command input, and returns the SCM prompt.
<Ctrl–A>	Recalls and displays the last command string you entered at the SCM prompt.
<Ctrl–C> [2]	Interrupts execution of an SCM command and returns the SCM prompt. This is a polled interrupt; some procedures complete before they break. If you do not have an auto–repeat keyboard, execute the Ctrl–C sequence repeatedly until you see the SCM prompt.
<Ctrl–I> [3]	Restores default configuration parameters to the following factory settings. System console port: 9600 baud, 8 data bits, no parity, ANSI character set, enabled flow control, U.S. English keyboard language. Parallel printer: Centronics interface. SCSI tape drives: block transfer mode. Mouse or modem port: 1200 baud. Also restores video timing parameters (see Ctrl–V below). Enter **<Ctrl–I>**, wait until you hear one beep; then, enter **1** if you have a 70 Hz. monitor, or enter **2** if you have a 60 Hz. monitor.
<Ctrl–P> [3]	Displays the current state of the Environment Control Word (ECW). Refer to Appendix B for more information.
<Ctrl–Q> [4]	Resumes SCM output display that was suspended with the.Ctrl–S sequence.
<Ctrl–S> [4]	Suspends SCM output display until you resume it with the Ctrl–Q sequence.
<Ctrl–U>	Erases the current line of text, from the left of the cursor to the SCM prompt.
<Ctrl–V>[3]	Allows you restore the default video parameters for the 70 Hz. graphics monitor, or to change the defaults for a 60 Hz. monitor. Use this control sequence only if the window that appears at powerup on your graphics monitor is distorted or does not appear. After you type Ctrl–V, wait until you hear a beep; then, type **1** for 70 Hz. parameters, or type **2** to configure for a 60 Hz. monitor. The monitor is configured to the new frequency when you hear a second beep. If the screen display on your graphics monitor is not clear after trying both frequencies, refer to the "Solving Power Up Problems" section in the hardware startup manual that came with your computer.

1 The New Line and Enter keys have special functions when used with the **EXAMINE** command. Refer to Table 3–5 for **EXAMINE** command key functions.
2 Only functions as an interrupt to SCM functions.
3 You can execute this sequence only while in the SCM.
4 Only works when Flow Control protocol is enabled within the SCM. (See Chapter 2).

Address and Data Conventions

The address arguments you use with SCM commands consist of two 16–bit, hexadecimal words. You may omit leading zeros. SCM commands support physical and logical address arguments according to the current mode of the memory management unit (MMU). When MMU translation is on, logical addresses map to physical addresses.

The assembler assumes data input is in a decimal radix unless there is a dollar sign ($) preceding the data to indicate a hexadecimal radix. The disassembler returns hexadecimal data, using the same convention. The SCM displays data output mnemonics and accepts data input mnemonics by default, since both the assembler and disassembler are enabled by default. (For information about disabling and enabling the disassembler, refer to the section “Change Test Parameters Menu” in Chapter 2.)

When the SCM displays the contents of a memory location, the display appears in the following format when MMU translation is on:

```
Memory logicaladdress - physicaladdress  /  data - mnemonic
```

For example, if the contents of memory location 10 is 5555FFFF16 the SCM displays the following:

```
Memory 00000010 - 00000010 / 5555FFFF - xor.u   r10 r21 $FFFF
```

The data mnemonic includes the opcode (xor.u in the example above), the registers pointed to by the first 16–bit word of the 32–bit address (r10 and r21), and the hexadecimal data in the second word of the address ($FFFF).

Command Reference Pages

This remainder of this chapter contains command reference pages for each SCM command. The reference pages describe the command function, its associated arguments, and examples of its use.

NOTE: The SCM in your system may not implement every command listed in this chapter. For instance, systems with one processor do not use the **ATTACH** command. You can use the **HELP** command on any AViiON system for a list of available SCM commands.

The following two sections (preceding the individual command reference pages) describe the format of the reference pages and summarize the commands for quick reference.

Format of Command Reference Pages

The sections that describe each SCM command follow the format shown below.

COMMAND–NAME
Brief command description.

COMMAND *argument [argument]*

NOTE: The minimal mnemonic for each command is always the *first letter* of the command name; it appears in larger type than the rest of the command name.

Description

This section provides a functional definition of the command, includes related comments and requirements, and describes any special key functions.

Arguments

This section defines each command argument.

None	The command accepts no arguments.
argument	The command requires the argument defined here.
[argument]	You have the option of including or omitting arguments defined here. Do not type the brackets.

Related Commands

This section lists closely related SCM commands.

Related Messages

This section lists any system messages that you might receive when executing this particular command. The messages listed here are only those that are unique to the command; general system messages are described at the beginning of this chapter.

Examples

This section provides realistic examples to show how the command works. Where possible, we include examples using each argument or type of argument. We number multiple examples and annotate long examples for clarity.

Summary of Commands

You can use SCM commands for several different purposes. Table 3–2 lists likely functions (debugging, program control, or system operation) for each SCM command. This may help you determine which commands you need to learn and use. For example, a system operator should not attempt to use debugging commands without understanding the meaning and results of viewing or changing register and memory location contents. Your system may not implement every command described in this manual. Use the **HELP** command to view the available commands for your system.

Table 3–2 Summary of Commands and Command Functions

Command	Description	Function
. (period)	Displays processor status	Debugging
ATTACH	Specifies attached processor	Program control, system operation
BOOT	Starts system from bootstrap device	System operation
CONTINUE	Restarts attached processor	Program control, debugging
DISPLAY	Shows register file contents	Debugging
EXAMINE	Opens or edits contents of registers and memory locations	Debugging
FORMAT	Displays View or Change Configuration menu	System operation, debugging program control
HELP	Lists valid SCM commands	System operation, debugging program control
INITIALIZE	Writes data to a range of memory	Debugging
LOCATE	Searches memory for a data pattern	Debugging
MOVE	Duplicates a memory block in new location	Debugging
ONESTEP	Executes the next program instruction	Debugging, program control
PROMPT	Changes text of SCM prompt	System operation
RESET	Initializes system to power–up state	System operation
START	Begins processor at specified address	Program control, system operation
TRAP	Views or inserts breakpoints	Debugging
UNTRAP	Removes breakpoints	Debugging
VIEW	Displays a range of memory	Debugging
WRITE	Inserts data in one memory location	Debugging
ZLOADER	Starts s–record loader utility	System operation

. (period) Displays job processor status.

. (period)

Description

The . (period) command displays status information about the attached job processor and its program registers. You see the same information each time the processor halts to enter the SCM.

The SCM processor status display includes the following:

- PSR (Processor Status Register)
  The processor state of the program that was last running on the attached processor. The PSR is the value in Control Register 1 (cr1).
- XPC (Execute Program Counter)
  The contents of the program counter (PC) of the program that caused entry into the SCM. The XPC is the value in Control Register 4 (cr4).
- DCSH (Data Cache Enable/Disable)
  The state of the user program data cache before entering the SCM. Y indicates enabled, and N indicates disabled.
- DMMU (Data Memory Management Unit Enable/Disable)
  The state of the user program data MMU before entering the SCM. Y indicates enabled, and N indicates disabled.
- ICSH (Instruction Cache Enable/Disable)
  The state of the user program instruction cache before entering the SCM. Y indicates enabled, and N indicates disabled.
- IMMU (Instruction Memory Management Unit Enable/Disable)
  The state of the user program instruction MMU before entering the SCM. Y indicates enabled, and N indicates disabled.

Arguments

None

Related Commands

ATTACH	Specifies the attached job processor in multiprocessor systems.
CONTINUE	Resumes program execution at the next program counter (NPC) address.
ONESTEP	Executes the next single instruction of a program and then displays trace information.

Related Messages

None

Examples

1. Display processor status on single processor system.

```
SCM> . ⤶

PSR       XPC       DCSH      DMMU    ICSH  IMMU
A00003F2  FFC039DE  N         N         N     N
```

2. Display processor status information on multiprocessor system.

```
Jp#0/SCM> . ⤶

Jp#0/PSR  XPC       DCSH      DMMU    ICSH  IMMU
A00003F2  FFC039DE  N         N         N     N
```

ATTACH
Specifies active job processor.

ATTACH *[jp#]*

Description

The **ATTACH** command is only valid for multiprocessor systems. It allows you to attach the SCM to a specified job processor for subsequent operations (in other words, it makes a particlar processor active). Unattached processors remain in an idle state. The SCM prompt indicates which job processor is currently attached (Jp#n/SCM> where n is the number of the attached job processor). By default, Jp#0 is the attached processor after powerup.

The system returns the currently attached processor if you use this command without an argument.

Arguments

[jp#]	The number of the job processor to attach.

Related Commands

. (period)	Displays status information about the attached processor.
CONTINUE	Resumes program execution at the program counter value on the currently attached processor.
PROMPT	Changes the SCM prompt text.
START	Begins program execution at a specified address on the currently attached processor.

Related Messages

Jp#n attached

Examples

1. Attach SCM to job processor #1.

   `Jp#0/SCM>` **A 1** ↵

   `Jp#1 attached`

2. Display the currently attached job processor.

   `Jp#1/SCM>` **A** ↵

   `Jp#1 attached`

BOOT
Starts the system from a specified device.

BOOT *[dev([cntrl],[unit],[file#])][filepath]]*

Description

The **BOOT** command first resets the system hardware; it then loads a bootstrap program from a valid device you specify in an optional argument. Valid boot device arguments vary according to your computer model and peripheral configuration. The format for a boot file argument conforms to the common object file format (coff) as given in section 4 of the Binary Compatibility Standard (BCS). Appendix A describes these arguments and their optional parameters in detail.

When you use the **BOOT** command *without* an argument, the SCM attempts to boot from a default boot path. Refer to Chapter 2 for information about using the Change Boot Parameters menu to initialize or change the default boot path. If the default boot path is not initialized or not valid, the SCM tries to find a valid bootstrap file according to a hardcoded probe sequence specific to the model of your AViiON system. For instance, on most workstations the SCM probes first for a bootstrap file on the first SCSI disk; if it cannot boot from the local SCSI disk, it probes for a device on the integrated Ethernet LAN. Finally, if the SCM cannot boot automatically, it displays the Change Boot Parameters menu.

If you type an *incorrect* argument to the **BOOT** command, the system does one of two things. If the device *mnenomic* was invalid, it interprets the text string you typed as a file path specification for a second–stage boot; it boots the default system boot path and passes what you typed as a second–stage bootstrap argument. The second–stage bootstrap then tries to find an executable image with the name of the string you typed. If the device mnenomic was valid but the *parameters* incorrect, you see an error message and return to the SCM prompt.

Arguments

[dev([cntrl],[unit],[file#])]	*dev* is a mnemonic name for the boot device (always lowercase). The device driver is named by *dev* followed by open and close parentheses. The parentheses contain optional parameters to fully specify the device. Tables 3–3 and 3–4 list supported devices with mnemonics and parameters; Appendix A describes boot path arguments in further detail.
[filepath]	Specifies an executable filename or an Internet host address, used and defined by the bootstrap program in a second–stage boot sequence. Refer to Appendix A for information about specifying a filepath argument.

BOOT (continued)

Table 3–3 Integrated Devices

Mnemonic	Device Type	Parameters		
		cntrl	unit	file#
inen	Integrated Ethernet controller	N/A	N/A	N/A
insc	Integrated SCSI controller	N/A	N/A	N/A
sd	SCSI disk	**insc( )**	SCSI ID[1]	**0**
st	SCSI tape	**insc( )**	SCSI ID[1]	Tape file number

[1] An integer 0 through 6, determined by configuration jumpers. Refer to Table NO TAG for values.

NOTE: If you have a computer with an integrated SCSI bus, arguments in Table 3–3 apply. If your computer has a primary VME bus, arguments in Table 3–4 apply.

Table 3–4 VME Devices

Mnemonic	Device Type	Parameters		
		cntrl	unit	file#
sd	SCSI disk	SCSI adapter **cisc( )**	SCSI ID[1]	**0**
st	SCSI tape	SCSI adapter **cisc( )**	SCSI ID[1]	Tape file number
cisc	Ciprico SCSI adapter	Adapter number or VME address	**0**	**0**
cied	Ciprico ESDI disk	Controller number or VME address	Unit number[2]	**0**
cimd	Ciprico SMD disk	Controller number or VME address	Unit number[2]	**0**
hken	Interphase Hawk Ethernet	**0**	**0**	**0**

[1] An integer 0 through 6, determined by configuration jumpers. Refer to Table NO TAG for values.
[2] An integer 0 through 3, assigned to differentiate devices on the same disk controller.

Related Commands

FORMAT Displays the View or Change System Configuration menu. You can display the Change Boot Parameters menu from this menu.

Related Messages

`Booting from ...`

`Unable to load boot file ...`

Examples

1. Boot the default system boot path.

   SCM> **B** ↵

2. On a workstation with an integrated SCSI bus, boot from the first 150 Mbyte QIC tape drive (SCSI ID#4).

   SCM> **B st(insc(0),4,0)** ↵

3. On a system with one Ciprico VME SCSI adapter, boot from the first 150 Mbyte QIC tape drive (SCSI ID#4).

   Jp#0/SCM> **B st(cisc(0),4,0)** ↵

4. Boot the file **diags** located in the directory called **stand** on the logical disk **usr** from the default system disk.

   SCM> **B sd(cisc(0),0)usr:/stand/diags** ↵ (VME SCSI disk)

   or

   SCM> **B sd(insc(0),0)usr:/stand/diags** ↵ (Integrated SCSI disk)

5. Boot a file called **bootfile** in the root directory on the second SCSI disk (SCSI ID#1).

   SCM> **B sd(cisc(0),1,0)root:/bootfile** ↵ or **B sd(insc(0),1)root:/bootfile** ↵

6. Boot from the third disk (Unit 2) on the second VME ESDI controller.

   Jp#0/SCM> **B cied(1,2)** ↵

7. Boot from the first host that responds on the Ethernet LAN.

   SCM> **B inen()** ↵ or Jp#0/SCM> **B hken()** ↵

8. Boot from the host at Internet address 128.111.5.6.

   SCM> **B inen()128.111.5.6:** ↵

   or

   Jp#0/SCM> **B hken( )128.111.5.6:** ↵

CONTINUE
Restarts attached processor.

CONTINUE *[trace–count]*

Description

The **CONTINUE** command resumes program execution at the address stored in the program counter (NPC) of the attached processor, for the number of instructions specified in the optional trace count argument. The NPC is CR5 (Control Register 5). When you use the command without the trace count argument, system control passes completely to the continued program.

Arguments

[trace–count]	The hexadecimal number of instructions you want the program to execute. The system displays the address, data, and mnemonic (in that order) after each instruction, then halts, displays processor status information, and returns the SCM prompt.

Related Commands

. (period)	Displays attached processor status information, including the value of the NPC.
ATTACH	Specifies the attached job processor in multiprocessor systems.
DISPLAY	Shows the contents of all register files without allowing you to modify them.
START	Begins program execution at a specified address.

Related Messages

None

Examples

1. Resume program execution at the NPC value; leave the SCM.

```
SCM> C ⤶
```

2. Resume program execution at the current PC value, display trace information after the next three instructions; then halt the processor and return to the SCM.

```
SCM> C 3 ⤶

Trace        00000010        5555FFFF xor.u     r10 r21 $ffff
Trace        00000014        00000000 xmen.bu   r0 r0 $0
Trace        00000018        00000000 xmen.bu   r0 r0 $0

 PSR      XPC      NPC      FPC       DCSH  DMMU   ICSH  IMMU
 A0000000 0000001A 0000001E 00000022   Y     Y      N     N

SCM>
```

DISPLAY
Shows contents of register files.

DISPLAY

Description

The **DISPLAY** command shows the contents of general register files (r), control register files (cr), and floating–point control registers (fcr) in the attached processor. Without an argument, only general register files are displayed.

Arguments

None

Related Commands

ATTACH Specifies the attached job processor in multiprocessor systems.

EXAMINE Displays and optionally changes specified registers or memory locations.

Related Messages

None

Examples

Display the current content of all register files.

```
SCM> D ⤶
  r00 = 00000000  r01 = 00066B58  r02 = FFC00000   r03 = 00000000
  r04 = 0000000C  r05 = FFF00000  r06 = 0016E570   r07 = 0016B340
  r08 = 00003230  r09 = 00000063  r10 = 00000998   r11 = FFFFFFA1
  r12 = 00000000  r13 = 00000000  r14 = 00000000   r15 = 00000000
  r16 = 00000000  r17 = 00000000  r18 = 00000000   r19 = 00000000
  r20 = 00000000  r21 = 00000000  r22 = 00000000   r23 = 00000000
  r24 = 0016E570  r25 = 0000002C  r26 = 000E0000   r27 = 000F0000
  r28 = 00010000  r29 = 00000000  r30 = 017DEFD0   r31 = 017DEFC8
 cr00 = FFF8113C cr01 = 800003FB cr02 = 800003f0  cr03 = 00000000
 cr04 = 0006009A cr05 = 0006009E cr06 = 000600A2  cr07 = FFC00000
 cr08 = 0000403F cr09 = 0000403F cr10 = 017DEF40  cr11 = 00000000
 cr12 = 017DEF58 cr13 = 017DEF4C cr14 = 00000000  cr15 = 00000000
 cr16 = 00000000 cr17 = 00066B58 cr18 = FFC00000  cr19 = 00000000
 cr20 = 00000000fcr00 = 00000000fcr01 = 00000000 fcr02 = 00000000
fcr03 = 00000000fcr04 = 00000000fcr05 = 000048A2 fcr06 = 00000000
fcr07 = 00000000fcr08 = F8004808fcr62 = 00000001 fcr63 = 00000000
```

EXAMINE

Opens and optionally changes the contents of selected registers or memory locations.

EXAMINE [*[M] address* | *H address* | *Q address* | *Rnumber* | *CRnumber* | *FCRnumber*]

Description

The **EXAMINE** command opens and displays the contents of a specified memory address or register file. After opening a memory location, you can use special key functions described in Table 3–5 to enter new data, open the next or previous memory location, or return to the SCM prompt without making changes.

CAUTION: There are no restrictions to the areas of memory you can modify; modifying system control registers or NOVRAM locations could halt your system or destroy necessary system information. Ctrl–C will not recover data already modifed.

Use one argument per command line to specify whether you want to open a memory location or register file, and whether to view memory locations in word (32 bit), half–word (16 bit), or quarter–word (8–bit, or single byte) increments.

Without an argument, the **EXAMINE** command opens and displays the last memory addressed examined. After the system halts or the first time after powerup, the SCM opens memory location 0 when you use the command without an argument.

Table 3–5 Special Key Functions for EXAMINE Command

Standard PC Keyboard	DASHER Keyboard	Function With EXAMINE Command
Escape	New Line or Break ESC	Writes data if entered; then closes the memory location or register and returns the SCM prompt.
Enter	Carriage Return	When a register file is open, closes the register and returns the SCM prompt. When a memory location is open, writes data if entered, closes the current memory location, and opens the next location.
Shift–6	Shift–6	When a register file is open, closes the register and returns the SCM prompt. When a memory location is open, writes data if entered, closes the current memory location, and opens the previous location.

Argument

[[M] address]	The 32–bit (single word) memory location you want to examine or modify. Typing **M** is optional. If you type a number without a **M**, **H**, **Q**, **R**, **CR**, or **FCR** before it, the SCM interprets the number as a 32–bit memory address. Any address is masked to be a true 32–bit word (bits 0 and 1 are cleared).
[H address]	The 16 bits (half–word) of memory you want to examine or modify. You must type **H**, followed by a space and the memory location. Any address is masked to be a true half–word (bit 0 is cleared).
[Q address]	The 8 bits (one byte) of memory you want to examine or modify. You must type **Q**, followed by a space and the memory location. No address masking.
[Rnumber]	The CPU register file you want to examine or modify. You must type **R**. Valid entries are R0 through R31.
[CRnumber]	The control register you want to examine or modify. You must type **CR**. Valid entries are CR0 through CR20.
[FCRnumber]	The floating point register you want to examine or modify. You must type **FCR**. Valid entries are FCR0 through FCR8, FCR62, or FCR63.

Related Commands

DISPLAY	Shows the contents of register files without allowing you to modify them.

Related Messages

None

EXAMINE (continued)

Examples

NOTE: The examples in this section include keyboard characters for a standard IBM PC AT–compatible keyboard. Refer to Table 3–5 for the appropriate keys to use if you have a Data General DASHER keyboard. Refer to the Preface for a description of how keyboard symbols are used in this manual.

1. Display the contents of memory address 1000 without modifying. Disassembler and MMU are disabled in this example.

   SCM> **E M 1000** ↵ or SCM> **E 1000** ↵

   Memory 00001000 / 58670004 <**Esc**>

   SCM>

2. Display the contents of memory address 1000 without modifying. Disassembler and MMU are enabled in this example.

   SCM> **E 1000** ↵

   Memory 00001000 - 00001000/12345678 - ld.d r17 r20 $5678 **Esc**

   SCM>

3. Display without modifying the contents of last memory address examined.

   SCM> **E** ↵

   Memory 00001000 - 00001000/12345678 - ld.d r17 r20 $5678 **Esc**

   SCM>

4. Display without modifying the contents of register file r03.

   SCM> **E R3** ↵ or SCM> **E R03** ↵

   R03 = 00000000

   SCM>

5. Display without modifying the contents of control register file cr01.

   SCM> **E CR01** ↵ or SCM> **E CR1** ↵

   r03 = 80000F3B ↵

   SCM>

6. Enter data 12345678 in memory address 1000; then exit from the command and return to the SCM.

    SCM> **E M1000 ↵** or SCM>**E 1000 ↵**

    Memory 00001000 / 58670004 **12345678 <Esc>**

    SCM>

7. Display the contents of memory address 1000; deposit data 12345678 in memory address 1004; then return to the SCM.

    Jp#0/SCM> **E 1000 ↵**

    Memory 00001000 / 12345678 <**CR**>

    Memory 00001004 / 58670004 **12345678 <Esc>**

    SCM>

8. Display the contents of floating–point control register file fcr62; then deposit data 5555FFFF into fcr62.

    SCM> **E FCR62 ↵**

    fcr62 = 00000000 **5555FFFF ↵**

    SCM>

9. Display the first 16 bits at location 1000.

    SCM> **E H 1000 ↵**

    Memory 00001000 / 1234 <**Esc**>

    SCM>

10. Display the first 8 bits at location 1000.

    SCM> **E Q 1000 ↵**

    Memory 00001000 / 12 <**Esc**>

    SCM>

FORMAT
Displays Configuration Menus.

FORMAT

Description

The **FORMAT** command displays the "View or Change System Configuration" menu. You access all SCM menus to set configuration parameters from the View or Change System Configuration menu. The text of menus may vary according to your AViiON system model.

Press New Line or select the last item at the View or Change System Configuration menu to return to the SCM prompt. Refer to Chapter 2 for further information about using SCM menus.

Arguments

None

Related Commands

BOOT	Boots a device or displays the Change Boot Parameters menu.

Related Messages

None

Examples

Display the View or Change System Configuration menu on an AViiON 5000 series system.

```
SCM> F ↵

View or Change System Configuration

1  Change boot parameters
2  Change console parameters
3  Change modem port parameters
4  View memory configuration
5  Change testing parameters
6  Change VME A24 configuration
7  Return to previous screen

Enter Choice(s) ->
```

HELP
Displays available SCM commands.

HELP

Description

Displays an alphabetical list of the minimal mnemonic for valid SCM commands, the arguments each command accepts, and a brief command description.

Arguments

None

Related Messages

None

Examples

Determine valid SCM commands (on an AViiON 5120 system, in this example), their associated arguments, and what you can do with each one.

SCM> **H** ↵

```
*** Model 5000/6000 Series - SCM Commands - Rev xx.xx ***
.                                   -Display processor status
A  [Jp#n]                           -Attach a processor
B  [dev([cntrl],[unit],[part])]     -Boot a device
C                                   -Continue
                                    -Display all registers
E  address                          -Examine/modify memory
E  register                         -Examine/modify R#, CR#,FCR#
F                                   -View/Change System Configuration
H                                   -Display help message
I  data  beg_addr  end_addr         -Initialize memory range
L  data  [beg_addr  end_addr]       -Locate data pattern
M  count source_addr dest_addr      -Move memory range
O  [trace count]                    -Single step
P  [new prompt]                     -Change prompt
R                                   -Reset system
S  address [trace count]            -Start processor
T  [address] [...]                  -Set breakpoint
U  [breakpoint #] [...]             -Delete breakpoint
V  beg_addr [end_addr]              -View memory range
W  address data                     -Write to a location without read
Z                                   -Start s-record loader
```

NOTE: The first command is a period; press the period key and then New Line.

INITIALIZE

Writes specified data to a range of memory.

INITIALIZE *data beg–addr end–addr*

Description

The **INITIALIZE** command writes specified data to a range of memory addresses that starts at the specified 32–bit beginning address and ends at the 32–bit ending address.

CAUTION: There are no restrictions to areas of memory you can initialize. Initializing system control registers or NOVRAM locations could halt your system or destroy necessary data. Use the Ctrl–C sequence to exit from the command during processing.

Arguments

data	Content written to each location of the memory range you are initializing.
beg–addr	First location in the range of memory.
end–addr	Last location in the range of memory.

Related Commands

MOVE	Duplicates the contents of a specified range of memory and writes to a range beginning at a specified destination address.
VIEW	Displays the contents of a range of memory.
WRITE	Writes data to a single memory location.

Examples

Initialize memory by writing data 5555FFFF into a range of memory addresses beginning at memory address 0 and ending at 10. Then view the memory range to see the results.

```
Jp#1/SCM> I 5555FFFF 0 10 ⤶

Jp#1/SCM> VIEW 0 10 ⤶

Memory 00000000  /  5555FFFF - xor.u      r10  r21  $FFFF
Memory 00000004  /  5555FFFF - xor.u      r10  r21  $FFFF
Memory 00000008  /  5555FFFF - xor.u      r10  r21  $FFFF
Memory 0000000C  /  5555FFFF - xor.u      r10  r21  $FFFF
Memory 00000010  /  5555FFFF - xor.u      r10  r21  $FFFF
```

LOCATE
Finds a specified data pattern in memory.

LOCATE *data [beg–addr] [end–addr]*

Description

The **LOCATE** command searches memory for the specified data pattern; then displays the address and contents of each location in which it finds the data. You can omit the address arguments and search all of physical memory, specify a starting address only, or specify a range of memory with starting and ending address arguments.

NOTE: A search through all of memory could take several hours! To stop a search before it completes, use Ctrl–C.

Arguments

data	Pattern in memory for which the system searches.
[beg–addr]	Location in memory where system begins searching. If you specify only one address argument, the system interprets it as a starting address.
[end–addr]	Location in memory where system stops searching.

Related Commands

MOVE	Duplicates the contents of a specified range of memory and moves a copy to a specified destination address.
VIEW	Displays the contents of a specified range of memory.

Related Messages

None

Examples

1. Locate each occurrence of data pattern 01234567 in a range of memory beginning at 0 and ending at 1000; then display the address of each occurrence.

```
Jp#1/SCM> L 01234567 0 1000 ⤶
Memory 00000800  /  01234567 - ld.d r17  r20  $4567
```

2. Locate each occurrence of data pattern 01234567 in all of main memory.

```
Jp#1/SCM> L 01234567 ⤶
Memory 00000800  /  01234567 - ld.d r17  r20  $4567
Memory 00001200  /  01234567 - ld.d r17  r20  $4567
```

MOVE

Duplicates a memory block.

MOVE *count source–addr dest–addr*

Description

The **MOVE** command copies the block of data that begins at the source address and ends after the specified count of 32–bit words; then moves the copy into a block of the same size starting at the destination address.

CAUTION: There are no restrictions to areas of memory into which you can move data. Overwriting data in system control registers or NOVRAM locations could halt your system or destroy necessary data. Use the Ctrl–C sequence to exit from the command during processing.

Arguments

count	Hexadecimal number of 32–bit words from the beginning address to the end of the block to be moved. There is no maximum or minimum block size.
source–addr	Memory location at the beginning of the range to be moved.
dest–addr	Memory location at the beginning of the range the block is moving to.

Related Commands

INITIALIZE	Writes data into a range of memory.
LOCATE	Searches memory (or a range optionally specified) to locate the specified data pattern.
VIEW	Displays the contents of a specified range of memory.

Related Messages

None

Examples

1. **Move the block that begins at 0 and spans 5 words to destination address 400.**

First, view the range of memory from locations 0 through 10.

```
Jp#1/SCM> VIEW 0 10 ↵

Memory 00000000  /  5555FFFF - xor.u      r10 r21 $FFFF
Memory 00000004  /  5555FFFF - xor.u      r10 r21 $FFFF
Memory 00000008  /  5555FFFF - xor.u      r10 r21 $FFFF
Memory 0000000C  /  5555FFFF - xor.u      r10 r21 $FFFF
Memory 00000010  /  5555FFFF - xor.u      r10 r21 $FFFF
```

Next, move the block.

```
Jp#1/SCM> M 5 0 400  ↵
```

Finally, view the contents of the block of memory from locations 400 through 410.

```
Jp#1/SCM> V 400 410  ↵

Memory 00000400  /  5555FFFF - xor.u      r10 r21 $FFFF
Memory 00000404  /  5555FFFF - xor.u      r10 r21 $FFFF
Memory 00000408  /  5555FFFF - xor.u      r10 r21 $FFFF
Memory 0000040C  /  5555FFFF - xor.u      r10 r21 $FFFF
Memory 00000410  /  5555FFFF - xor.u      r10 r21 $FFFF
```

The contents of the memory range 0 through 10 moved to the range 400 through 410.

2. **Move the data block beginning at memory address 0 and spanning 4F words to destination address 8000.**

```
Jp#1/SCM> M 4F 0 8000  ↵
```

ONESTEP
Executes next single step of a program.

ONESTEP *[trace–count]*

Description

The **ONESTEP** command begins program execution at the address stored in the program counter (CR5) in increments of one instruction. It displays trace information after each instruction executes. With no argument, the system executes the single program instruction stored in CR5; then halts and displays status information.

Arguments

[trace–count]	Hexadecimal number of instructions to be executed one by one (in single steps) before the processor halts.

Related Commands

. (period)	Displays processor status information.
CONTINUE	Begins program execution at the address stored in the program counter (PC).
START	Begins program execution at a specified address.

Related Messages

None

Examples

1. Execute the instruction pointed to by the program counter.

```
SCM> O ↵

PSR       XPC       DCSH      DMMU      ICSH   IMMU
A0000000  FF001602  Y         N           N      N
```

2. Execute three instructions one by one, beginning with the instruction pointed to by the PC.

```
SCM> O 3 ↵

Trace        00000010        5555FFFF xor.u     r10 r21 $FFFF
Trace        00000014        00000000 xmen.bu   r0 r0 $0
Trace        00000018        00000000 xmen.bu   r0 r0 $0

PSR       XPC       DCSH      DMMU      ICSH   IMMU
A0000000  0000001A  N         N           N      N
```

PROMPT

Changes SCM prompt text prefix.

PROMPT *[new–prompt]*

Description

The **PROMPT** command changes the default SCM prompt prefix to a specified ASCII string. This can be useful to uniquely identify multiple systems. The right bracket symbol (>) appears after the text prefix; if you change the prompt text to a null text string, your prompt is the right bracket symbol.

NOTE: Multiprocessor systems add the text string Jp#n/ (n = the number of the attached job processor) to the default prompt text.

Arguments

[new–prompt]	Text string of ASCII characters to replace the prompt. The ASCII string can have as many as 1510 characters. There are no character or symbol restrictions.

Related Commands

ATTACH	Specifies the attached job processor in a multiprocessor system.

Related Messages

Argument(s) out of range

Examples

1. Display the current SCM prompt; then change it to **AV5000**.

   `Jp#0/SCM>` **P ⤶**

   `Jp#0/SCM>`

   `Jp#0/SCM>` **P AV5000 ⤶**

   `Jp#0/AV5000>`

2. Change the default SCM prompt text to **Station1**.

   `SCM>` **P Station1 ⤶**

   `Station1>`

RESET

Restores system to power–up state.

RESET

Description

Initializes system elements (excluding memory) to their original power–up state. Unlike a *cold reset* (power applied to the system), a *warm reset* (initiated by software, the **RESET** command, or a Reset switch) does not initialize memory or run power–up diagnostics.

CAUTION: Be careful not to enter **R** *at the SCM prompt accidentally. You cannot use Ctrl–C or an SCM command to recover.*

Arguments

None

Related Commands

BOOT	Boots a device.

Related Messages

```
System Reset
```

Examples

Reset the system (processors, keyboard port, graphics controller, etc.).

```
SCM> R ⤶

PSR       XPC       DCSH      DMMU      ICSH  IMMU
A00003F2  FFC039DE  N         N          N     N
```

START

Begins job processor at specified address.

START *address [trace–count]*

Description

The **START** command begins executing a program at the main memory address specified. The operating system or user program resumes system control unless you use the *trace–count* argument.

Arguments

address	Memory location at which the processor starts executing.
[trace–count]	The system displays the address, data, and mnemonic (in that order) after executing the hexadecimal number of instructions you specify with this argument. Then the system halts and the monitor displays status information.

Related Commands

BOOT	Boots a device.
CONTINUE	Begins program execution at the address stored in the program counter (PC).
ONESTEP	Executes the program instruction at the address pointed to by the program counter (PC) and displays status information.

Related Messages

None

Examples

1. Start processor executing at address 398F0.

   SCM> **S 398F0** ⤶

2. Start processor executing at address 398F0 with a trace count of 3.

   SCM> **S 398F0 3** ⤶

```
Trace        000398F0       5555FFFF xor.u     r10 r21 $FFFF
Trace        000398F4       00000000 xmen.bu   r0 r0 $0
Trace        000398F8       00000000 xmen.bu   r0 r0 $0

PSR      XPC      DCSH      DMMU   ICSH  IMMU
A0000000 000398FA Y         Y        N     N
```

TRAP
Views or inserts breakpoints.

TRAP *[address] ...*

Description

Insert a breakpoint at the specified address. You can insert up to 2010 breakpoints. The command does not allow duplicate breakpoints.

With no argument, **TRAP** displays a list of all current breakpoints.

NOTE: The SCM implements breakpoints with the trap exception; hence, the command names **TRAP** and **UNTRAP**.

Arguments

[address]	Memory location at which you want to insert a breakpoint.
...	The memory location of each subsequent breakpoint (when you insert more than one breakpoint per command line).

Related Commands

UNTRAP	Deletes breakpoint(s)

Related Messages

None

Examples

1. Insert a breakpoint at addresses 5000 and 77000.

```
SCM> T 5000 77000 ↵

Breakpoint # 1 - 00005000  Set
Breakpoint # 2 - 00077000  Set
```

2. Display the current list of breakpoints and their addresses.

```
SCM> T ↵

Breakpoint # 1 - 00005000
Breakpoint # 2 - 00077000
```

No response indicates there are no current breakpoints.

UNTRAP
Deletes breakpoints.

UNTRAP *[breakpoint]* ...

Description

The **UNTRAP** command removes the breakpoint number(s) specified, or removes all current breakpoints with no argument. To view current breakpoint numbers, use the **TRAP** command.

Arguments

breakpoint	Number of the breakpoint you want to remove. (A decimal number 1 through 20.)
...	Numbers of each additional breakpoint number you want to remove.

Related Commands

TRAP	Inserts breakpoint at specified address, or displays current breakpoint numbers.

Related Messages

Invalid breakpoint

Examples

1. Find the three current breakpoints; then remove two of them.

   SCM> **T** ⤶

   ```
   Breakpoint # 1 - 00005000
   Breakpoint # 2 - 00077000
   Breakpoint # 3 - 000C6001
   ```

   SCM> **U 1 3** ⤶

   ```
   Breakpoint # 1 deleted
   Breakpoint # 3 deleted
   ```

2. Remove all current breakpoints.

   SCM> **U** ⤶

VIEW
Displays a range of memory.

VIEW *beg–addr [end–addr]*

Description

The **VIEW** command displays the contents of a specified block of contiguous memory. If you use the command with only a beginning address, the SCM displays all of memory from the specified address to the top of memory address.

NOTE: There are no restrictions to areas of memory you can specify. Viewing large blocks of memory could take hours, or cause memory exceptions. Use the Ctrl–C sequence to exit before completing the operation.

From left to right, the system displays the logical memory address, the physical memory address (only when MMU is enabled), the contents of the address, and the instruction mnemonic.

Arguments

beg–addr	First memory address in the range you want to view.
[end–addr]	Last memory address in the range you want to view.

Related commands

INITIALIZE	Writes data into a range of memory
LOCATE	Searches memory (or a range optionally specified) to locate the specified data pattern.
MOVE	Duplicates the contents of a specified range of memory and relocates the copy to a specified destination.
WRITE	Writes data to a single memory location.

Examples

View the contents of a block of contiguous memory beginning at location 0 and ending at location 00000010.

SCM> **V 0 10** ⤶

```
Memory 00000000  /  5555FFFF - xor.u    r10 r21 $FFFF
Memory 00000004  /  5555FFFF - xor.u    r10 r21 $FFFF
Memory 00000008  /  5555fFFF - xor.u    r10 r21 $FFFF
Memory 0000000C  /  5555FFFF - Xor.u    r10 r21 $FFFF
Memory 00000010  /  5555FFFF - xor.u    r10 r21 $FFFF
```

WRITE
Inserts data in one memory location.

WRITE *[H] [Q] address data*

Description

The **WRITE** command stores data in a specified memory address without reading its contents first.

CAUTION: *This command executes immediately; you cannot exit using the Ctrl–C sequence. There are no restrictions to areas of memory to which you can write data. Overwriting data in system control registers or NOVRAM locations could halt your system or destroy necessary data.*

Arguments

[H]	Specify 16–bit (half–word) data and address arguments.
[Q]	Specify 8–bit (quarter–bit, or one byte) data and address arguments.
address	Memory location to write new data.
data	Hexadecimal value or assembler command to write to memory address.

Related Commands

EXAMINE	Displays and optionally changes specified registers or memory locations.

Related Messages

None

Examples

1. Write 32 bits (one word) of data (01234567) in memory address 800.

   SCM> **W 800 01234567** ⤶

2. Write 16 bits (one half–word) of data (1234) in memory address 800.

   SCM> **W H 800 1234** ⤶

3. Write 8 bits (one byte) of data (12) in memory address 800.

   SCM> **W Q 800 12** ⤶ or SCM> **W B 800 12** ⤶

ZLOADER

Starts the s–record load utility.

ZLOADER

Description

The **ZLOADER** command starts a loader utility program that downloads files in Motorola s–record format (S3/S7 type) into memory from a device connected to the system console serial port. The SCM recognizes s–records only after you use the **ZLOADER** command.

An s–record is a file that contains an unlimited number of records; each record has a maximum of 256 bytes. A server machine uses s–records to download an entire file serially. The loader utility copies s–records in from the serial port into system memory on the local (client) system. The SCM reads information appended to individual s–records, stores each record at locations specified in the s–record header, and then verifies checksums. Information appended to the last s–record notifies the SCM when the entire file has been sent.

For this command to function, you must configure your system to use an s–record utility and a server machine must be sending s–record files. If you are not familiar with your system's s–record loader utility, if your system is not configured to receive s–records, or if the server is not sending, press the Ctrl–C sequence to exit from the **ZLOADER** command.

NOTE: Your system will pause indefinitely until it receives the last s–record in a file, or until you execute a Ctrl–C command sequence to exit to the SCM prompt.

Arguments

None

Related Commands

None

Related Messages

`DLL Started`	DLL (downline loader) is an s–record load utility
`DLL Done`	

Examples

Execute utility program to receive s–record files from a server system.

SCM> **Z** ⤶

End of Chapter

Appendix A
Specifying a Boot Path

This appendix describes the format for specifying boot devices and file paths when you require them as arguments to the **BOOT** command or when you specify your system's *default* boot device argument in the Change Boot Parameters menu. Refer to Chapter 3 for information about using the **BOOT** command and to Chapter 2 for information about setting, viewing, or changing your system's default boot paths in the Change Boot Parameters menu.

Whenever you identify a peripheral device, you provide a *device specification*, a software descriptor that uniquely identifies that device. A *boot path* is a device specification for a bootable device and, optionally, a second software descriptor that points to an executable image on the booted media called the *file path*. The SCM passes the file path to the bootstrap program *after* loading the bootstrap from the boot device.

Together, the boot device specification and the optional file path comprise a full boot path; they represent a first and second stage boot process. Figure A–1 shows their combined formats. A full SCM boot path contains a maximum of 80 characters.

device([controller][,unit][,file#])[logical–disk–name:][/dir...]/file

Boot device specification (first stage boot)

File path (second stage boot)

Figure A–1 Full Boot Path Format

NOTE: A LAN boot format is slightly different; refer to the section later in this appendix, "Booting Over a LAN."

The first part of this appendix explains the full boot path format for standard boot devices in detail. Tables A–1 and A–2 provide a quick reference of boot path specifications for standard AViiON system devices. Latter sections describe how to specify a file path argument, a LAN boot argument, and a nonstandard boot device argument.

Identifying and Specifying Standard Boot Devices

To identify a peripheral device, you must first identify the controller that manages it. Since all AViiON computers use memory–mapped I/O, device drivers refer to a memory address when accessing a device controller. Device specifications include that memory address, either explicitly or via device tables. You receive devices supplied by Data General preconfigured to default memory–mapped I/O addresses; you do not need to explicitly name the addresses. In this appendix, we refer to these devices as *standard* boot devices.

NOTE: The distinction between *standard* and *nonstandard* peripheral devices in this appendix does *not* refer to whether a device is considered to be industry–standard.

This section describes how to identify and specify standard boot devices currently available for use with AViiON systems, listed in Tables A–1 and A–2. Table A–4 contains the default memory–mapped I/O address for these standard devices. If you configure devices that are *not* listed in Tables A–1 or A–2, if your devices are configured at I/O addresses *not* listed in Table A–4, or if your operating system's device drivers do not follow SCM conventions, the device specifications described in this section do not apply. In this appendix, we refer to these devices as *nonstandard* boot devices. The last section of this appendix describes a general format for specifying nonstandard devices.

Some AViiON computers use a VMEbus interface for peripheral I/O, some use an integrated SCSI bus to manage peripherals; still other AViiON models use *both* an integrated SCSI and a VME bus. Table A–1 provides valid boot device specifications for standard devices managed by an integrated SCSI or Ethernet controller. Table A–2 provides valid boot device specifications for standard devices connected to a VMEbus (either directly or through a SCSI adapter). In this appendix we distinguish between VME *controllers*, directly addressable on the VMEbus, and *adapters*, connected to a SCSI bus.

Table A–1 Specifications for Standard Integrated Boot Devices

Device	Specification
First disk on integrated SCSI controller	**sd(insc(0),0)**
Second disk on integrated SCSI controller	**sd(insc(0),1)**
Third disk on integrated SCSI controller	**sd(insc(0),2)**
Fourth disk on integrated SCSI controller	**sd(insc(0),3)**
First tape drive on integrated SCSI controller	**st(insc(0),4)**
Second tape drive on integrated SCSI controller	**st(insc(0),5)**
Third tape drive on integrated SCSI controller	**st(insc(0),6)**
Integrated Ethernet LAN	**inen( )**

Table A–2 Specifications for Standard VME Boot Devices

Device	Specification
First tape drive on *first* Ciprico SCSI adapter	**st(cisc( ),4)**
Second tape drive on first Ciprico SCSI adapter	**st(cisc( ),5)**
Third tape drive on first Ciprico SCSI adapter	**st(cisc( ),6)**
First disk on *second* Ciprico SCSI adapter	**sd(cisc(1),0)**
Second disk on second Ciprico SCSI adapter	**sd(cisc(1),1)**
Third disk on second Ciprico SCSI adapter	**sd(cisc(1),2)**
Fourth disk on second Ciprico SCSI adapter	**sd(cisc(1),3)**
First VME (Interphase Hawk) Ethernet LAN	**hken(0)**
Second VME (Interphase Hawk) Ethernet LAN	**hken(1)**
First ESDI disk on *first* Ciprico controller	**cied(0,0**
Second ESDI disk on first Ciprico controller	**cied(0,1)**
Third ESDI disk on first Ciprico controller	**cied(0,2)**
Fourth ESDI disk on first Ciprico controller	**cied(0,3)**
First SMD disk on *first* Ciprico controller	**cimd(0,0)**
Second SMD disk on first Ciprico controller	**cimd(0,1)**
Third SMD disk on first Ciprico controller	**cimd(0,2)**
Fourth disk on first SMD Ciprico controller	**cimd(0,3)**
First ESDI disk on *second* Ciprico controller	**cied(1,0)**
Second ESDI disk on second Ciprico controller	**cied(1,1)**
Third ESDI disk on second Ciprico controller	**cied(1,2)**
Fourth ESDI disk on second Ciprico controller	**cied(1,3)**
First SMD disk on *second* Ciprico controller	**cimd(1,0)**
Second SMD disk on second Ciprico controller	**cimd(1,1)**
Third SMD disk on second Ciprico controller	**cimd(1,2)**
Fourth SMD disk on second Ciprico controller	**cimd(1,3)**
First ESDI disk on *third* Ciprico controller	**cied(2,0)**
Second ESDI disk on third Ciprico controller	**cied(2,1)**
Third ESDI disk on third Ciprico controller	**cied(2,2)**
Fourth SMD disk on third Ciprico controller	**cied(2,3)**
First SMD disk on *third* Ciprico controller	**cimd(2,0)**
Second SMD disk on third Ciprico controller	**cimd(2,1)**
Third SMD disk on third Ciprico controller	**cimd(2,2)**
Fourth SMD disk on third Ciprico controller	**cimd(2,3)**

NOTE: Each Ciprico controller manages *either* ESDI or SMD disks; not both.

Understanding the Device Specification

This section describes in detail the format for specifying a boot path. Each boot device specification consists of a device mnemonic that identifies the device type and names the device driver, plus three optional parameters that provide additional information to fully specify that device.

A standard boot device includes the device mnenomic (*dev*) followed by parentheses. Mnemonics are always in lowercase type. Controller (*cntrl*), unit (*unit*), and file number (*file#*) parameters are included within the parenthesis, separated by a comma or a space, according the following format:

dev([cntrl][,unit][,file#])

Table A–3 lists standard boot devices with the mnemonic names used by their drivers and definitions for each of the three driver parameters. All three device parameters use zero–based logical numbering; when you use 0 for a parameter you specify the *first* value for that parameter. Device drivers interpret missing parameters as 0, so when you omit any parameter you are specifying its first value. Even when all three device parameters are 0, you *must* include both the open and close parentheses after the device mnemonic (no space between them). The minimum specification for any boot device is **dev()**; this is equivalent to **dev(0,0,0)**.

Table A–3 Standard Boot Path Components

Device Mnemonic		Device Parameters		
	dev	cntrl	unit	file#
SCSI disk	sd	SCSI adapter[1]	SCSI ID[2]	0
SCSI tape drive	st	SCSI adapter[1]	SCSI ID[2]	Tape file number
Ciprico ESDI disk	cied	Controller[1]	Unit number[3]	0
Ciprico SMD disk	cimd	Controller[1]	Unit number[3]	0
VME Ethernet LAN	hken	0	0	0
Integrated Ethernet	inen	0	0	0

1 A logical number, mnemonic, or I/O address. See Table A–4 for values.
2 An integer 0 through 6. See Table A–5 for values.
3 An integer 0 through 3.

First Parameter: Specifying a Controller

The first parameter (*cntrl*) indicates the device controller. You can specify the controller by its device mnemonic or by its memory–mapped I/O address, as seen in Table A–4. Which value you use depends upon your system configuration. You can *always* use a standard controller's I/O address or mnemonic, but in most cases can substitute the less complicated logical controller number. If you name the controller by specifying its mnemonic value, the controller must be accessible by the memory–mapped I/O address listed in Table A–4.

For example, **sd(cisc(0),0)** specifies the first disk unit on the first Ciprico SCSI controller, while **sd(insc(0),0)** specifies the first disk unit on the first integrated SCSI controller. In both cases, the disk hardware is jumpered according to its factory configuration (the default controller address listed in Table A–4 and the default SCSI ID number listed in Table A–5).

To *explicitly* specify a controller, you enter its I/O address as the first parameter. For example, rather than **sd(cisc(0),0)** the specification is **sd(fffff300,0)**. You should only need to do this if you have nonstandard devices configured with your system; refer to the "Specifying Nonstandard Boot Devices" section of this appendix.

Table A–4 Controller (*cntrl*) Parameter Values

To specify this controller:	Enter one of these values: Mnemonic	I/O Address
SCSI Adapters		
Integrated SCSI adapter	**insc( )**	N/A
First Ciprico SCSI adapter	**cisc(0)**	**fffff300**
Second Ciprico SCSI adapter	**cisc(1)**	**fffff500**
Third Ciprico SCSI adapter	**cisc(2)**	**fffff700**
Fourth Ciprico SCSI adapter	**cisc(3)**	**fffff900**
Ciprico Controllers[1]		
First Ciprico ESDI disk controller	**cied(0)**	**ffffef00**
Second Ciprico ESDI disk controller	**cied(1)**	**fffff100**
Third Ciprico ESDI disk controller	**cied(2)**	**fffffb00**
Fourth Ciprico ESDI disk controller	**cied(3)**	**fffffd00**
First Ciprico SMD disk controller	**cimd(0)**	**ffffef00**
Second Ciprico SMD disk controller	**cimd(1)**	**fffff100**
Third Ciprico SMD disk controller	**cimd(2)**	**fffffb00**
Fourth Ciprico SMD disk controller	**cimd(3)**	**fffffd00**
LAN Controllers		
First Interphase Hawk Ethernet	**hken(0)**	**ffff4000** and **5590000**
Second Interphase Hawk Ethernet	**hken(1)**	**ffff5000** and **5598000**
Integrated Ethernet	**inen( )**	N/A

[1] Your Ciprico controllers support *either* ESDI *or* SMD disks; not both.

Second Parameter: Specifying a Unit

The second parameter (*unit*) specifies the logical number of the device on the specified controller or adapter. As shown in Table A–3, each controller or adapter has a specific type of unit parameter. For example, the *unit* parameter for a SCSI device is its SCSI ID number; the *unit* parameter for an ESDI device is its unit number. Logical numbering for the *unit* parameter begins at 0 and is determined by configuration jumpers. By omitting the second parameter, you specify the first unit for that device type (i.e., SCSI ID 0 or ESDI unit 0).

Table A–5 defines the SCSI ID numbering scheme used for AViiON systems.

Table A–5 Default SCSI Device ID Numbers

Drive	ID Number
First disk[1]	0
Second disk	1
Third disk	2
First diskette[2] (LUN 0)	3
Second diskette (LUN 1)	3
First cartridge tape	4
Second cartridge tape	5
Third cartridge tape	6

1 Hard disk drives include Winchester and CD–ROM types.

2 A floppy diskette is not a bootable device. The SCSI ID of a diskette drive is set on a SCSI adapter board, not on the drive. If you have more than one diskette drive managed by the same SCSI adapter board, the drives have the same SCSI ID number. The LUN (Logical Unit Number) differentiates drives managed by the same SCSI adapter.

Third Parameter: Specifying a Partition

You rarely use the third parameter (*file#*); it supplies additional information when the second parameter does not uniquely identify the bootstrap file. Currently, this parameter is only valid to specify a file number for a tape device. By omitting the third parameter, you specify file number 0 on the tape in the specified tape drive.

Specifying a Second–Stage Boot File

After the SCM boot (the first–stage boot) has completed, any additional text in the boot path is passed to the booted program for further processing. Typically, the booted program is the operating system second–stage bootstrap, which uses the additional text to bring up and properly initialize your operating system. You can use this second stage, or *file path*, argument to specify a particular program or program parameter (such as run level) to come up in the automatic boot sequence.

To specify a file path within a boot path argument, append its specification just after the device specification (do not type a space after the right parenthesis). Include the name of the executable image (*file*), preceded by the directory path to that file (*dir...*) and the name of the logical disk (*ld*) which contains the file. Include a colon after the logical disk name (:) and separate each directory partition with a slash, according the following format:

[ld:][/dir...]/file

For example, the file path **usr:/stand/diags** refers to the AViiON System Diagnostics program file, called **diags**, located in the directory **stand** on the logical disk **usr**. To boot **diags** from the disk at SCSI ID #0 connected to a workstation, use the following full boot path:

sd(insc(),0)usr:/stand/diags

You can omit the boot device specification and use the file path argument alone when booting from the default boot device (the system boot path stored in the Change Boot Parameters menu). The logical disk name and directory path are also optional. When you omit them, the file path starts from the root (**root:**) by default. The following is sufficient to boot the executable image **file** in the root directory of the default boot device:

SCM> **b /file**

When booting from tape, you do not specify a file path at all; instead, use the third part of the boot path to specify the tape file number. For example, if **diags** is file #0 on a 150 megabyte QIC cartridge tape (SCSI ID #4), the full boot path is **st(insc(),4,0)**.

Booting over a LAN

There are several exceptions to the formats for identifying a boot device and file path when booting as a client to a local area network server. When you boot over a LAN, the bootstrap device is your computer's LAN controller. You do not use the device parameters described in Table A–3. In the second–stage bootstrap argument, you optionally specify the Internet address of the server system with the file path to the executable image on the server's root ldu (logical disk). Figure A–2 shows the format for specifying a LAN boot path argument:

LAN–controller–mnemonic()[Server–Internet–address:][/dir...][/file]

Boot device specification (first stage boot) — **File path (second stage boot)**

Figure A–2 LAN Boot Format

For example, the following boot path specifies the AViiON System Diagnostics file **diags** in the **stand** directory on the root disk of a server at Internet address **128.111.2.3** on the first Interphase Hawk Ethernet LAN:

hken(0)128.111.2.3:/stand/diags

You can also specify the server's Internet address *without* specifying the file path ,for example, **hken(0)128.111.2.3:** The operating system on the server system contains a boot parameters file that contains the default boot file for each configured client. Refer to your operating system and network administration documentation for information about these boot parameters.

When you omit the file path argument in a LAN boot, the SCM probes the LAN for any server that recognizes your computer's Ethernet address and then boots the default boot file. Therefore, the minimum LAN boot path is simply your LAN controller's device driver name: **hken(0), hken(1)** ,or **inen( )**.

Specifying Nonstandard Boot Devices

Without the default parameter values used for standard devices, a boot path specification is even more complex. The extended format for an SCM boot path includes information you do not need when specifying standard boot devices. (Your operating system may store device information in this extended format, however.)

There are several reasons for using the extended, nonstandard–device format for boot device specifications. You may need to configure a device at a different I/O address than the current Data General convention, listed in Table NO TAG; or, you may have more devices configured than can be named by the current defaults. You may want to change device configurations to accommodate your operating system or to configure additional devices not supplied by Data General.

Whenever you use nonstandard boot devices, you need to set configuration jumpers on the device and include its I/O address as the first parameter (*param1*) in device specifications.

The following is the format for an expanded or nonstandard boot path argument when booting from disk:

dev[@vector]([param1][,param2][,param3])[logical–disk–name:][/dir...][/file]

The following is the format for an expanded or nonstandard boot path argument when booting from tape:

dev[@vector]([param1][,param2][,param3])

The following is the format for an expanded or nonstandard boot path argument when booting over a LAN:

dev[@vector]([param1][,param2][,param3])[Internet–address:][/dir...][/file]

Where	Means
dev0	The name of the device driver that supports the disk or tape device, or the LAN controller.
[@vector]	The interrupt vector number.
	NOTE: The DG/UX operating systems considers the vector number to be the device code.
[.param1]	The memory–mapped I/O address of the controller.
	NOTE: Insert one comma to separate each parameter.
[.param2] and *[.param3]*	Additional parameters, defined by the device driver, to fully identify the device.
[logical–disk–name]	The name of the logical disk, defined by the operating system, which contains the second–stage bootstrap file.
[Internet–address]	The unique identifier for a network server in a TCP/IP LAN.
[/dir...]	The directory path to the second–stage bootstrap file.
[/file]	The name of the executable second–stage bootstrap file.

For example, the default specification for a Ciprico SCSI disk with default parameters (the first unit on the first controller) is **sd(cisc(0),0)**. The disk hardware is jumpered according to its factory configuration and the default controller address listed in Table A–2. To explicitly specify this same disk, the device specification is **sd(cisc(fffff300),0)**. To specify the first Ciprico SCSI disk managed by a *different* controller at nonstandard address FFFF0016, the device specification is **sd(cisc(ffff0016),0)**.

In another example: **cied@70(ffffe000,1)usr:/ops/program** specifies an executable image called **program** in directory **ops** located on the logical disk **usr**, the second physical disk connected to a Ciprico ESDI controller (identified by the device mnemonic **cied**). This example assumes that the device driver **cied** is present and that the controller board is jumpered at the nonstandard base address FFFFE000; its device code (interrupt vector) is 70_{16}.

End of Appendix

Appendix B
System Programming and Debugging Tools

This appendix lists SCM tools that are available for operating system and diagnostic program development. For detailed information about system PROM or address space and about programming in the AViiON RISC–based environment, refer to the hardware programming manuals for your AViiON model, listed in the Preface.

System Calls

The SCM supports a standard set of system calls that use CPU registers accessible in Programmable Read Only Memory (PROM) vectors. Programs can pass control to the SCM using these optional system calls.

Operating system software may need to support the SCM system calls for certain value–added functions. The SCM currently provides the following services to the operating system via system calls:

- Access to standard input/output devices.
- System configuration information.
- Panic and error reporting.

Software accesses SCM system calls through vectors in the boot PROM vector space. A program must do the following to access the SCM system calls:

1. Set CR7, the Vector Base Register (VBR), to the PROM VBR. If this changes the value of CR7, software must save the changes to copy back later. The VBR defaults to the PROM values after powerup.
2. Load R9 and other argument–specified registers with the offset value defined in Table B–1 (the values are hexadecimal unless specified otherwise).
3. Execute the following trap instruction:

    tb0 0,R0,496

Table B–1 on the following two pages lists and describes all SCM system calls. Your AViiON model may not support every SCM system call; refer to your hardware programming manual (listed in the Preface) for specific information about supported system calls.

Table B–1 SCM System Calls

System Call	Argument(s)	Data Returned	Function
.BANNER	R9=113	R2=Pointer to string	Returns pointer to system banner string.
.CHAR	R9=0	R2(LSB)=ASCII character	Waits for an ASCII character from the default input port, reads it, and returns the character in the least significant byte of Register 2. (A null indicates a break.)
.CHFLOW	R9=116 R2=0 or R2 < > 0 R3=Value	R2=Flow control flag	Reads or writes the character flow control (XON/XOFF) flag. If R2 = 0 initially, then R2 will contain the current flag value. If R2 < > 0 initially, then stores value from R3. A value of F816 indicates flow control enabled; any other value indicates disabled.
.CHKSUM	R2=Pointer to data R3=Byte count R9=68	R2=Checksum	Performs data checksum test and returns the value in R2. (Adds all the bytes and complements the result.)
.CHSTAT	R9=5	R2(LSB)=default input status	Polls the standard input port for character status, and returns this value in the least significant byte of R2.
.COMMID	R9=114	R2=Pointer to address	Returns pointer to Ethernet address.
.CPUID	R9=102	R2=CPU ID	Returns the CPU ID.
.DATETIME	R2=0 for read or 1 to set R3=Pointer to date/time R9=200	R2=0 or 1 for error if values are invalid.	Reads or sets the date and time.
.GDMP	R9=105 R2=0	R2=Pointer	Reads video timing parameters into SPAD buffer, and returns pointer as byte–packed data in R2. If the original value of R2 < > 0, writes byte–packed data pointed to by R2 into BBSRAM.
.GMT	R9=205 R3=New value R2=0 for read or 1 to set	R2=GMT offset value	Reads or writes the GMT offset value (offset from Universal Time). Valid range is –720 to +720 minutes.
.GTLINE	R2=32–bit string buffer address R9=2	R2=String length	Reads a character string of 256 characters or less from the standard input port, echoes them, and places the character string in the buffer address in R2. Terminator characters are: \n = New Line, \l = Carriage Return, and \f = form feed. Supports SCM screen edit control functions (see Table 3–1).
.HALT	R9=63	None	Halts the user program and enters the SCM.
.INVALID	R9=112 R2=JP# or –1	None	Invalidates the instruction cache (Icache). If R2 = a JP number, then only that JP Icache is invalidated. If R2 = –1, then all JP Icaches are invalidated.
.JPSTART	R2=JP# to start R3=Starting address R9=100	R2=Status	Starts another processor (JP#) after an initial boot (used only in multiprocessor systems). The status returned to R2 is 0 Start successful 1 Illegal or missing JP 2 Single JP configuration 3 JP not halted 4 JP does not respond

(continued)

NOTE: A value of –1 returned to R2 indicates an error for any SCM system call.

Table B–1 SCM System Calls

System Call	Argument(s)	Data Returned	Function
.KBLAN	R9=106	R2=Language	Returns language code to R2. The codes and languages are 1 U.S. English 6 Spanish 2 German 7 Swiss 3 U.K. English 8 Italian 4 French 9 Japanese 5 Swedish
.MSIZE	R9=103 R2=0 or R2 < > 0	R2=Top of memory	Returns top of memory to R2. If R2 = 0 initially, then R2 will contain top of physical memory. If R2 < > 0 initially, then R2 will contain top of user memory.
.NBLOCAL	R9=115 R2=0 or R2 < > 0 R3=Value	R2=LAN port number	Reads or writes the LAN port number. If R2 = 0 initially, then R2 will contain the LAN port number. If R2 < > 0 initially, then stores value from R3.
.OCHAR	R9=20 R2(LSB)=ASCII character	R2=0	Prints the value in the least significant byte of R2 to the standard output device.
.OCRLF	R9=26	R2=0	Prints a Carriage Return/line feed to the standard output device.
.PRINTER	R9=117	R2=Printer type	Returns printer type to R2. R2 = 0 is Centronics; R2 < > 0 is Data Products.
.POLLKEY	R9=5	R2=Key hit	Returns an indication of whether or not a key was pressed. If R2 = 0, no key was pressed. If R2 < > 0, a key was pressed.
.PTLINE	R9=21 R2=32–bit address of string	R2=0	Prints the character string pointed to by the address in R2 to the standard output device. Does not return until it encounters the null terminator in the string. Note that this call allows 5 additional arguments and uses the C **printf** characteristics.
.REBOOT	R9=101 R2=Pointer to boot path	None	Resets and reinitializes the system, initializes the boot time registers, and enters the Boot menu. If R2 = 0, the call uses the default boot path. If R2 < > 0, the call uses the pointer in R2.
.REVNUM	R9=104	R2=Revision number	Returns PROM revision to R2 in the format: bit 31 (if 1), engineering revision; bits 30–16, major revision number; bits 15–0, minor revision number. For example, 80050002 = Rev E05.02 30000 = Rev 3.0
.STDIO	R9=70	R2=I/O device number	Returns the standard input and output ports. Device number values are 0 Serial input and output 1 Serial input/serial and graphics output 2 Keyboard input/graphics output
.SYSID	R9=31	R2=System ID	Returns the unique system identification number for the machine.
.TECW	R9=108 R2=0 or < > 0 R3=New ECW value	R2=ECW	Returns or sets Environmental Control Word (ECW). If R2=0, returns ECW to R2. If R2 < > 0, writes R3 value to ECW. Table B–4 lists the ECW bit values, functions, and default states at powerup.

(concluded)

Subroutines

In addition to system calls, the SCM supports hardwired entry points to the subroutines in Table B–2 (accessible with a **jsr** instruction containing the appropriate entry point). See the hardware programming manual for your AViiON model (listed in the Preface) for more information.

Table B–2 SCM Subroutines

Entry Point (Hex)	Subroutine	Argument	Description
1000	putchar to stdio	R2=char	Outputs the character in R2.
1004	getchar from stdio	R2=char	Returns a character to R2.
1008	KBD_reset	R2=0	Performs a keyboard hard reset.
100C	console_reset	R2=0	Performs hard reset to the system console or graphics display monitor, if present.
1010	GDM_load_fonts		Reloads the fonts for the graphics display monitor, if present.
1014	GDM_putchar	R2=char	Outputs the character in R2 to the graphics display monitor, if present.

Environment Control Word (ECW)

Enabling or disabling bits in the Environment Control Word (ECW) allows you to alter test parameters while debugging programs. You can read the current value of the ECW or write to it with the **.TECW** system call. Refer to Table B–1 for information about using the **.TECW** system call. Table B–3 on the following page describes the function of each ECW bit. A bit's function is enabled when the value is 1; a bit is disabled when the value is 0.

You can quickly view or change the state of ECW bits using a keyboard control sequence. Type **Ctrl–P** and press New Line to see the current status of bits displayed as one hexadecimal word. You can set particular bits by typing **Ctrl–P** followed by the bit number you want to toggle; then press New Line. For example, with bit 8, "Report all," currently disabled, enter **Ctrl–P**, type **8**, and then press New Line to toggle bit 8 and enable verbose message reporting to the system console.

Refer to the section "Change Test Parameters Menu" in Chapter 2 for information about enabling or disabling individual bits in the ECW using an SCM menu.

Table B–3 Environmental Control Word (ECW) Contents

Bit	Function	State at Powerup
0	Reserved	Disabled
1	Loop on error 0 Disables testing when program encounters an error. 1 Continues testing when program encounters an error.	Enabled
2	Output to console 1 Directs program output to the system console	Enabled
3	Percent failure 0 Disables reporting of this error. 1 Enables reporting of percent of errors after looping (errors per total number of loops). Note that bit 1 (loop on error) must also be enabled.	Disabled
4	Print pass messages 0 Disables printing of message. 1 Enables printing of messages to the system console after each test pass completes.	Enabled
5	Output to printer 0 Disables output to printer. 1 Enables program output to the default printer port.	Disabled
6	Disassembler 0 Disables display. 1 Enables displaying an additional output field that contains the mnemonics of memory address contents.	Enabled
7	Print subtest message 0 Disables printing message. 1 Enables printing subtest messages to the system console.	Enabled
8	Report all 0 Print brief messages to the system console. 1 Print verbose messages to the system console.	Disabled
9	Halt on error 1 Enables halting the program after an error and returning the SCM prompt.	Disabled
10	Enable error logging 0 Disables error logging. 1 Enables recording all errors in system error log.	Disabled
11	Continue on exception 0 Enter SCM when program encounters an exception. 1 Display an exception message and attempt to continue when program encounters an exception.	Disabled
12	Reserved	Disabled
13	Page mode 0 Disables page mode. 1 Enables displaying output on the system console one screen (page) at a time.	Disabled
14–31	Reserved	Disabled

End of Appendix

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

Z

TIPS ORDERING PROCEDURES

TO ORDER

1. An order can be placed with the TIPS group in two ways:
   a) MAIL ORDER – Use the order form on the opposite page and fill in all requested information. Be sure to include shipping charges and local sales tax. If applicable, write in your tax exempt number in the space provided on the order form.

   Send your order form with payment to: Data General Corporation
   ATTN: Educational Services/TIPS G155
   4400 Computer Drive
   Westboro, MA 01581–9973

   b) TELEPHONE – Call TIPS at (508) 870–1600 for all orders that will be charged by credit card or paid for by purchase orders over $50.00. Operators are available from 8:30 AM to 5:00 PM EST.

METHOD OF PAYMENT

2. As a customer, you have several payment options:
   a) Purchase Order – Minimum of $50. If ordering by mail, a hard copy of the purchase order must accompany order.
   b) Check or Money Order – Make payable to Data General Corporation.
   c) Credit Card – A minimum order of $20 is required for MasterCard or Visa orders.

SHIPPING

3. To determine the charge for UPS shipping and handling, check the total quantity of units in your order and refer to the following chart:

Total Quantity	Shipping & Handling Charge
1–4 Items	$5.00
5–10 Items	$8.00
11–40 Items	$10.00
41–200 Items	$30.00
Over 200 Items	$100.00

If overnight or second day shipment is desired, this information should be indicated on the order form. A separate charge will be determined at time of shipment and added to your bill.

VOLUME DISCOUNTS

4. The TIPS discount schedule is based upon the total value of the order.

Order Amount	Discount
$0–$149.99	0%
$150–$499.99	10%
Over $500	20%

TERMS AND CONDITIONS

5. Read the TIPS terms and conditions on the reverse side of the order form carefully. These must be adhered to at all times.

DELIVERY

6. Allow at least two weeks for delivery.

RETURNS

7. Items ordered through the TIPS catalog may not be returned for credit.
8. Order discrepancies must be reported within 15 days of shipment date. Contact your TIPS Administrator at (508) 870–1600 to notify the TIPS department of any problems.

INTERNATIONAL ORDERS

9. Customers outside of the United States must obtain documentation from their local Data General Subsidiary or Representative. Any TIPS orders received by Data General U.S. Headquarters will be forwarded to the appropriate DG Subsidiary or Representative for processing.

TIPS ORDER FORM

Mail To: Data General Corporation
Attn: Educational Services/TIPS G155
4400 Computer Drive
Westboro, MA 01581 - 9973

BILL TO:	SHIP TO: (No P.O. Boxes - Complete Only If Different Address)
COMPANY NAME ______	COMPANY NAME ______
ATTN: ______	ATTN: ______
ADDRESS ______	ADDRESS (NO PO BOXES) ______
CITY ______	CITY ______
STATE ______ ZIP ______	STATE ______ ZIP ______

Priority Code ____________________ (See label on back of catalog)

Authorized Signature of Buyer (Agrees to terms & conditions on reverse side)	Title	Date	Phone (Area Code)	Ext.

ORDER #	QTY	DESCRIPTION	UNIT PRICE	TOTAL PRICE

A SHIPPING & HANDLING

☐ UPS	ADD
1–4 Items	$5.00
5–10 Items	$8.0
11–40 Items	$10.00
41–200 Items	$30.00
200+ Items	$100.00

Check for faster delivery

Additional charge to be determined at time of shipment and added to your bill.

☐ UPS Blue Label (2 day shipping)
☐ Red Label (overnight shipping)

B VOLUME DISCOUNTS

Order Amount	Save
$0–$149.99	0%
$150–$499.99	10%
Over $500.00	20%

Tax Exempt # or Sales Tax (if applicable) ______________

ORDER TOTAL	
Less Discount See B	–
SUB TOTAL	
Your local* sales tax	+
Shipping and handling – See A	+
TOTAL – See C	

C PAYMENT METHOD

☐ Purchase Order Attached ($50 minimum)
P.O. number is_________. (Include hardcopy P.O.)
☐ Check or Money Order Enclosed
☐ Visa ☐ MasterCard ($20 minimum on credit cards)

Account Number ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ Expiration Date ☐☐☐☐

Authorized Signature ______________________________

(Credit card orders without signature and expiration date cannot be processed.)

THANK YOU FOR YOUR ORDER

PRICES SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.
PLEASE ALLOW 2 WEEKS FOR DELIVERY.
NO REFUNDS NO RETURNS.

* Data General is required by law to collect applicable sales or use tax on all purchases shipped to states where DG maintains a place of business, which covers all 50 states. Please include your local taxes when determining the total value of your order. If you are uncertain about the correct tax amount, please call 508–870–1600.

Form 702
Rev. 8/87

DATA GENERAL CORPORATION
TECHNICAL INFORMATION AND PUBLICATIONS SERVICE
TERMS AND CONDITIONS

Data General Corporation ("DGC") provides its Technical Information and Publications Service (TIPS) solely in accordance with the following terms and conditions and more specifically to the Customer signing the Educational Services TIPS Order Form. These terms and conditions apply to all orders, telephone, telex, or mail. By accepting these products the Customer accepts and agrees to be bound by these terms and conditions.

1. CUSTOMER CERTIFICATION
Customer hereby certifies that it is the owner or lessee of the DGC equipment and/or licensee/sub–licensee of the software which is the subject matter of the publication(s) ordered hereunder.

2. TAXES
Customer shall be responsible for all taxes, including taxes paid or payable by DGC for products or services supplied under this Agreement, exclusive of taxes based on DGC's net income, unless Customer provides written proof of exemption.

3. DATA AND PROPRIETARY RIGHTS
Portions of the publications and materials supplied under this Agreement are proprietary and will be so marked. Customer shall abide by such markings. DGC retains for itself exclusively all proprietary rights (including manufacturing rights) in and to all designs, engineering details and other data pertaining to the products described in such publication. Licensed software materials are provided pursuant to the terms and conditions of the Program License Agreement (PLA) between the Customer and DGC and such PLA is made a part of and incorporated into this Agreement by reference. A copyright notice on any data by itself does not constitute or evidence a publication or public disclosure.

4. LIMITED MEDIA WARRANTY
DGC warrants the CLI Macros media, provided by DGC to the Customer under this Agreement, against physical defects for a period of ninety (90) days from the date of shipment by DGC. DGC will replace defective media at no charge to you, provided it is returned postage prepaid to DGC within the ninety (90) day warranty period. This shall be your exclusive remedy and DGC's sole obligation and liability for defective media. This limited media warranty does not apply if the media has been damaged by accident, abuse or misuse.

5. DISCLAIMER OF WARRANTY
EXCEPT FOR THE LIMITED MEDIA WARRANTY NOTED ABOVE, DGC MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR PARTICULAR PURPOSE ON ANY OF THE PUBLICATIONS, CLI MACROS OR MATERIALS SUPPLIED HEREUNDER.

6. LIMITATION OF LIABILITY
A. CUSTOMER AGREES THAT DGC'S LIABILITY, IF ANY, FOR DAMAGES, INCLUDING BUT NOT LIMITED TO LIABILITY ARISING OUT OF CONTRACT, NEGLIGENCE, STRICT LIABILITY IN TORT OR WARRANTY SHALL NOT EXCEED THE CHARGES PAID BY CUSTOMER FOR THE PARTICULAR PUBLICATION OR CLI MACRO INVOLVED. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO CLAIMS FOR PERSONAL INJURY CAUSED SOLELY BY DGC'S NEGLIGENCE. OTHER THAN THE CHARGES REFERENCED HEREIN, IN NO EVENT SHALL DGC BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES WHATSOEVER, INCLUDING BUT NOT LIMITED TO LOST PROFITS AND DAMAGES RESULTING FROM LOSS OF USE, OR LOST DATA, OR DELIVERY DELAYS, EVEN IF DGC HAS BEEN ADVISED, KNEW OR SHOULD HAVE KNOWN OF THE POSSIBILITY THEREOF; OR FOR ANY CLAIM BY ANY THIRD PARTY.

B. ANY ACTION AGAINST DGC MUST BE COMMENCED WITHIN ONE (1) YEAR AFTER THE CAUSE OF ACTION ACCRUES.

7. GENERAL
A valid contract binding upon DGC will come into being only at the time of DGC's acceptance of the referenced Educational Services Order Form. Such contract is governed by the laws of the Commonwealth of Massachusetts, excluding its conflict of law rules. Such contract is not assignable. These terms and conditions constitute the entire agreement between the parties with respect to the subject matter hereof and supersedes all prior oral or written communications, agreements and understandings. These terms and conditions shall prevail notwithstanding any different, conflicting or additional terms and conditions which may appear on any order submitted by Customer. DGC hereby rejects all such different, conflicting, or additional terms.

8. IMPORTANT NOTICE REGARDING AOS/VS INTERNALS SERIES (ORDER #1865 & #1875)
Customer understands that information and material presented in the AOS/VS Internals Series documents may be specific to a particular revision of the product. Consequently user programs or systems based on this information and material may be revision–locked and may not function properly with prior or future revisions of the product. Therefore, Data General makes no representations as to the utility of this information and material beyond the current revision level which is the subject of the manual. Any use thereof by you or your company is at your own risk. Data General disclaims any liability arising from any such use and I and my company (Customer) hold Data General completely harmless therefrom.

Using the
AViiON®
System Control
Monitor (SCM)

014–001802–04

Cut here and insert in binder spine pocket

Data General

Data General Corporation, Westboro, Massachusetts 01580