SI505 RELEASE NOTES

The SI505 Release Notes describe System Industries' SI505 product release. SI505 is compatible with VAX/VMS Version 5.0 through Version 5.5. Changes to the system, new features, and restrictions are discussed. This information is for all users of SI505.

Refer to the *Software Modification User Guide VAX/VMS V5.0 Through V5.5* for more information on the SI505.

The following VAX/VMS releases are supported by SI505.

- V5.0-2
- V5.1
- V5.1-1
- V5.2
- V5.3
- V5.3-1
- V5.3-2
- V5.4
- V5.4-1
- V5.4-2
- V5.4-3
- A5.5
- V5.5

Changes from the VAX/VMS V.5 releases are highlighted below.

SI500

1. VMSINSTAL

   The System Industries software modifications are installed with the VMSINSTAL utility.

2. AUTOCONFIGURATION

   System Industries supplies the command procedure SI_AUTOCONFIG.COM that allows autoconfiguration of SI Q/UNIBUS devices on primary and secondary controllers at system boot time. The ability to autoconfigure SI devices at boot time allows both dual-ported and served-disk paths support in VAXclusters.

   SI recommends customer system upgrade to VAX/VMS V5.0-2 before running SI_AUTOCONFIG.COM. If VAX/VMS upgrades are installed later, SI_AUTOCONFIG.COM must be run again after the upgrade.

   The format of SYSGEN parameter USER3 has been modified. For more information, refer to the *Software Modification User Guide VAX/VMS V5.0 Through V5.5.*

22. When attempting to patch a version of VMS which is not supported by the patch kit, the user has the following options:
    a. Entering a new patch address from the user's terminal.
    b. Using the most recent patch addresses available.
    c. Skipping the patch and going on to the next one.
    d. Quitting the installation entirely.

CLUSTOR CI/M

23. If any member of a VAXCluster is accessing the shared CLUSTOR disks via a MSCP-type connection, **all** VAXes in the cluster **must** be configured to see the shared disks as DU device types. This is achieved by running the SI_AUTOCONFIG.COM command procedure as detailed in Chapter 5 of the *Software Modification User Guide.*

NOTE

When a number of CLUSTOR disks are shared by a group of VAXes in a VAXCluster, it is imperative that all the VAXes in the cluster "see" the shared disks as having the same name.

This is a requirement because disks accessed via a SCSI CI connection use the DEC DUDRIVER (MSCP disk driver). All disks accessed using this device driver have the name DU and this cannot be changed. Therefore, it is necessary to ensure that all VAXes which access these same shared disks using RM-type emulation also name the shared disks DU.

If this naming convention is not adhered to, there is a risk of corrupting or losing data held on the shared disks. Further information on this topic may be found in Chapter 6 of the *Software Modification User Guide.*

SI504B

24. VMS V5.4-3 is now supported.
25. SIBACKIT now functions correctly for all versions of VMS supported by this kit.
26. A problem which prevented the MicroVAX II boot feature from working with certain configurations has been corrected.

SI505

27. VMS V5.5 is now supported.

3. DIDRIVER

   Because of a naming conflict with Digital's DMB32 synchronous port driver, SIDRIVER is renamed DIDRIVER. When devices are autoconfigured (see note 2), disk names are assigned as DRxy. The device name DR was chosen for consistency with MASSBUS devices on other nodes in a VAXcluster.

4. DRDRIVER

   DRDRIVER is no longer patched by System Industries, but is supplied in full. It is a replacement for the DEC DRDRIVER.

5. MMDRIVER

   MMDRIVER contains VAX/VMS V.5 support, but not SMP support. This is because MMDRIVER is supported on non-VAXBI machines only.

6. ETHERNET Interfaces

   The DEQNA revision level must be at K3 or greater for Ethernet cluster configurations. DEBNT is not supported for VAXBI nodes. Instead DEBNT must be upgraded to the DEBNA interface.

SI501

7. AUTOSIZING

   The Autosizing feature has been added to both DRDRIVER and DIDRIVER. Using this feature, the drivers automatically determine disk drive sizes (geometry), instead of using a table of known drive sizes. This allows effortless support of newly released SI disk drives.

8. "A" Releases

   System Industries does not support DEC's special hardware releases of VAX/VMS (for example, V5.0-2A) with software modifications described in this manual. If the installation procedure detects a special hardware release of VAX/VMS, it issues a warning message, but allows the user to continue.

9. 86xx Support

   Support for VAX 86xx machines with more than one SBI has been added by use of the USER4 SYSGEN parameter. This optional parameter is for the rare case of multiple SBIs on the 86xx machines only.

10. Patch Output not Displayed

    The installation procedure and SI_AUTOCONFIG.COM no longer display the output of patch commands to the terminal.

11. TMDRIVER

    At the time of the SI501 release, DEC had released a BETA copy of TMDRIVER which fixes one of the many backup problems. The installation procedure supports patching the BETA release.

SI502

12. CLUSTOR/9900 MicroVAX II Boot Feature

    Support has been added for booting a MicroVAX II using a locally connected SI disk attached to a CLUSTOR or 9900 Controller. This feature is in addition to the already existing capability of booting an SI MSCP-compatible disk. This feature is supported on all VMS versions listed above and is supplied on TU58, RX50, and Magtape. The SYSGEN parameter MSCP_LOAD must be set to one. This ensures the system disk name is consistent throughout the Cluster.

13. The Patch Journal file (SIPATCH.JNL) now resides in SYS$UPDATE, not SYS$LOADABLE_IMAGES.

SI503

14. QBUS Boot Disk Name Change

    SIVMB has been changed to call the CLUSTOR/9900 boot device "DR" on UNIBUS systems. This allows the disk to be included in more VAXcluster configurations.

15. Manual Menu Change

    The VMS/INSTAL "Options D" qualifier is no longer used to invoke the manual installation menu. Instead, the user is prompted during the installation.

SI504

16. USER3 Change

    The setting of the USER3 SYSGEN parameter will no longer affect the allocation class assigned to DEC RM devices. Only devices attached to SI controllers will have their allocation class derived from the USER3 parameter.

SI504A

17. VMS V5.4-1 and V5.4-2 are now supported.

18. The procedure SIBACKIT (used for building stand-alone BACKUP kits) has been fixed to prevent a loop condition occurring while building kits on RX01 floppy disks.

19. A bug in MMDRIVER (UNIBUS tape) which prevented DCL COPY from working correctly has been fixed.

20. A bug in SIVMB which prevented a system from booting if a CI was present has been fixed.

21. A more obvious error message is now generated if a patch fails to apply correctly.

SOFTWARE MODIFICATION USER GUIDE FOR VAX/VMS V5.0 THROUGH V5.5

PB9900-9070-008

® SYSTEM INDUSTRIES

1855 Barber Lane
Milpitas, CA 95035, USA.

PROPRIETARY STATEMENT

REVISION RECORD

REVISION NUMBER	DATE	DESCRIPTION	EO NUMBER
A	11/08/88	Initial Release (SI500)	5479
B	04/28/89	VMS 5.1 Release (SI501)	5890
003	11/03/89	VMS 5.2 Release (SI502). Incorporate new publications numbering system	6250
004	03/09/90	VMS 5.3 Release (SI503)	6424
005	11/14/90	VMS 5.4 Release (SI504)	6666
006	04/26/91	VMS 5.4-1, 5.4-2 Release (SI504)	6833
007	12/05/91	VMS 5.4-3 Release (SI504)	6994
008	02/25/92	VMS 5.5 Release (SI505)	7052

TABLE OF CONTENTS

Section **Page**

Section **Page**

LIST OF ILLUSTRATIONS

Figure **Page**

LIST OF TABLES

Table **Page**

1
INTRODUCTION

The *Software Modification User Guide for VAX/VMS* explains how to configure System Industries devices with VAX processors running the VMS operating system. The manual consists of seven sections listed in Table 1-1.

Table 1-1. VAX Software Modification User Guide

SECTION	DESCRIPTION
Section 1 Introduction	Contains the manual preface, audience, related publications, conventions, terms list, and materials needed.
Section 2 Software Modifications	Provides an overview and discusses the configurations and restrictions of the VAX/VMS software modifications.
Section 3 Software Components	Provides a physical and functional description of the VAX/VMS software modification components.
Section 4 Installation	Describes automatic and manual installation of the VAX/VMS software modifications.
Section 5 Operation	Describes operation of the VAX/VMS software modifications.
Section 6 Cluster Configuration	Provides cluster installation and operation information.
Section 7 Maintenance and Troubleshooting	Contains system verification, maintenance, and troubleshooting information.

1.1 Manual Audience

The *Software Modification User Guide for VAX/VMS* is designed for customers familiar with Digital Equipment Corporation's VAX processor and peripherals, the VAX/VMS operating system, and SYSGEN procedures.

1.2 Related Publications

Additional information on related subjects is provided in the System Industries host computer and operating system manuals listed in Table 1-2.

Table 1-2. Related Publications

PUBLICATION NUMBER	TITLE
PB9900-9056	EXOR VAX Disk Exerciser User Guide
PB9904-9014	SIDOS User's Guide
AA-LA27B-TE	VMS VAXcluster Manual
AA-LA17B-TE	VMS System Messages and Recovery Procedures Reference Manual: Part 1
AA-LA18B-TE	VMS System Messages and Recovery Procedures Reference Manual: Part 2

1.3 Manual Conventions

Version numbers use the following symbols in naming the SI VAX/VMS software modification kit SIxyz:

- x = VMS major version number supported by the kit.
- y = Lowest minor version number supported by the kit.
- z = Highest minor version number supported by the kit.

Refer to the following documentation conventions as a guide to using this manual.

- Typed computer entry is shown in **boldface.** Type all boldface characters exactly as they appear. For example:

  Type: **f <RETURN>**

- Screen messages are displayed in a different typestyle, as follows:

  Printer attached to terminal? (Y/N) [N]:

- Key names are in **boldface** and shown in angle brackets. For example:

  <RETURN>
  <CONTROL> c

- Interactive sequences that include computer input and output are shown as follows:

  Printer attached to terminal? (Y/N) [N]: **y <RETURN>**
  Initializing...

- Variable typed entries, or text you must replace, are shown in *italics*. In the following example:

  Testing Cylinder *XXX* of *XXX* cylinders: *XXX* defects found

  XX and *XXXX* are italicized and replaced with the actual cylinder number tested.

- Three types of notes are used in this manual: a standard NOTE, a CAUTION note, and a WARNING note.

NOTE

The standard NOTE highlights important or additional information.

CAUTION

The CAUTION note is for situations that are potentially dangerous or destructive to data.

WARNING

A WARNING note is required if system failure or bodily injury is possible.

1.4 Terms List

Following are System Industries and industry-standard terms and acronyms used in this manual.

BBF	Bad Block Forwarding; a System Industries disk sector formatting process that provides a complete set of logical sectors arranged in order on the drive.
CI	Computer Interconnect, a DEC subsystem I/O interface.
CLUSTER	Configuration of VAX processors.
CLUSTOR	A System Industries mass storage subsystem.
CPU	Central Processing Unit; a computer.
DEC	Digital Equipment Corporation.
DCL	DEC Command Language.
ECC	Error Correcting Code; means for repairing data fields.
ETHERNET	DEC's loosely-coupled CPU Network.

EXOR	System Industries disk utility.
I/O	Input/Output; a data path.
MSCP	Mass Storage Control Protocol.
SI	System Industries.
SIDOS	System Industries Diagnostic Operating System; Diagnostic program developed by System Industries for disk and tape subsystems.
SYSGEN	System Generation.
TR	Transfer Request.
VAX	Virtual Address eXtension; a DEC computer.
VMS	Virtual Memory System; an operating system.

1.5 Materials Needed

The installation procedures described in this manual require a VMS operating system and a console terminal LA36 or any terminal that emulates the DEC VT series terminals. The software modifications are distributed on:

- 1600 BPI magnetic tape
- RX01/RX50 floppy diskette
- TK50/TU58 cartridge

2
SOFTWARE MODIFICATIONS

This section discusses the configurations and features of SI software modifications to the VAX/VMS operating system. The modifications are for use on DEC VAX computers. The software runs under the VAX/VMS operating system version specified in the title.

The software modifications provide operating system level connectivity to extended size disk and tape subsystems on VAX/VMS computers. This allows maximum use of the extended functionality of these peripherals. The software modifications support the connection of DEC QBUS, UNIBUS, and MASSBUS interfaces to storage subsystems.

The software modifications are distributed as a single save set, a file created by the DEC backup utility. The software modifications consist of:

- disk and tape I/O drivers
- patches to the DEC VMB boot routine
- a disk utility
- patches to the DEC stand-alone SYSGEN utility
- a stand-alone backup kit creation utility
- an autoconfiguration utility
- an installation procedure for the software

The software modifications are installed using the standard DEC VMSINSTAL command procedure. When VMSINSTAL is used, it automatically installs the software modifications into several system disk directories.

2.1 Configurations

The software modifications provide connection to devices over DEC interfaces. Hardware and software drivers are supported on VAX bus configurations. However, support is not provided on the MicroVAX I; an attempt to install SI software on a MicroVAX I results in an error message.

NOTE

All released drives are supported by the geometry calculation capabilities of the autosizing function.

The parameters that establish the capacity and format of a disk drive include:

- logical device units per drive
- cylinders per device unit
- tracks per cylinder
- sectors per track

The following terms are used to describe supported disk drives.

Mapped Drives

Mapped disk drives are drives whose logical properties recognized by the operating system are different from the actual physical properties. For example, two logical drives can be mapped to one physical drive. Using this, two logical DEC drives can be mapped to one extended size drive.

Mapping also provides the capability for a physical drive to emulate one or more different drive types. Disk drive mapping is performed in controller firmware and does not require operating system modifications. Mapped drives do not use their full physical storage capacity.

NOTE

When more than one logical unit is mapped to a physical drive, the logical unit numbers occupy consecutive unit numbers starting with that of the physical drive.

Direct Disk Drives

Direct disk drives are drives whose logical properties recognized by the operating system are the same as the actual physical properties. Therefore, there is one logical drive per physical drive. Direct drives may require operating system modifications. Direct drives use their full physical storage capacity.

Bad Block Forwarding Devices

Bad block forwarding (BBF) is a method of marking bad blocks on the disk pack. Disk drives use BBF instead of DEC's bad block replacement algorithm. A BBF drive reserves spare sectors at the end of each cylinder. When a bad block is found by the drive, it is marked as a skip sector. The sector number of every sector past the skip sector up to the end of the cylinder is incremented, then the first spare sector is used.

This process continues until all bad blocks are found and marked. BBF processing is performed at the system level and therefore is transparent to the user. Because spare sectors are required at the end of each cylinder, a BBF disk pack has one more physical sector per track than is logically available to the user.

Dual Disk Drives

Dual disk drives are two identical physical drives recognized by the operating system as one logical drive. This provides twice the amount of storage as one drive. Since the maximum number of logical storage nodes connected to a controller is eight, a dual drive configuration allows up to sixteen drives to be connected to one controller.

2.2 Restrictions

VMS system builds and updates can not be applied directly to an extended size disk drive. System builds and updates must be applied to a DEC disk drive or a disk drive that is mapped to emulate a DEC drive. However, an extended size disk drive is used as the system disk by copying the system from the standard size disk to the extended size disk after the system build or update is completed. The standard size disk must be maintained to facilitate system builds and updates.

Serial number conflict can cause a DEC drive to be recognized as an SI device. If this occurs, the DEC drive unit number should be changed. If this does not resolve the conflict, call the local SI field office.

3
COMPONENTS

This section describes the files in the primary distribution saveset and the files created on the target system during software modification. The MicroVAX boot-function files and saveset are described in Appendix A. VMSINSTAL is used to install the software modifications into various system directories, as shown in Figure 3-1.

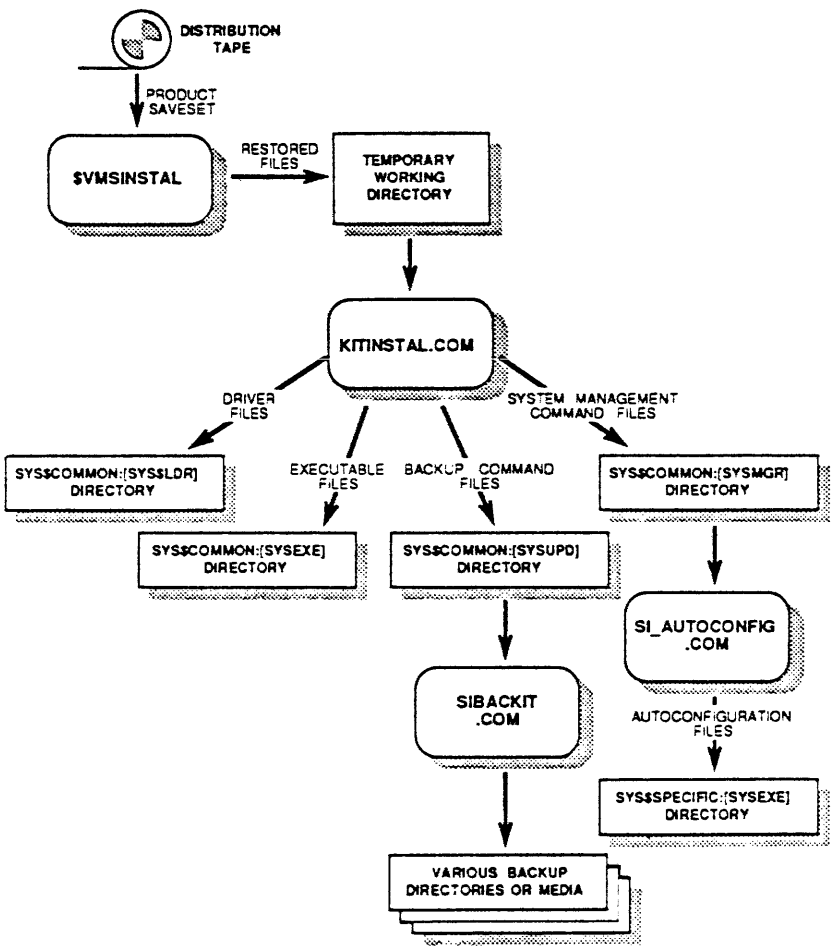

Figure 3-1. Installation of the Software Modifications

VMSINSTAL moves the saveset files to a temporary directory on the system disk. File KITINSTAL.COM then automatically completes installation. KITINSTAL.COM patches system images, links drivers, and moves files to the target directories. After KITINSTAL.COM terminates, VMSINSTAL deletes the temporary working directory and returns to the system level prompt. Next, the user executes SI_AUTOCONFIG.COM to create the patches necessary to autoconfigure SI devices. Then the user executes SIBACKIT.COM to make a stand-alone backup kit.

3.1 Files

The distribution saveset contains the files listed in Table 3-1. After installation, the files in Table 3-2 are on the system disk. To configure a disk as a system device, create the files in Table 3-3 and place them onto the console media to allow device booting. The files in Tables 3-4 are created when the user executes SI_AUTOCONFIG.COM.

Table 3-1. Distribution Saveset Files

DISTRIBUTION MEDIA SAVESET	FILE	DESCRIPTION
SIxxxxxx.A	KITINSTAL.COM	An interactive command file used to install all SI code.
	DIDRIVER.OBJ	Q/UNIBUS I/O driver for SI Disk Controller and attached drives.
	SI_AUTOCONFIG.COM	Autoconfiguration command procedure.
	MMDRIVER.OBJ	UNIBUS I/O driver for the SI Tape Controller and attached drives.
	EXOR.OBJ	Disk exercising diagnostic program.
	STAEXOR.OBJ	Disk exercising diagnostic program for stand-alone mode.
	SIBACKIT.COM	.COM file for building stand-alone BACKUP and EXOR.
	DRDRIVER.OBJ	MASSBUS I/O driver for the SI Disk Controller and attached drives.

Table 3-2. Created Files After Installation

DIRECTORY	FILE	DESCRIPTION
SYS$COMMON: [SYS$LDR]	DIDRIVER.EXE	SI QBUS/UNIBUS disk driver.
	DRDRIVER.EXE	SI MASSBUS disk driver.
	MMDRIVER.EXE *	SI UNIBUS tape driver.
	DRDRIVER.OLD	Original DEC MASSBUS disk driver.
	TMDRIVER.EXE	Patched version of DEC MASSBUS tape driver.
	TMDRIVER.OLD	Original DEC MASSBUS tape driver.
SYS$COMMON: [SYSEXE]	SIVMB.EXE	SI version of VMB.EXE primary boot file.
	EXOR.EXE	Disk exerciser for SI disks.
	STAEXOR.EXE	Stand-alone version of EXOR.
	SISYSGEN.EXE	SI version of STASYSGEN for stand-alone kits.
SYS$COMMON: [SYSUPD]	SIBACKIT.COM	.COM file for building stand-alone EXOR and BACKUP.
	SIPATCH.JNL	Patch journal file.
	SIBACKIT_TABLE.DAT	Data table used by SIBACKIT.COM.
SYS$COMMON: [SYSMGR]	SI_AUTOCONFIG.COM	.COM file for CLUSTOR autoconfiguration.
	MMDRIVER.COM *	.COM file for loading the MMDRIVER.

* Installed on non-BI machines only.

Table 3-3. Device Booting Files

FILE	DESCRIPTION
SInBOO.CMD	Boot file for A controller, SIn drive (one for each drive).
SIxBOO.CMD	Boot file for x controller (one for each controller).
SIxGEN	Conversational boot file (one for each controller).
SIxXDT	XDelta boot file (one for each controller).

Table 3-4. Autoconfiguration Files

DIRECTORY	FILE	DESCRIPTION
SYS$SPECIFIC: [SYSEXE]	STACONFIG.EXE SYSGEN.EXE	Patched version of STACONFIG Patched version of SYSGEN
SYS$COMMON: [SYSEXE]	STACONFIG.OLD SYSGEN.OLD	Original DEC version of STACONFIG Original DEC version of SYSGEN

NOTE

The .EXE versions of the above images also remain unpatched in the SYS$COMMON directory.

4
INSTALLATION

This section lists the steps required to install the software modifications. It includes commands to be entered into the system and the expected system responses. The MicroVAX boot function is described in Appendix A.

VMSINSTAL is a DEC tool provided to install software products. This interactive command procedure configures the system for SI disk and tape devices. Two modes are used for installing SI software products: automatic and manual.

For initial device installation, the automatic procedure is recommended. Use manual installation only when executing part of the installation procedure to correct any errors encountered in automatic mode.

4.1 VMS Version Numbers

If a DEC update has just been installed and the machine has not been rebooted, this installation procedure sees the old system version number. The default version number must be changed to the new DEC update version when prompted.

If attempting to patch a version of VMS that is above the supported level of this command procedure, carefully watch for patch verify errors indicating the file has changed with this new version. If errors are seen, save the printout of the procedure and contact the SI field office.

4.2 VAXCluster Installation

The software modifications are installed one time only for each system disk in a VAXCluster. The modifications are installed on any node using the target system disk. For example; in a VAXCluster with a single system disk for all nodes, the software modifications are installed once on any one of the nodes. The files that are created during installation (Table 3-2) are placed into SYS$COMMON and can therefore be accessed by all nodes using the system disk.

After the software modifications are installed, the SI_AUTOCONFIG utility (Section 5) is run on **each node** in the VAXCluster. Due to the hardware differences between the nodes and the user input supplied to SI_AUTOCONFIG, the files that are created by this utility (Table 3-4) contain node specific information. Since they are node specific, the files are placed into SYS$SPECIFIC for access by the local node only.

4.3 Console Media

VMSINSTAL will not directly change the console media. However, when following the installation procedures, manual changes to the media may be desired. SI recommends that the changes be made to a copy of the media. Copies of the media are made by the VMS utility, CONSCOPY.COM.

4.4 Prepare for Installation

Before installing SI software modifications into the system, make sure the following steps are completed:

1. Verify that the system contains only the DEC version of any images to be patched by the SI software modifications. If not, restore the DEC version of the images.
2. Log in to the system account on the operator's console, OPA0.

4.5 Automatic Installation

For automatic installation of SI software modifications, follow these steps:

1. Invoke VMSINSTAL by typing:

   $ @SYS$UPDATE:VMSINSTAL SI5xx MTA0 <RETURN>

   MTA0 is the mnemonic for the device being used. The actual device mnemonic can be different. If this node does not have a device compatible with the distribution media, refer to "Operation". SI5xx is the same number version as the title of this document.

   NOTE

   Always enter product name (SIxxx) explicitly. Do not use wildcard (*) as product name.

   The following message is displayed:

   VAX/VMS Software Product Installation Procedure V5.x

   It is 27-SEP-1988 at 18:29.
   Enter a question mark (?) at any time for help.

2. A series of questions follow that must be answered either YES, NO, or ? for help:

   * Are you satisfied with the backup of your system disk [YES]?

Backing up the system disk before performing any installation with VMSINSTAL is recommended. If **YES** or **RETURN** is entered at this prompt, the installation proceeds. Entering **NO** and **RETURN** exits the procedure.

3. Load the distribution media into the appropriate disk or tape drive. Type **YES** **<RETURN>** at the following prompt:

```
Please mount the first volume of the set on _MTA0:
* Are you ready?
```

The installation begins, displaying the following:

```
The following products will be processed;

SI5xx V5.x

    Beginning installation of SI5xx V5.x at 18:30

%VMSINSTAL-I-RESTORE, Restoring product saveset A ...
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                            SYSTEM INDUSTRIES                                      !
!                                                                                   !
!                                  SI5xx                        OCT. 1988           !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

    SYSTEM INDUSTRIES DISK AND TAPE INSTALLATION PROCEDURE

    THIS INSTALLATION KIT IS FOR VAX/VMS V5.x through V5.x

This is a VAX 8SSS Type CPU   SYSTEM ID 86518548   RUNNING VMS V5.x

The current time is: 27-SEP-1989   15:30:49.93

                              *** NOTE ***

    Check the VMS Version above. If you have just installed a DEC Update and have not rebooted,
    this installation procedure will see the old system version number therefore, you must manually
    change the default version number to that of the new DEC Update Version in the following inquiry;
    Type "RETURN" otherwise.

    * * Identify Version - (default V5.x    ):
```

4. The following message is displayed:

```
* Do you want automatic installation [YES}?
```

A response of **YES** is recommended for most installations. To alternately invoke manual installation, enter **NO** and see *"Manual Installation"* (section 4.7).

5. Check the VMS version. Enter the correct version number.

NOTE

Enter only the major and minor version number, for example, V5.1.

6. The following message is displayed:

The following default device configuration information will be used by this installation procedure:

If you're doing a UNIBUS tape installation:

# of UNIBUS TAPE CONTROLLERS	1
UNIBUS adaptor for the controller	UBA
Number of drives on this controller	2
Controller CSR	772440
Controller Vector	224

If you're doing a UNIBUS disk installation:

	Primary	Secondary
UNIBUS disk CSR	176700	176300
UNIBUS disk Vector	254	150

If you wish to use alternate device configuration information, you will be prompted for the new numbers in the appropriate section below.

Do you wish to use alternate device information? (Y/[N]): :

Enter **Y** or **N** and **RETURN**.

To change the default device configuration information for UNIBUS tape and disk devices, enter YES. For non-UNIBUS installations, enter NO at the prompt, as the changes do not affect the system.

If the answer is YES, VMSINSTAL asks for new configuration information. If the answer is NO, VMSINSTAL runs to completion with no user interaction.

7. A series of messages follows, indicating the automatic processing being performed by VMSINSTAL. These messages are self-explanatory. These are the processing steps being performed:

Link the SI Q/UNIBUS disk driver (DIDRIVER)
Link the SI MASSBUS disk driver (DRDRIVER)
Create the SIVMB primary boot file boot file (SIVMB
Link the SI disk exerciser (EXOR)
Link the SI UNIBUS tape driver (MMDRIVER)
Create the MMDRIVER SYSGEN connect file
Patch the DEC TMDRIVER
Patch STASYSGEN
Create the SI utility, SIBACKIT.COM
Create the SI utility, SI_AUTOCONFIG.COM

8. On completion of VMSINSTAL, the following message is displayed:

```
**************************************************************
*                                                            *
*     >>>  SYSTEM INDUSTRIES Disk and Tape Configuration is complete  <<<     *
*                                                            *
**************************************************************
```

VMSINSTAL procedure done at 18:34

4.6 Manual Installation

This manual software installation mode should only be used when executing part of the installation procedure in order to correct errors encountered in automatic node.

Menu items are arranged in the order they are performed. Execute menu selections from the last step where the error in automatic mode occurred and continue through to the end of the list. Each time a component is installed correctly, the menu is displayed again.

Repeat steps 1-3 as listed in *"Automatic Installation"* (section 4.6). At step 4, enter "NO." The following menu displays:

COMPONENT SELECTION MENU

0 EXIT

1 Link the SI Q/UNIBUS disk driver (DIDRIVER)
2 Link the SI MASSBUS disk driver (DRDRIVER)
3 Create the SIVMB primary boot file boot file (SIVMB)
4 Link the SI disk exerciser (EXOR)
5 Link the SI UNIBUS tape driver (MMDRIVER)
6 Create the MMDRIVER SYSGEN connect file
7 Patch the DEC TMDRIVER
8 Patch STASYSGEN
9 Create the SI utility, SIBACKIT
10 Create the SI utility, SI_AUTOCONFIG.COM

Please make a selection from the menu:

4.7 Rebooting the System

When installing SI software modifications, reboot after completion of the procedure. This ensures the patches and new executables are loaded.

5
OPERATION

This section explains the components of the installation and the operation of the software modifications.

5.1 Installation Components

The following paragraphs discuss the operation of the installation steps. The software modifications are installed in the following steps.

1. Link the SI Q/UNIBUS disk driver (DIDRIVER)
2. Link the SI MASSBUS disk driver (DRDRIVER)
3. Create the SIVMB primary boot file boot file (SIVMB)
4. Link the SI disk exerciser (EXOR)
5. Link the SI UNIBUS tape driver (MMDRIVER)
6. Create the MMDRIVER SYSGEN connect file
7. Patch the DEC TMDRIVER
8. Patch STASYSGEN
9. Create the SI utility, SIBACKIT
10. Create the SI utility, SI_AUTOCONFIG.COM

DIDRIVER Q/UNIBUS Disk Driver

SI Q/UNIBUS disk driver, DIDRIVER, is linked to the operating system software. DIDRIVER.EXE is created in SYS$COMMON:[SYS$LDR]. The driver communicates with SI RP0X or RM0X type drives attached by means of the Q/UNIBUS interface.

The patches to SYSGEN made by SI_AUTOCONFIG.COM cause the primary and secondary SI controllers to be autoconfigured at boot time. Therefore, no SYSGEN connect statements are needed for the DIDRIVER.

Alternately, for manual connection or for more than two controllers, use the following SYSGEN connect commands. These manual connect statements for the DIDRIVER are included in the start-up procedure SYS$MANAGER:SYSCONFIG.COM.

```
$ RUN SYS$SYSTEM:SYSGEN <RETURN>
$ LOAD DIDRIVER <RETURN>
$ CONNECT DRC0/ADAPTER=UB0/VECTOR=%O250/CSR=%O777000/DRIVER=DIDRIVER
  <RETURN>
```

```
$ CONNECT DRC1/ADAPTER=UB0/VECTOR=%O250/CSR=%O777000/DRIVER=DIDRIVER
  <RETURN>
$ EXIT <RETURN>
```

DRDRIVER MASSBUS Disk Driver

SI MASSBUS disk driver, DRDRIVER, is linked with the operating system software. DRDRIVER.EXE is created in SYS$COMMON:[SYS$LDR]. The driver communicates with SI RM type drives attached by means of the MASSBUS interface. It is also compatible to DEC's RM type drives and becomes a replacement to DEC's DRDRIVER.

When using a standard size SI drive, SI's DRDRIVER is not needed. However, to use the SI disk exerciser, EXOR, the extended functionality of SI's DRDRIVER is needed.

DEC does error correction code (ECC) correction in its version of the DRDRIVER. SI devices do ECC correction in hardware. Part of the functionality of SI's DRDRIVER is to set a bit disabling hardware error correction for the I/O function codes Format and Datacheck. Since EXOR and stand-alone EXOR use Format and Datacheck in the process of badding drives, the software changes impact effectiveness in locating all errors. For this reason SI recommends that SI's DRDRIVER always be installed for SI drives.

SIVMB Primary Boot File

SIVMB.EXE is created in SYS$COMMON:[SYSEXE]. This allows SI extended size disks or RM disks to boot off on the UNIBUS.

After the completion of the installation procedure, SIVMB.EXE is copied from SYS$COMMON:[SYSEXE] to the console device. To do this, use the VMS EXCHANGE utility. After SIVMB.EXE is copied to the console media, this primary boot file is used for booting all SI extended capacity disks. SIVMB is capable of booting all SI disks (UNIBUS and/or SBI/CMI) and is still compatible with all DEC disks except the RK07.

The boot command file used to boot any SI disk must therefore load SIVMB and not VMB (this load is done in the next to last line of the boot command file). Any existing DEC MASSBUS boot command files (for example, CSA1:DBBBOO.CMD if the SI controller is on the second adapter) can be copied to the system, renamed, edited, and then copied back to the console media with the following commands:

```
$ EXCHANGE COPY CSA1:DBBBOO.CMD DS0BOO.CMD <RETURN>
$ EDIT/EDT DS0BOO.CMD <RETURN>
```

edits the file
adds the deposit of 0 into R3 - to indicate unit number zero
changes VMB.EXE to SIVMB.EXE

```
$ EXCHANGE COPY DS0BOO.CMD CSA1: <RETURN>
>>>B DS0 <RETURN>
```

used boot time on a VAX-780.

NOTE

DEFBOO.CMD can also be modified to boot the SI device as the default system device.

To boot the disk from a PROM (VAX 11/750 only), use the DEC WRITEBOOT utility to make the boot block point to the SIVMB.EXE file. A contiguous copy of SIVMB.EXE must continue to reside on the system disk in the SYS$COMMON:[SYSEXE] directory.

To reinstall a DEC disk as a system disk, use the proper boot file to boot that disk. The system disk can alternate with the appropriate boot file. The installation kit does not inhibit the standard operation of VMS.

If the default boot file was not changed to boot the SI disks, the boot console device will have new files, but all existing files remain unchanged. If the default boot file has been altered, changes it back to the DEC boot file to reinstall the DEC disk as a system disk. This is very important in systems with automatic rebooting enabled.

EXOR

EXOR and STAEXOR are linked and placed into SYS$COMMON:[SYSEXE]. EXOR is the on-line utility for the SI controller and disks. After installation is complete, EXOR is used for performing diagnostics, formatting, and badding SI disks. STAEXOR is the same as EXOR except that it is a stand-alone image used in stand-alone mode.

One EXOR test runs patterns on SI disks and maintains the bad block file. EXOR is more thorough than the DEC command language (DCL) ANAL/MEDIA and is recommended for all SI disks.

After installation is complete, EXOR is run to verify proper installation, formatting, and badding of the disks. EXOR is also used to check the factory-built bad block file.

EXOR tests RP04, RP05, RP06, RM03, and RM05 type drives. Running EXOR requires DIAGNOSE and PHY_IO process privileges. EXOR prompts for the drive to be tested and for the tests to be run. It runs the tests given, in the order given, and then returns to ask for more tests.

Respond to EXOR questions as follows:

```
Drive: (enter logical device name such as DRA0:)
```

Check to see if the drive is correctly identified and verify the disk geometry printout. If the geometry is not correct, check the hardware, driver patches, and hardware configuration. Do not continue until the problem is resolved.

Tests: (enter the test letter to run)

When the disk is formatted, use the tests below:

B: List the bad block file
H: List all possible tests
L: Initialize the bad block file and assign serial number
P: Run pattens on the disk, build bad block file
X: Exit EXOR

If the disk is fixed media type, use the EXOR B test and list the bad block file. In the B test the bad blocks are automatically listed, then add any new blocks. If there are no bad block files initialize the bad block file with test L, then select P test to run patterns and bad the disk.

It is recommended that the first two patterns of the P test be run on the new SI drive to verify that no new bad spots have occurred during shipping. If the test finds any, they are reported as new bad blocks in the listing. This test also verifies the existence of a good factory bad block file.

If the disk is unformatted, use the tests below:

F: Format the disk (type "Y" to format last track)
L: Initialize the bad block file and assign serial no.
P: Run pattens on the disk, build bb file
X: Exit EXOR

NOTE

If the disk is reformatted, run as many patterns as time allows before using the new disk.

For a detailed explanation of EXOR, see the *EXOR VAX Disk Exerciser User Guide.*

MMDRIVER UNIBUS Tape Driver

SI UNIBUS tape driver, MMDRIVER, is linked to operating system software. MMDRIVER.EXE is created in SYS$COMMON:[SYS$LDR]. The driver communicates with SI type drives attached by means of the UNIBUS interface.

The driver loading procedure, MMDRIVER.COM is created in SYS$COMMON:[SYSMGR]. The start up is modified to use this procedure. This section is bypassed on QBUS systems.

TMDRIVER Tape Driver

The DEC standard tape driver, TMDRIVER.EXE is patched in order to make use of the 6250 BPI capability of the SI SBI/CMI tape drives. The original DEC version of TMDRIVER is saved as TMDRIVER.OLD. Keep TMDRIVER.OLD for future DEC updates that must patch the original driver. This section is bypassed on QBUS systems.

Stand-Alone Utilities

The SI utility SIBACKIT.COM is copied to SYS$COMMON:[SYSUPD]. It makes stand-alone BACKUP and EXOR kits for use with SI devices. After completion of VMSINSTAL, invoke the command procedure with the following command:

```
$ @SYS$COMMON:[SYSUPD]SIBACKIT <RETURN>
```

The dialogue that follows is identical to DEC's STABACKIT utility.

The DEC utility STASYSGEN.EXE is patched to create SISYSGEN.EXE. This autoconfigures all SI devices on the Q/UNIBUS while the BACKUP utility boots up. When using the stand-alone build procedure SIBACKIT.COM, this file is copied to the BACKUP kit as SYSINIT.EXE. SIBACKIT.COM should is also used for SI MASSBUS devices.

A stand-alone BACKUP or EXOR kit is made on most any device on the system by using SIBACKIT.COM. The most common device used is the console (CSA1:).

A disk or system disk is also made to boot stand-alone BACKUP or EXOR kits. If a system disk containing BACKUP or EXOR kits is selected, the disk is not reinitalized and the files are forced to directories on the system disk. The BACKUP kit is placed in the SYSE.SYSEXE directory and the EXOR kit is placed in the SYSD.SYSEXE directory. Both of these directories on the system disk have a SYS$COMMON root which points to SYS$TOPSYS for normal systems and to SYS$COMMON for shared system disks.

The stand-alone BACKUP or EXOR kit configures SI Q/UNIBUS disks as DIxx, regardless of device names specified in the SI_AUTOCONFIG utility. It is necessary, therefore, to use a name in the form DIxx in the BACKUP or EXOR commands.

SI_AUTOCONFIG Utility

The SI utility SI_AUTOCONFIG.COM is copied to SYS$COMMON:[SYSMGR]. After completion of VMSINSTAL, invoke the command procedure SI_AUTOCONFIG.COM with the following command:

```
$ @SYS$COMMON: [SYSMGR]SI_AUTOCONFIG.COM <RETURN>
```

The purpose of this command procedure is to apply patches to DEC's STACONFIG.EXE and SYSGEN.EXE images, allowing SI Q/UNIBUS disks to be autoconfigured at boot time. In addition, this command procedure aids in setting the special SYSGEN parameter USER3 for cluster configurations.

The SI Q/UNIBUS disks are autoconfigured with any device name and controller designator desired. Care must be taken in entering device names in response to the prompt in the SI_AUTOCONFIG procedure.

CAUTION

If the VMS autoconfigure routine called by DEC's STACONFIG or SYSGEN configures a new device with a name that already exists, system crashes may result with the BUGCHECK message of inconsistent data structures.

The patched STACONFIG and SYSGEN utilities handle two Q/UNIBUS disk controller addresses. The default disk controller CSR and vector values are listed in Table 5-1.

Table 5-1. Default Disk Controller CSR and Vector Values (octal)

PRIMARY	SECONDARY
CSR - 176700	CSR - 176300
VEC - 254	VEC - 150

The CSR and vector values of the primary and secondary controllers are changed by means of this utility, but using the default CSR and vector is recommended.

If the device is dual ported in a VAXcluster with the other ports being MASSBUS connections, the appropriate device name is DR. The controller designator is also set to match the MASSBUS device name by means of this utility.

TR Levels

TR level of the SI controller is determined by using the SYSGEN SHOW/CONFIGURE command as follows.

```
$ RUN SYS$SYSTEM:SYSGEN <RETURN>
SYSGEN> SHOW /CONFIGURATION <RETURN>

System CSR and Vectors on 15-JUL-1988 10:13:18.49

Name: DRA  Units: 1  Nexus:8  (MBA)
Name: MTA  Units: 1  Nexus:9  (MBA)
Name: DRC  Units: 4  Nexus:10 (MBA)
```

```
Name: DRD  Units: 4  Nexus:11 (MBA)
Name: DRE  Units: 8  Nexus:12 (MBA)
Name: XEA  Units: 3  Nexus:3  (UBA) CSR: 774510 Vector1...
Name: LPA  Units: 1  Nexus:3  (UBA) CSR: 777514 Vector1...
Name: TTA  Units: 8  Nexus:3  (UBA) CSR: 760100 Vector1...
Name: SIA  Units: 8  Nexus:3  (UBA) CSR: 776700 Vector1...
Name: XEA  Units: 3  Nexus:3  (UBA) CSR: 774510 Vector1...
```

In this example, notice that the DRC devices attached to a SI controller on this CPU are installed at a TR level (or Nexus) of 10.

The example shown is from a VAX 780. In a VAX 750, the logically equivalent TR level is 4 less than the 780's. A summary of several CPU TR levels is shown in Table 5-2.

Table 5-2. Summary of TR Levels

ADAPTERS	730 TR#	750 TR#	780 TR#	86xx TR#
UBA 0	3	8	3	3
UBA 1	-	9	4	4
UBA 2	-	-	5	5
UBA 3	-	-	6	6
MBA 0	-	4	8	8
MBA 1	-	5	9	9
MBA 2	-	6	10	10
MBA 3	-	-	11	11

SYSGEN USER3 Parameter for Shared Devices

A SI disk controller can be connected to a maximum of eight nodes. When multiple nodes are connected to the SI controller, the single value assigned with the SYSGEN parameter ALLOCLASS may be inadequate, as this may leave several local disks with the same name on several nodes leading to disk corruption.

To accommodate this situation, SI uses an alternate method of determining an allocation class for shared devices. The user defined SYSGEN parameter USER3 determines the allocation class for both MASSBUS and UNIBUS devices. The lower word of USER3 corresponds to the TR number of the MASSBUS devices for which allocation class is to be set. The upper two bytes of USER3 sets the allocation class for MASSBUS and UNIBUS devices respectively.

NOTE

This feature is reserved for SI attached devices only and will not affect the standard naming convention for DEC RM devices.

If the user plans to serve the SI MASSBUS or Q/UNIBUS devices, the SYSGEN parameter ALLOCLASS must match the allocation class set in the upper 2 bytes of the SYSGEN parameter USER3.

For Massbus connections, DRDRIVER uses the lower word of USER3 as a mask of the device TR levels for which allocation class is set. TR levels determine the controller designator. Each bit corresponds to a TR level, with bit0 corresponding to TR0, and bit15 corresponding to TR15. The upper word contains the allocation class value for the MASSBUS and UNIBUS devices. For example, allocation class is set to 255 for the MASSBUS devices at TR8 and TR10 on controllers DRA and DRC, the USER3 long word is shown in Figure 5-1.

NOTE

As for all nondynamic SYSGEN parameters, the new value for USER3 is only valid after the system is rebooted.

If the user plans to set allocation class for SI UNIBUS devices, the upper byte of USER3 is set. The MASSBUS allocation class and TR's are ignored by the UNIBUS driver. This allows the user to set USER3 to contain allocation class for MASSBUS and UNIBUS devices.

Using this as an example, shared disks on controllers at TR8 and TR10 will have the device names 255DRA0 through 255DRA7 and 255DRC0 through 255DRC7 respectively. The SI_AUTOCONFIG routine sets this value based on configuration questions; understanding the bit mask is not necessary.

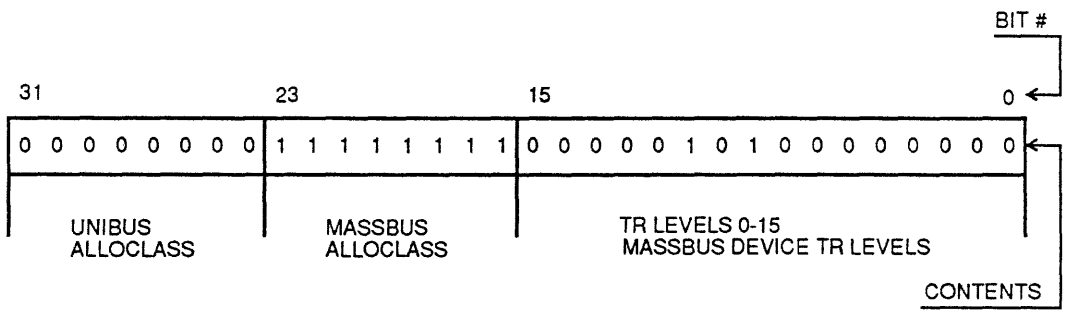

Figure 5-1. USER3 Long Word

SYSGEN USER4 Parameter for 86xx Users

For VAX 86xx systems only with an RH780 attached to a second SBI bus, support is extended for the setting of SI allocation class. The SYSGEN parameter USER4 is used to set TR levels in the same manner as USER3. USER4 parameter must be set manually.

The low 16-bits of the SYSGEN parameter USER4 is a bit mask describing the TR levels on which the allocation class is set. Each bit corresponds to a TR level, with bit0 corresponding with TR16, and bit15 corresponding to TR31. TR16 through TR23 are reserved for processor slots. For example, the allocation class is set to the value in the upper word of USER3 for MASSBUS devices at TR17 and TR19 on second SBI. The USER4 long word is shown in Figure 5-2. Bit0 of USER3 must be set manually before USER4 is used.

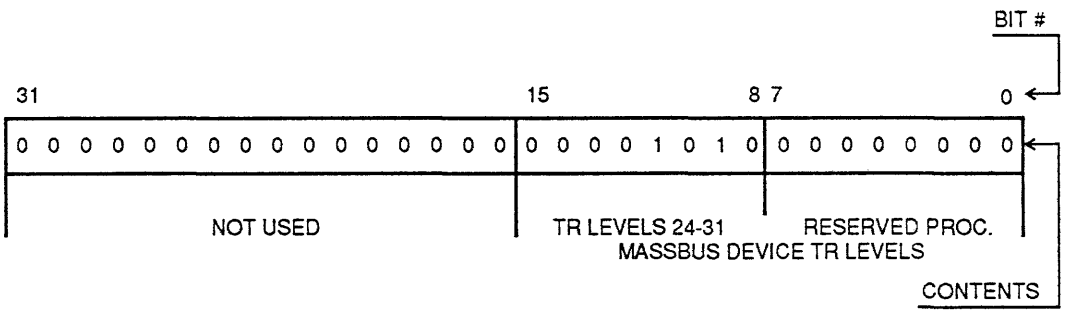

Figure 5-2. USER4 Long Word

5.2 UNIBUS Boot Devices

The SIVMB patches replaced the RK07 boot code with SI device boot code. This means that the user can not use SIVMB to boot off the RK07. To boot off that device, the boot command file must use the original DMxBOO to invoke DEC's VMB instead of SIVMB, or enter the correct parameters in response to the console bootstrap prompt.

UNIBUS Boot Files

The following paragraphs describe UNIBUS boot files for SI disks.

The user can create files similar to the examples below and then use EXCHANGE to copy them to the console media. Any of the copied boot files may be changed to the default boot file by executing the standard VMS procedure:

```
$ @SYS$UPDATE:SETDEFBOO
```

Examples of UNIBUS boot files for VAX 11/750, VAX 11/780, and VAX 8600 are listed in the following paragraphs.

VAX 11/750 UNIBUS Boot File

This example is named SI0BOO.COM. It is the unit 0 boot file for the primary controller.

```
SI0BOO BOOT COMMAND FILE - SI0BOO.COM
```

```
D/G 0 0AC0001                ! VECTOR (primary 254 oct. = 0AC hex.)
D/G 1 FFE000                 ! BASE OF UNIBUS I/O PAGE
D/G 2 3FDC0                  ! CSR ADDRESS OFFSET (primary 776700 oct. = 3FDC0 hex)
D/G 3 0                      ! UNIT NUMBER (zero)
D/G 4 0                      ! BOOT BLOCK LBN (UNUSED)
D/G 5 0                      ! CONVERSATIONAL/DEBUG FLAG
D/G E 200                    ! ADDRESS OF WORKING MEMORY + 200
LOAD SIVMB.EXE/START:200     ! LOAD PRIMARY BOOTSTRAP
START 200                    ! AND START IT
```

The first and third lines of this sample COM file indicate booting off the primary controller whose vector is 254 octal and whose CSR is 776700 octal. In order to indicate booting off the secondary controller, change the first and third lines to:

```
D/G 0 0680001                ! VECTOR (primary 150 oct. = 068 hex.)
D/G 2 3FCC0                  ! CSR ADDRESS OFFSET (primary 776300 oct. = 3FCC0 hex)
```

The fifth line of this sample COM file indicates booting off unit zero of the selected controller. In order to indicate booting off another unit, change the fifth line to indicate that unit number. For example, to boot off unit 2, change the fifth line to:

```
D/G 3 2                      ! UNIT NUMBER (two)
```

The debug command file, SI0XDT.COM, is created by using a copy of the example, changing the sixth line to:

```
D/G 5 7                      ! CONVERSATIONAL/DEBUG FLAG
```

The conversational boot file, SI0GEN.COM, is created by using a copy of the example, changing the sixth line to:

```
D/G 5 1                      ! CONVERSATIONAL/DEBUG FLAG
```

VAX 11/780 UNIBUS Boot File

This example is named SI0BOO.COM. It is the unit 0 boot file for the primary controller on UBO.

```
SI0BOO  BOOT COMMAND FILE -- SI0BOO.COM

HALT                         ! HALT PROCESSOR
UNJAM                        ! UNJAM SBI
INIT                         ! INIT PROCESSOR
DEPOSIT/I 11 20003800        ! SET UP SCBB
DEPOSIT R0 0AC0001           ! DISK VECTOR (primary 254 oct. = 0AC hex. in high word)
DEPOSIT R1 3                 ! UBA ADAPTER (UB0 = 3)
DEPOSIT R2 3FDC0             ! CSR ADDRESS OFFSET (primary 776700 oct. = 3FDC0 hex)
```

```
DEPOSIT R3 0                    ! UNIT NUMBER (zero)
DEPOSIT R4 0                    ! BOOT BLOCK LBN (UNUSED)
DEPOSIT R5 4000                 ! CONVERSATIONAL/DEBUG FLAG
DEPOSIT FP 0                    ! SET NO MACHINE CHECK EXPECTED
START 20003000                  ! START ROM PROGRAM
WAIT DONE                       ! WAIT FOR COMPLETION
                                !
EXAMINE SP                      ! SHOW ADDRESS OF WORKING MEMORY+200
LOAD SIVMB.EXE/START:@          ! LOAD PRIMARY BOOTSTRAP
START @                         ! AND START IT
```

The fifth and seventh lines of this sample COM file indicate booting off the primary controller whose vector is 254 octal and whose CSR is 776700 octal. In order to indicate booting off the secondary controller, change the fifth and seventh lines to

```
DEPOSIT R0 0680001              ! DISK VECTOR (primary 150 oct. = 068 hex.)
DEPOSIT R2 3FCC0                ! CSR ADDRESS OFFSET (primary 776300 oct. = 3FCC0 hex)
```

The sixth line of this sample COM file indicates booting off Q/UNIBUS adapter 0 (UB0). In order to indicate booting off another Q/UNIBUS adapter use the values in Table 5-3.

Table 5-3. Q/UNIBUS Adapter Values

Q/UNIBUS ADAPTER	r1 CONTENTS
UB0	3
UB1	4
UB2	5
UB3	6

To boot off UB2, change the sixth line to:

```
DEPOSIT R1 5                    ! UBA ADAPTER (UB0 = 5)
```

The eighth line of this sample COM file indicates booting off unit zero of the selected controller. In order to indicate booting off another unit, change the eighth line to indicate that unit number. To boot off unit 2, change the eighth line to:

```
DEPOSIT R3 2                    ! UNIT NUMBER (two)
```

The debug command file, SI0XDT.COM, is created by using a copy of the example, changing the tenth line to:

```
DEPOSIT R5 4007                 ! CONVERSATIONAL/DEBUG FLAG
```

The conversational boot file, SI0GEN.COM, is created by using a copy of the example, changing the tenth line to:

```
DEPOSIT R5 4001                ! CONVERSATIONAL/DEBUG FLAG
```

VAX 8600 UNIBUS Boot File

This example is named SI0BOO.COM; the unit 0 boot file for the primary controller on UB0.

```
SI0BOO  BOOT COMMAND FILE – SI0BOO.COM

SET SNAP ON                    ! ENABLE ERROR_HALT SNAPSHOTS
SET FBOX OFF                   ! VMS WILL TURN ON FBOX
INIT                           ! INIT PROCESSOR
UNJAM                          ! UNJAM SBI
DEPOSIT CSWP 8                 ! TURN OFF THE CACHE (VMS TURNS THIS ON)
DEPOSIT R0 0AC0001             ! DISK VECTOR (primary 254 oct. = 0AC hex.)
DEPOSIT R1 3                   ! UBA ADAPTER (UB0 = 3)
DEPOSIT R2 3FDC0               ! CSR ADDRESS OFFSET (primary 776700 oct. = 3FDC0 hex)
DEPOSIT R3 0                   ! UNIT NUMBER (zero)
DEPOSIT R4 0                   ! BOOT BLOCK LBN (UNUSED)
DEPOSIT R5 0                   ! CONVERSATIONAL/DEBUG FLAG
FIND/MEMORY                    ! LOCATE 64KB OF GOOD MEMORY
EXAMINE SP                     ! SHOW ADDRESS OF WORKING MEMORY+200
LOAD/START:@ SIVMB             ! LOAD PRIMARY BOOTSTRAP
START @                        ! AND START IT
```

The sixth and eighth lines of this sample COM file indicate booting off the primary controller whose vector is 254 octal and whose CSR is 776700 octal. In order to indicate booting off the secondary controller, change the sixth and eighth lines to:

```
DEPOSIT R0 0680001             ! DISK VECTOR (primary 150 oct. = 068 hex.)
DEPOSIT R2 3FCC0               ! CSR ADDRESS OFFSET (primary 776300 oct. = 3FCC0 hex)
```

The seventh line of this sample COM file indicates booting off Q/UNIBUS adapter 0 (UB0). To indicate booting off another Q/UNIBUS adapter use the values in Table 5-3.

```
DEPOSIT R1 5                   ! UBA ADAPTER (UB0 = 5)
```

The ninth line of this sample COM file indicates booting off unit zero of the selected controller. In order to indicate booting off another unit, change the ninth line to indicate that unit number. To boot off unit 2, change the ninth line to:

```
DEPOSIT R3 2                   ! UNIT NUMBER (two)
```

The debug command file, SI0XDT.COM, is created by using a copy of the example, changing the eleventh line to:

6
CLUSTER CONFIGURATION

This section provides information on using SI devices in a VAXcluster environment. Refer to the *VMS VAXcluster Manual* for more information.

6.1 Mounting Shared Data Disks

To give all nodes in a cluster synchronized access to a shared SI disk, set the disk to dual ported and mount it with the /CLUSTER qualifier. Do this on each node directly connected to the SI controller accessing the shared device.
SYS$MANAGER:SYSTARTUP_V5.COM, the site specific start up command procedure sets this up. For example, insert these lines into the command procedure:

```
$ IF ""F$GETSYI("CLUSTER_MEMBER")"" .NES. "TRUE" - THEN $GOTO LABEL_A <RETURN>
$ ! <RETURN>
$ SET DEVICE/DUAL_PORT $10$DRA2: <RETURN>
$ ! <RETURN>
$ MOUNT/CLUSTER $10$DRA2: WORK1 <RETURN>
.
.
$ LABEL_A <RETURN>
.
.
```

CPUs not directly connected to the SI controller can access SI disks by setting the device to served by using the following command:

```
$ SET DEVICE/SERVED $10$DRA2: <RETURN>
```

CAUTION

Disk sharing between clusters or between a cluster and a non-cluster member is unsupported and results in data corruption on the shared disk.

Device names of multiported disks must be identical across the cluster. For example, a disk referred to as 1DRA0: on node A must be seen as 1DRA0: on node B. This means that a disk shared between a large VAX using SBI/CMI and a MicroVAX using a QBUS or another VAX using a UNIBUS requires the same name, 1DRA0:, as the SBI/CMI system. This naming convention is established during the question/answer section of SI_AUTOCONFIG.

6.2 MSCP Served Disks

The VAXcluster mass storage control protocol (MSCP) server makes locally connected disks available to all nodes in the cluster. However, MSCP served disks cause considerable overhead. All disk block transfers to and from the MSCP served disks must be copied into memory on the local node, and then transferred over computer interconnect (CI) or Ethernet to and from the remote nodes.

Double memory transfers are performed on both the local node and the remote node to receive or send disk blocks over CI or Ethernet. This is much slower than non-MSCP served disks, generating more overhead on the nodes and Ethernet, and is used with discretion.

Local Disks

Local disks attached to a node are not cluster accessible until they are served. Before serving any device, set the SYSGEN MSCP_LOAD parameter to 1 and reboot the system. After serving the disks and mounting them with the /CLUSTER qualifier, they are available cluster wide. For example, the commands in SYS$MANAGER:SYSTARTUP_V5.COM show how to make local disks available to the cluster:

```
$ IF ""F$GETSYI("CLUSTER_MEMBER")"" .NES. "TRUE" - THEN $GOTO LABEL_A <RETURN>
$ RUN SYS$SYSTEM:SYSGEN MSCP <RETURN>
$ SET DEVICE/SERVED $1$DRA1: <RETURN>
$ SET DEVICE/SERVED $1$DRA2: <RETURN>
.
$ MOUNT/CLUSTER $1$DRA1: WORK1 <RETURN>
$ MOUNT/CLUSTER $1$DRA2: WORK2 <RETURN>
$ GOTO LABEL_B <RETURN>
.
$ LABEL_A: <RETURN>
$ MOUNT $1$DRA1: WORK1 <RETURN>
$ MOUNT $1$DRA2: WORK2 <RETURN>
.
$ LABEL_B: <RETURN>
.
```

In the previous example, cluster related commands are bypassed if the node is not a cluster member and a normal MOUNT procedure is initiated.

Shared Disks on SI Controllers

If nodes in the cluster are not directly connected to the SI controller, you can serve shared disks. However, SI strongly recommends that all cluster nodes be directly connected to the SI controller. This avoids the overhead caused by served disks which can seriously degrade cluster performance.

6.3 System Disks

The volume label for the system disk on each node in a DEC cluster environment needs a unique name. For example, if the system disk on one node is labeled WORK1, the system disk on any other node cannot have that name. If a conflict exists, change the label using the following DCL command as an example:

```
$ SET VOLUME DRA1:/LABEL=WORK2 <RETURN>
```

When trying to boot a node with a system disk volume label that is identical to another node's system disk volume label, the following messages (or something similar) are displayed on the operator's console:

```
waiting to form or join a VAXcluster
%CNXMAN, Discovered system V780B
%CNXMAN, Established connection to system V780B
%CNXMAN, Sending VAXcluster membership request to system V780B
%CNXMAN, Now a VAXcluster member -- system V750C
%SYSINIT-E-error mounting system device, status = 007280B4
%SYSINIT-E-error opening or mapping F11BXQP, status = ...
%SYSINIT-E-message file not found, or insufficient SPT to map it
```

Similar error messages can appear on the operator's console if the system disk has the same device name as a device on another node. This can occur when using the ALLOCLASS SYSGEN parameter.

Shared Disk Naming Conventions

All disks attached to a multi-ported controller in a cluster must have unique path independent device names. VMS assigns these names using an allocation class on the nodes connected to the controller. The allocation class is an integer value used to create a disk name:

```
$ allocation_class$device_name = disk name = $10$DRA2:
```

To access disks on a shared SI controller, assign an allocation class. If there are no two local devices with the same name on any of the cluster nodes, use the SYSGEN parameter ALLOCLASS. Assign the allocation class for each node connected to the SI controller, as follows:

```
$ RUN SYS$SYSTEM:SYSGEN <RETURN>
SYSGEN> USE CURRENT <RETURN>
SYSGEN> SET ALLOCLASS n <RETURN>
SYSGEN> WRITE CURRENT <RETURN>
SYSGEN> EXIT <RETURN>
```

The value of "n" is an integer from 1 to 255. "n" is identical for every cluster node connected to the SI controller. If local devices have the same name on any two cluster nodes, ALLOCLASS causes a conflict in device names across the cluster.

6.4 Removing Nodes From a Cluster

Remove a node from a cluster by setting the SYSGEN parameter VAXcluster to 0, and rebooting the node.

CAUTION

Do not set VAXcluster to a 0 if booting from a common system disk. For a node to boot "out" of the cluster, it must boot to a dedicated system disk.

To remove an active node from a cluster, modify the SYSGEN VAXcluster parameter as follows:

```
$ RUN SYS$SYSTEM:SYSGEN <RETURN>
SYSGEN> USE CURRENT <RETURN>
SYSGEN> SET VAXCLUSTER 0 <RETURN>
SYSGEN> WRITE CURRENT <RETURN>
SYSGEN> EXIT <RETURN>
```

After you shut the node down and reboot, the node is not part of the cluster.

To remove a node that is already down, perform a conversational boot. While running SYSBOOT, set the SYSGEN parameter VAXcluster to 0, as follows:

```
SYSBOOT> SET VAXCLUSTER 0 <RETURN>
SYSBOOT> CONTINUE <RETURN>
```

The node boots, but it does not join the cluster.

NOTE

After removing a node from a cluster, adjust quorum as described in the following section.

6.5 Partitioning and Quorum

Partitioning refers to nodes associated with one cluster, operating independently as members of another cluster. With partitioning, access to the shared disks is not coordinated and data can be corrupted. The VAXcluster connection manager uses quorum to prevent cluster partitioning.

To prevent cluster partitioning when a node leaves the cluster, the cluster quorum value is not lowered by the connection manager. However, should any remaining nodes leave the cluster, lowering the quorum reduces cluster vulnerability.

Resetting Quorum

To lower the quorum after a node has left the cluster, enter the following DCL command on one of the remaining nodes:

```
$ SET CLUSTER/EXPECTED_VOTES
```

In this version QUORUM is replaced by EXPECTED_VOTES which determines how many votes a node expects to see in a cluster before it forms or joins one. The connection manager recalculates quorum using the formula (VOTES+2)/2.

Lost Quorum

When there are not enough votes to maintain a quorum, the connection manager portion of the operating system issues a message to the operator's console for all remaining nodes:

```
%CNXMAN, Lost connection to system ...
%CNXMAN, Quorum lost, blocking activity
```

At this point, the CPUs are waiting for the lost node to rejoin the cluster, and are unable to function. To recover and regain a quorum, issue a **CONTROL-P** at one of the operator's consoles. At the console prompt (>>>), issue the following commands:

```
>>> H <RETURN>
>>> D/I 14 C <RETURN>
>>> C <RETURN>
IPC> Q <RETURN>
IPC> <CONTROL-Z> <RETURN>
$
```

This procedure causes the connection manager to recompute quorum and free the cluster.

Quorum Disks

In a cluster, a quorum disk functions as a virtual node, contributing a specified number of votes to the quorum. The quorum disk device name is specified in the SYSGEN parameter DISK_QUORUM, and the votes assigned to that disk are specified in the QDSKVOTES parameters.

A quorum disk increases the cluster availability in an even number node cluster. If half the nodes are removed from the cluster, there are still enough votes available to maintain a quorum on the remaining nodes.

NOTE

Any disk configured at boot time may be used as a quorum disk connected to one or more nodes in the cluster.

6.6 OPCOM Messages

OPCOM messages appear occasionally on each operator's console, and are not a cause for concern. If the message is not from the console's own node, the source of the message is given. For example:

```
%%% OPCOM 17-JAN-1987 23:54:12.55 %%% (from node V750C...)
Request 1243, from user MELISSA on V750C
Please mount tape...
```

VMS SHUTDOWN messages echo across the cluster to all attached terminals. For example, the following SHUTDOWN message might appear on the terminals attached to VAXcluster member V780B, when node V780A is shutting down:

```
SHUTDOWN message on V780B from user SYSTEM at _V780A...
V780A will shut down in 5 minutes; back up later...
Please log off node V780A.
```

The system manager can disable these messages by defining the logical SHUTDOWN$INFORM_NODES as 1 instead of 0.

7
MAINTENANCE AND TROUBLESHOOTING

This section provides system verification programs and troubleshooting checklists, along with possible error messages and problem indications.

7.1 System Verification

To verify system operation, run the SI disk exercising program EXOR, as specified in *EXOR VAX Disk Exerciser User Guide.* For drive diagnostics, refer to *SIDOS User Guide.*

7.2 System Maintenance

Under normal operating conditions, the software modifications do not require operator maintenance.

7.3 System Troubleshooting

Use the following checklist for system troubleshooting:

- ☐ 1. Confirm that the system is receiving power. If not, check power cables and fuses.
- ☐ 2. Check for loose cables, poorly attached connectors, and bent pins.
- ☐ 3. Check jumpers, switch settings, and the computer system configuration file.
- ☐ 4. Run the internal diagnostics.
- ☐ 5. Run the external diagnostics.
- ☐ 6. Verify the peripherals.
- ☐ 7. Run EXOR to determine if the system is working.

Cluster Troubleshooting Checklist

This checklist provides a guide to the SI field engineer or the VMS cluster manager for troubleshooting the system.

- ☐ 1. Are the disks on the multi-ported controller manually connected on all VAX's with a UNIBUS or QBUS connection?
- ☐ 2. What is the allocation class of the shared disks on the multi-ported SI controller? Determine the value of the SYSGEN parameters USER3, USER4, and ALLOCLASS for each VAX in the cluster.

☐ 3. Are the shared disks set for dual porting and mounted with the cluster qualifier on all VAX's with a direct connection to the multi-ported controller?

☐ 4. Has the cluster configuration and access for all local and cluster accessible disks been determined? To set up, name, mount and manage disk devices in the cluster. Refer to the *VMS VAXcluster Manual* and Section 6 of this manual. Note that all cluster accessible and shared volumes must have unique volume labels across the cluster.

☐ 5. Does the system disk of each node in the cluster have a unique device name and volume label across the cluster? If the SYSGEN parameter ALLOCLASS is used, then the device name is preceded by the allocation class. Use the SYSGEN parameter USER3 if necessary.

☐ 6. Have the values of the SYSGEN parameters VOTES and EXPECTED_VOTES been determined for each node in the cluster, based on the VAXcluster configuration and the user applications? The VAXcluster software uses a quorum scheme to maintain the integrity of the cluster.

For a two node cluster, a quorum disk is desirable to increase system availability. Refer to the *VMS VAXcluster Manual.*

☐ 7. Has the system manager prepared for the cluster environment? This is discussed in the *VMS VAXcluster Manual*, and includes: shared system files, system command procedures, and cluster-wide, batch and print queues.

In addition, user applications need to be set up to run in the cluster environment.

☐ 8. Are all nodes in the cluster running version V5.x of VMS? Unless it is a rolling update of VMS, DEC recommends that all VAX's in a cluster run the same version of VMS.

Error Messages

Error messages can appear on the terminal during automatic or manual software installation. For a detailed description of error messages associated with automatic or manual installation, refer to the *VMS System Messages and Recovery Procedures Reference Manuals.* This section discusses two of the most common error messages.

If the automatic software installation is having trouble, the following message, or one similar to it can appear:

```
**********************************************************************
*                                                                    *
*>>>>>>>>>>>>>>>>>> ERROR CONFIGURING SYSTEM <<<<<<<<<<<<<<<<<<*
*                                                                    *
**********************************************************************
```

If more than half of the installation procedure has been completed, use the manual menu to perform the remaining uninstalled sections in sequential order. Retain the distribution cartridge or diskette in the console.

This message indicates which component failed:

```
**********************************************************************
*                                                                    *
*>>>>>>>>>>>>>>>>>>>> ERROR IN THIS SECTION <<<<<<<<<<<<<<<<<<<<*
*                                                                    *
**********************************************************************
```

After resolving the problem, that component section is reinstalled using manual mode.

APPENDIX A
CLUSTOR/9900 MICROVAX II BOOT

Clustor/9900 MicroVAX II boot allows booting a MicroVAX II using a locally connected SI disk attached to a Clustor or 9900 controller. This feature is an addition to the already existing capability of booting an SI MSCP compatible disk, and should only be installed by MicroVAX II users who require it. Other users can skip this section.

The MicroVAX boot software is contained in a separate saveset on the distribution media. This software is installed using VMSINSTAL in a similar manner to the Software Modification Components explained earlier.

VMSINSTAL places the MicroVAX boot software onto a user specified Intermediate Br Device (IBD). An example of an IBD is the TK70 tape drive or the RX50 disk drive. user boots the IBD using the standard boot command syntax at the console. Contr is then passed to the SI boot software located on the IBD, and the boot driver for the Clustor/9900 disk is loaded. VMS is then booted from any disk on the Clustor/9900.

A.1 Configurations

A typical configuration of where the Clustor/9900 MicroVAX boot feature is most useful is shown in Figure A-1. DRA0 is the common system disk for the VAXCluster. This configuration is recommended because the SBI/CMI connection to the Clustor/9900 makes major VMS upgrades easier. Configurations without the SBI/CMI connection are possible. Refer to "VMS Upgrades" later in this section for procedural details.

In this configuration, install the Clustor/9900 MicroVAX boot feature by running VMINSTAL on both MicroVAX A and MicroVAX B, and boot each IBD. Note that the root numbers are set as part of the installation procedure.

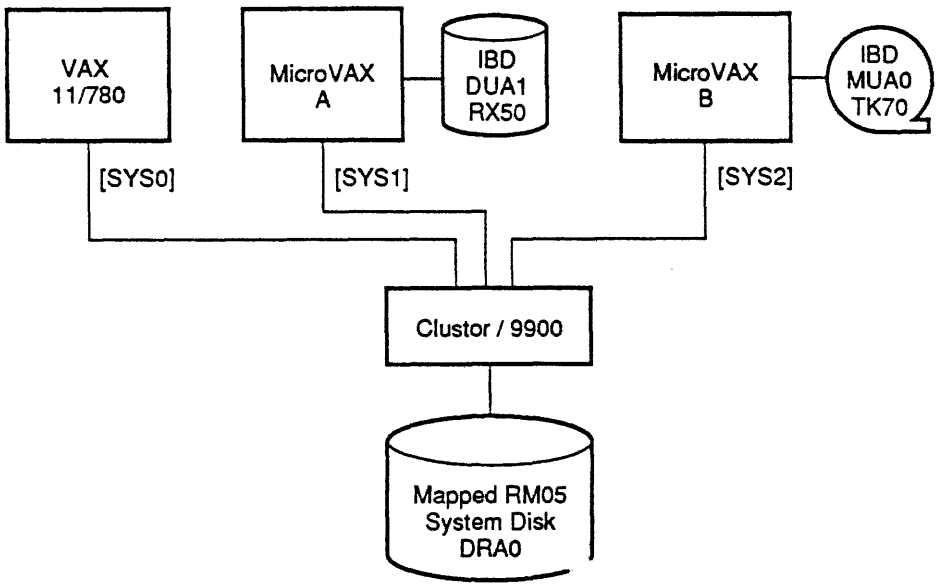

Figure A-1. Clustor/9900 MicroVAX II Configuration

A.2 Installation

In a VAXCluster, the MicroVAX boot feature is installed on EACH MicroVAX node that is to boot the CLUSTOR/9900-based disk. This means that each node has a local IBD. The IBD may be any supported MicroVAX boot device.

It is necessary to run SI_AUTOCONFIG.COM on each node in order to configure the attached Clustor/9900 controller as DRA or DRB. The Clustor/9900 MicroVAX boot feature does not support booting with controller designators other then DRA or DRB. The "DR" controller naming convention was chosen for compatibility with Massbus connections from other machines to the same controller.

The MicroVAX boot software is installed onto a user specified IBD and consists of a single executable image. During the installation, the user is prompted for the following items:

1. Name of the IBD.
2. Name of the Clustor/9900-based system disk.
   NOTE: Only the A and B controllers are supported.
3. Default system root on that Clustor/9900-based disk.

Finally, the installation procedure initializes the IBD, and automatically places this special intermediate boot program into the root directory of the IBD.

1. Invoke VMSINSTAL by typing:

   $ @SYS$UPDATE:VMSINSTAL SIUVB MUA0 <RETURN>

   MTA0 is the mnemonic for the device being used. The actual device mnemonic can be different.

   The following message is displayed:

   ```
   VAX/VMS Software Product Installation Procedure V5.2

   It is 26-OCT-1989 at 16:22.
   Enter a question mark (?) at any time for help.
   ```

2. A series of questions follow that must be answered either YES, NO, or ? for help:

   ```
   * Are you satisfied with the backup of your system disk [YES]?
   ```

   This installation places software on the IBD, not the system disk. Ignore this message.

3. The installation begins, displaying the following:

   ```
   The following products will be processed:

   SIUVB V5.2

   Beginning installation on SIUVB V5.2 at 16:23

   %VMSINSTAL-I-RESTORE, Restoring product saveset A ...
   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
   !                              SYSTEM INDUSTRIES                               !
   !                                                                              !
   !          MicroVAX II Intermediate Boot Device Installation Procedure        !
   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
   !   This installation procedure installs the SYSTEM INDUSTRIES intermediate boot   !
   !   program on a MicroVAX II bootable device. Booting this device, the Intermediate Boot   !
   !   Device (IBD) will allow users to boot MicroVAX II's directly from a System Industries   !
   !   Clustor/9900 controller. After completion of this procedure, boot the IBD from the   !
   !   MicroVAX Console. The IBD will, in turn, boot the Clustor/9900 based system disk.   !
   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
   ```

Enter Clustor/9900 boot device (DRA0):

Enter: **name of the Clustor/9900-based system disk <RETURN>**
Note: DRA0 is the default.

4. The following message is displayed:

   Enter system root number (0):

   Enter: **root number which to boot <RETURN>**

5. The following message is displayed:

   Enter intermediate boot device (MUA0): DUB0

   Enter: **name of the IBD <RETURN>**

6. On completion of VMINSTAL, the following message is displayed:

   Ending time 26-OCT-1989 16:24:36.02

   Starting time 26-OCT-2989 16:23:45.58

   The installation is complete.

   %VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories...

   Installation of SIUVB V5.2 completed at 16:24

   VMSINSTAL procedure done at 16:24

A.3 Operation

The user boots the IBD using standard boot command syntax at the system console. The program on the IBD is given control after the completion of MicroVAX ROM VMB. Next, the IBD program locates and boots the Clustor/9900 controller-based system disk. Normal boot flag capability is supported using standard boot commands at the initial system console prompt (>>>). For example:

B	to boot the disk and root specified at installation
B/R5:00001	to stop at SYSBOOT on the Clustor/9900 disk.
B/R5:00006	to boot XDELTA on the Clustor/9900 disk.

The capability is provided to boot a disk other than that specified at installation time. As with standard MicroVAX boot commands, it is necessary to specify the root number. In order to boot an alternate Clustor/9900-based disk on root zero, enter:

B/R5:80000 at the initial system console prompt.

When the MicroVAX is booted in this manner, booting stops during the execution of the SI boot software on the IBD. The user is then prompted for an alternate Clustor/9900-based disk and the system is booted using that disk.

The name entered must be the same as that configured using SI_AUTOCONFIG.COM and that disk must have the same name on all nodes in the VAXCluster. In order to boot an alternate Clustor/9900-based disk on root two, enter:

B/R5:20080000 at the initial system console prompt.

A.4 VMS Upgrades in MicroVAX Boot Environment

Using the recommended configuration has the advantage that all VMS upgrades can be performed directly on the Clustor/9900-based system disk. However, in Cluster configurations that do not provide an SBI/CMI connection to a software transparent (mapped RM05) Clustor/9900-based system disk, upgrades are performed using the following procedure.

There are two types of VMS upgrades: those that require phased reboots and those that do not. Generally, remastered VMS updates require phased reboots. When performing this type of upgrade, it is first necessary to move the system from the Clustor/9900-based disk to an MSCP compatible disk (SI or DEC) by means of a BACKUP.

The upgrade is performed on that disk and then the system is moved from the MSCP disk back to the Clustor/9900 disk. This is necessary because the phased upgrade does not recognize the Clustor/9900-based system disk during all the reboot phases.

Upgrades that do not require phased reboots may always be done without moving the system to an MSCP compatible disk. The upgrade may be applied directly to any Clustor/9900-based system disk.