digital

KAV30

Hardware Installation and User's Information

Order Number: AA-PFM6C-TE

KAV30

Hardware Installation and User's Information

Order Number: AA-PFM6C-TE

August 1993

This guide describes how to install and operate the KAV30 hardware.

Revision Information:

This manual supersedes the *KAV30 Hardware Installation and User's Information, Version 1.1* (Order Number AA-PFM6B-TE).

Digital Equipment Corporation
Maynard, Massachusetts

August 1993

The postpaid Reader's Comments forms at the end of this document request your critical evaluation to assist in preparing future documentation.

This document was prepared using VAX DOCUMENT, Version 2.1.

Contents

4 KAV30 Hardware

5 Operating the KAV30

6 Replacing the KAV30 User ROM

A KAV30 Hardware Specifications

B Diagnostic Tests

C VAXELN Pascal Program

D KAV30 Port Connectors

Glossary

Index

Figures

Tables

Preface

This guide describes how to install and operate the KAV30 hardware.

Who Should Read This Guide

This guide is intended for anyone who wants to install or operate the KAV30. However, to use this guide effectively, you must be familiar with the basic backplane hardware concepts and the VMEbus environment.

Structure of This Guide

This guide is divided into six chapters, three appendixes, a glossary, and an index:

- Chapter 1 describes how to prepare to install the KAV30.
- Chapter 2 describes how to install the KAV30.
- Chapter 3 describes how to test the KAV30.
- Chapter 4 describes the KAV30 hardware.
- Chapter 5 describes how to operate the KAV30.
- Chapter 6 describes how to replace the KAV30 user read-only memory (ROM).
- Appendix A gives the KAV30 hardware specifications.
- Appendix B describes the KAV30 diagnostic tests.
- Appendix C contains a sample VAXELN Pascal program to read diagnostic error information.
- Appendix D describes the KAV30 port connectors.
- The Glossary defines some of the important terms used in this guide.

Associated Documents

For more information, see the following documents:

- *KAV30 Hardware Cover Letter* (AV-PFSSB-TE)
- *KAV30 Hardware Release Notes* (AV-PJRKA-TE)
- *KAV30 Software Cover Letter* (AV-PEYFB-TE)
- *KAV30 Software Product Description* (AE-PFB5B-TE)
- *KAV30 System Support Addendum* (AE-PFB6B-TE)
- *VAXELN KAV30 Programming Guide* (AA-PRFBA-TE)

Related Documents

For additional information, see the following documents:

- *VMEbus Specification, Revision C.1* (PRINTEX, Phoenix, AZ, USA)
- *The VME Subsystem Bus (VSB) Specification, Revision B.1* (Motorola®, Phoenix, AZ, USA)
- *Small Computer Systems Interface Developer's Guide* (Digital™, EK-SCSIS-SP-001)
- *rtVAX 300 Hardware User's Guide* (Digital, EK-382AA-UG)
- *VAXELN Utilities Guide* (Digital, AA-PGXDA-TE)
- *VAXELN System Development Guide* (Digital, AA-PGXCA-TE)

Conventions

The following conventions are used in this guide:

Convention	Description
Note	A note contains information that is of special importance to the reader.
Caution	A caution contains information that the user needs to know to avoid damaging the software or hardware.
Warning	A warning contains information that is essential to people's safety.
Ctrl/*x*	Ctrl/*x* indicates that you hold down the Ctrl key while you press another key (indicated here by *x*).

Convention	Description
[]	Brackets enclose optional command arguments in command descriptions. Brackets are also used in the syntax of a directory name in a VMS file specification.
italic type	Italic type emphasizes important information and indicates the complete titles of manuals.
boldface type	Boldface type indicates the first occurrence of terms defined either in text, in the glossary, or both.
Monospace Type	Monospace type indicates both system displays and user input.
n.nn	A period in numerals signals the decimal point indicator. For example, *1.75* equals *one and three-fourths*.
nn nnn.nnn nn	A space character separates groups of 3 digits in numerals with 5 or more digits. For example, *10 000* equals *ten thousand*.
UPPERCASE	Words in uppercase indicate software commands, the name of a file, or hardware signal lines.
<n..n>	Three or more consecutive signal line numbers are enclosed in angle brackets. The first line number is separated from the last line number with two periods *(..)*. For example, signal lines *<1..4>* represent signal lines *1, 2, 3,* and *4*.

1

Preparing to Install the KAV30

This chapter describes how to prepare to install the KAV30. It gives information on the following:

- Physical and environmental requirements (see Section 1.1)
- Unpacking instructions (see Section 1.2)
- KAV30 orientation (see Section 1.3)

1.1 Physical and Environmental Requirements

This section describes the physical and environmental requirements for installing the KAV30. The KAV30 is a double-height[1] VMEbus module. To satisfy physical and environmental requirements, follow these steps:

1. Use a crate that has the following characteristics:
   - The crate is a double-height crate.
   - The crate contains a double-height VMEbus backplane.
   - The crate is conductive.
   - The crate is connected to earth.

Caution

If you do not house the KAV30 in a conductive crate connected to earth, you risk the following:

- Malfunction in the KAV30
- Damage to the KAV30

[1] VMEbus documents often use the term 6U to describe double-height modules.

- The crate is product safety compliant with the local regulations such as the UL1950, EN60950, and CSA950 standards that include the Safety Extra Low Voltage (SELV) requirements.

Warning

If you do not house the KAV30 in a product safety compliant crate, you risk personal injury, fire, or damage to equipment.

- The power supply to the crate is product safety compliant with the local regulations such as the UL1950, EN60950, and CSA950 standards that include the SELV requirements.

Warning

If the power supply to the crate is not product safety compliant, you risk personal injury, fire, or damage to equipment.

2. Use a VMEbus backplane that has the following characteristics:
   - The J1 and J2 backplane connectors must supply power to the modules with which they mate.

Warning

If the J2 backplane connector on the VMEbus backplane does not supply power to the modules, there is an overload of power on the J1 backplane connector. This power overload can cause personal injury, fire, or damage to equipment.

   - The J2 backplane connector contains 64 user-defined pins. The KAV30 uses these pins to communicate with a VME subsystem bus (VSB) backplane. Connect only VSB devices to these pins.

Caution

If you connect non-VSB devices to the J2 backplane connector, you risk damaging the VSB interface on the KAV30.

3. Before you install the KAV30, ensure that the VMEbus system configuration does not exceed the system's limits for power supply and bus loads.

Caution

If the VMEbus system configuration exceeds the system's limits for power supply and bus loads, an equipment malfunction results.

To determine the VMEbus system's limits for power supply and bus loads, see the documentation for the system components. See Appendix A for the KAV30 power supply and bus loads specifications.

Caution

Digital strongly recommends that you operate the module in creates or environments with a forced cooling of at least 100 lfm. If you do not follow this advice, the MTBF of your module may degrade significantly.

1.2 Unpacking Instructions

This section describes how to unpack the KAV30 hardware kit. To unpack the KAV30 hardware kit, follow these steps:

1. Look for external damage on the shipping container. If there is external damage contact your Digital representative.
2. Do not dispose of the packing material until you successfully install the KAV30. You might want to repack the KAV30 and return it to your Digital representative. Keep the antistatic bag for storing or transporting the KAV30. If you are a self-maintenance customer, keep the packing material.
3. To protect the KAV30 from ESD, wear an antistatic wrist strap while handling the KAV30. Place the KAV30 on an antistatic mat.

Caution

If you do not wear an antistatic wrist strap while handling the KAV30, or if you do not place the KAV30 on an antistatic mat, you risk damaging the KAV30.

4. Unpack the KAV30 hardware kit.

5. Use the shipping list to verify that all the kit items are present. If some items are missing, contact your Digital representative.

   Table 1–1 lists the KAV30 hardware kit items.

Table 1–1 KAV30 Hardware Kit Items

Item	Part Number
Shipping material	99-05016-01
Antistatic bag	99-07092-05
Antistatic bag seal	36-16159-00
KAV30 module	T6010-xx†
SCSI port ESD cover	2G-1908A-01‡
KAV30 Hardware Installation and User's Information	AA-PFM6C-TE
KAV30 Hardware Cover Letter	AV-PFSSB-TE
KAV30 Hardware Release Notes	AV-PJRKA-TE
VAXELN/KAV30 Run-time License Letter	AG-PG5BB-TE

†A lowercase x denotes a character that depends on the version of the module. See Table 1–2 for information about versions of the KAV30 module.
‡The kit contains this item only if the KAV30 has a SCSI port.

There are four KAV30 module options. Table 1–2 lists the part number and gives a brief description of each module.

Table 1–2 KAV30 Module Options

Module Option	Part Number	Description
KAV30-AA	T6010-AA	Module with 4M bytes of system RAM
KAV30-AB	T6010-BA	Module with 16M bytes of system RAM
KAV30-AC	T6010-CA	Module with 4M bytes of system RAM and a SCSI port

(continued on next page)

Table 1–2 (Cont.) KAV30 Module Options

Module Option	Part Number	Description
KAV30-AD	T6010-DA	Module with 16M bytes of system RAM and a SCSI port

6. Inspect the KAV30 for shipping damage. If there is shipping damage contact your Digital representative.

Depending on how you want to use the KAV30, you might want one or more hardware items that are not part of the KAV30 hardware kit. Table 1–3 lists these items.

Table 1–3 Additional KAV30 Hardware Installation Items

Item	Supplier	Part Number
DEC™ 423 serial line cable	Digital	BC16E-xx†
Ethernet cable	Digital	BNE4C-xx†
Ethernet loopback connector	Digital	12-22196-01
DELNI™ Ethernet transceiver	Digital	DELNI-xx†
DESTA™ Ethernet transceiver	Digital	DESTA-xx†
H4000 Ethernet transceiver	Digital	H4000-xx†
H4005 Ethernet transceiver	Digital	H4005-xx†
SCSI cable	Digital	BC56H-xx†

†A lowercase x denotes a variable character in a part number. Variable characters in cable part numbers depend on the length of the cable. Variable characters in other part numbers depend either on the version of the item, or on the country in which you order the item.

1.3 KAV30 Orientation

This section describes the orientation of the KAV30. The KAV30 is a double-height VMEbus module. Figure 1–1 shows the KAV30 front panel, component side, and backplane connectors. This guide uses these terms as points of reference.

Figure 1–1 KAV30 Orientation

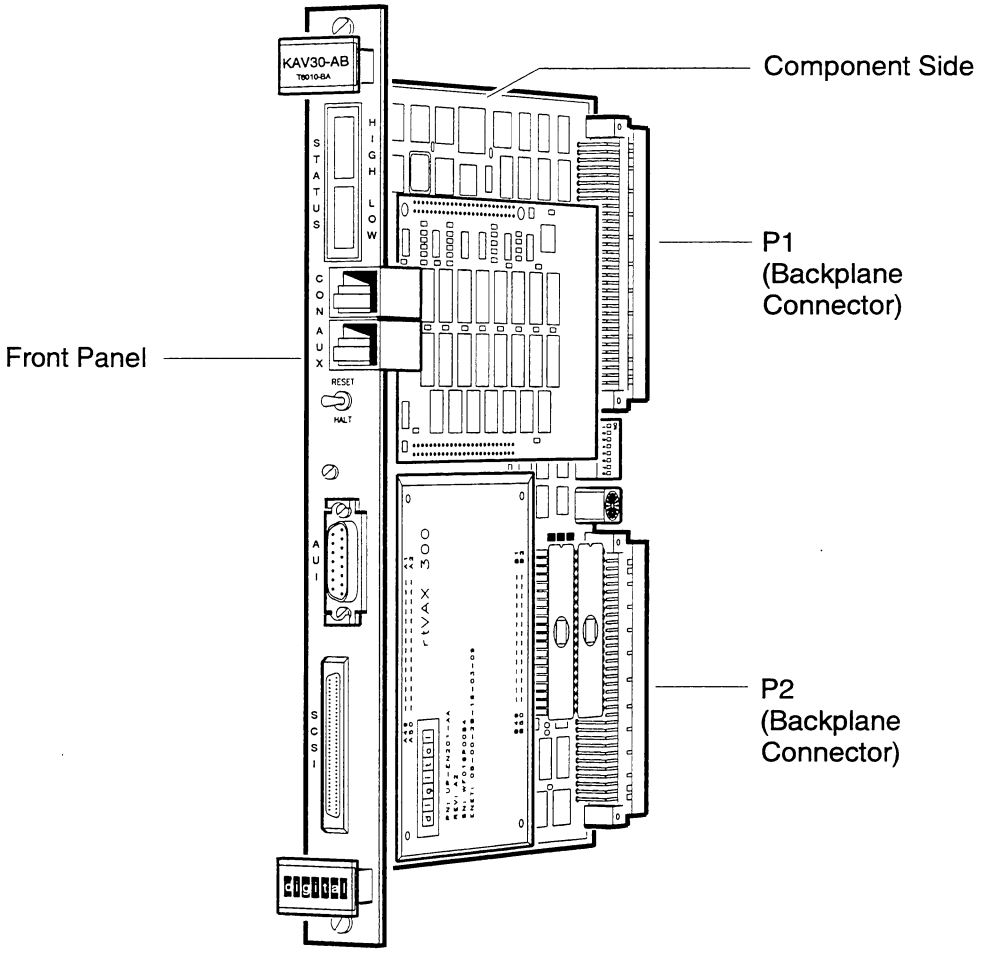
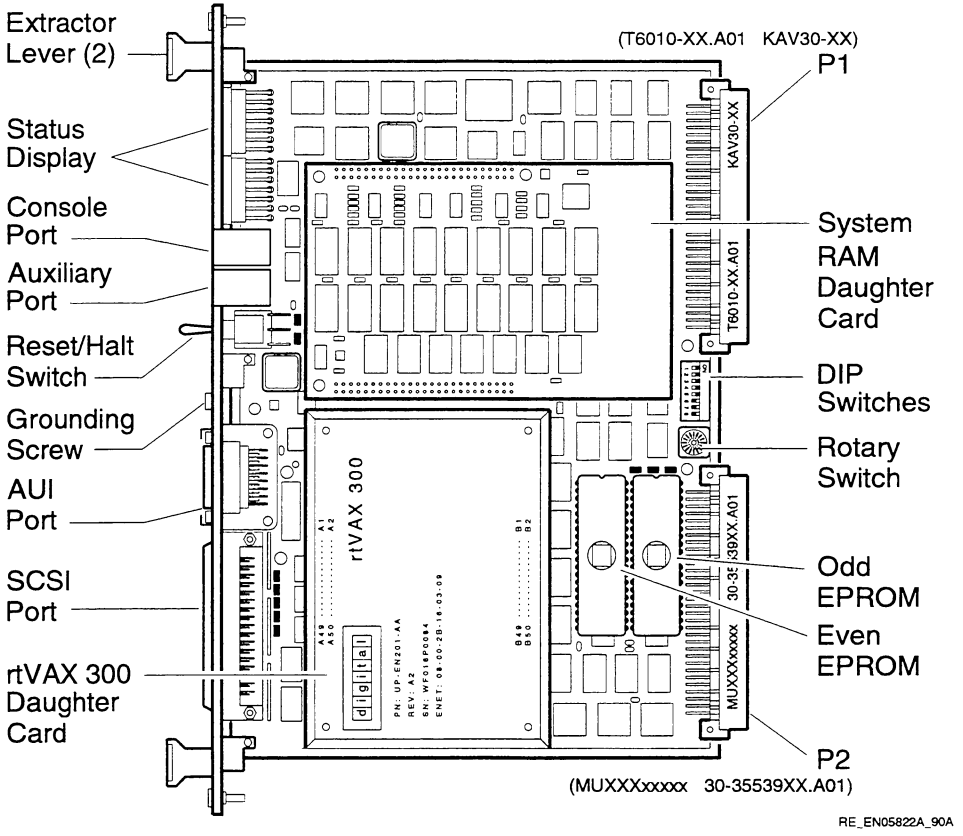


Figure 1–2 shows the component side of the KAV30 and its parts. This figure also shows the location of the KAV30 revision information and serial number.

Figure 1–2 KAV30 Parts



2
Installing the KAV30

This chapter describes how to install the KAV30. It gives information on the following:

- Beginning the installation (see Section 2.1)
- Configuring the VMEbus backplane (see Section 2.2)
- Configuring the VSB backplane (see Section 2.3)
- Configuring the KAV30 (see Section 2.4)
- Inserting the KAV30 into a VMEbus crate (see Section 2.5)
- Connecting devices to the serial line ports (see Section 2.6)
- Connecting devices to the attachment unit interface (AUI) port (see Section 2.7)
- Connecting devices to the small computer systems interface (SCSI) port (see Section 2.8)
- Finishing the installation (see Section 2.9)

2.1 Beginning the Installation

This section describes how to begin installing the KAV30. To begin installing the KAV30, follow these steps:

1. To protect the KAV30 from ESD, wear an antistatic wrist strap while handling the KAV30. Place the KAV30 on an antistatic mat.

Caution

If you do not wear an antistatic wrist strap while handling the KAV30, or if you do not place the KAV30 on an antistatic mat, you risk damaging the KAV30.

2. Back up all the software and data on the VMEbus system.
3. Set the power switches on the SCSI devices connected to the VMEbus system to the off position.
4. Set the power switches on the devices connected to the VMEbus system to the off position.
5. Set the power switch on the VMEbus crate to the off position.
6. Choose the backplane slot in which you want to insert the KAV30. If you want the KAV30 to act as the system controller, choose the leftmost backplane slot (slot 1).

2.2 Configuring the VMEbus Backplane

This section describes how to configure the VMEbus backplane. The VMEbus backplane configuration depends on the configuration of the KAV30 you want to install and on the backplane slot in which you want to install the KAV30. See the VMEbus crate and backplane documentation for information on configuring the VMEbus backplane.

2.3 Configuring the VSB Backplane

This section describes how to configure the VSB backplane. If you want to use the KAV30 with a VSB backplane and you do not have a VSB backplane in the VMEbus crate, connect one now. See the VSB backplane documentation for information on connecting the VSB backplane to the crate.

The VSB backplane configuration depends on the configuration of the KAV30 you want to install and on the backplane slot in which you want to install the KAV30. See the VMEbus crate and VSB backplane documentation for information on configuring the VSB backplane.

2.4 Configuring the KAV30

This section describes how to configure the KAV30. You can use a combination of jumpers, DIP switches, and a rotary switch to configure the KAV30. The KAV30, as it is shipped, has the following default hardware configuration:

- The user ROM size is 256K bytes
- The KAV30 responds to the VMEbus RESET signal
- The KAV30 does not supply power to the SCSI bus TERMPWR signal
- Break key assertions on devices connected to the auxiliary port do not assert the VAX™ HALT signal

- Ethernet devices can assert the VAX HALT signal
- There is no power source for the battery backed-up devices
- The boot source is the Ethernet
- The KAV30 allows Ethernet trigger boots
- The KAV30 is not the VMEbus arbiter
- Break key assertions on devices connected to the serial line ports assert the VAX HALT signal
- The VMEbus AC_FAIL signal asserts the VAX POWER_FAIL signal
- The VMEbus A24 base slave address is zero

The rest of this section describes how to use the KAV30 jumper posts, DIP switches, and the rotary switch to change the KAV30 hardware configuration.

2.4.1 Jumper Posts

The KAV30 contains jumper posts that allow you to configure the KAV30 functionality.

Caution

Read this section carefully to ensure that you configure the KAV30 correctly. If you do not configure the KAV30 correctly, you risk causing a malfunction in the module.

Figure 2–1 shows the location of the jumper posts. You can use the jumper posts to configure the following:

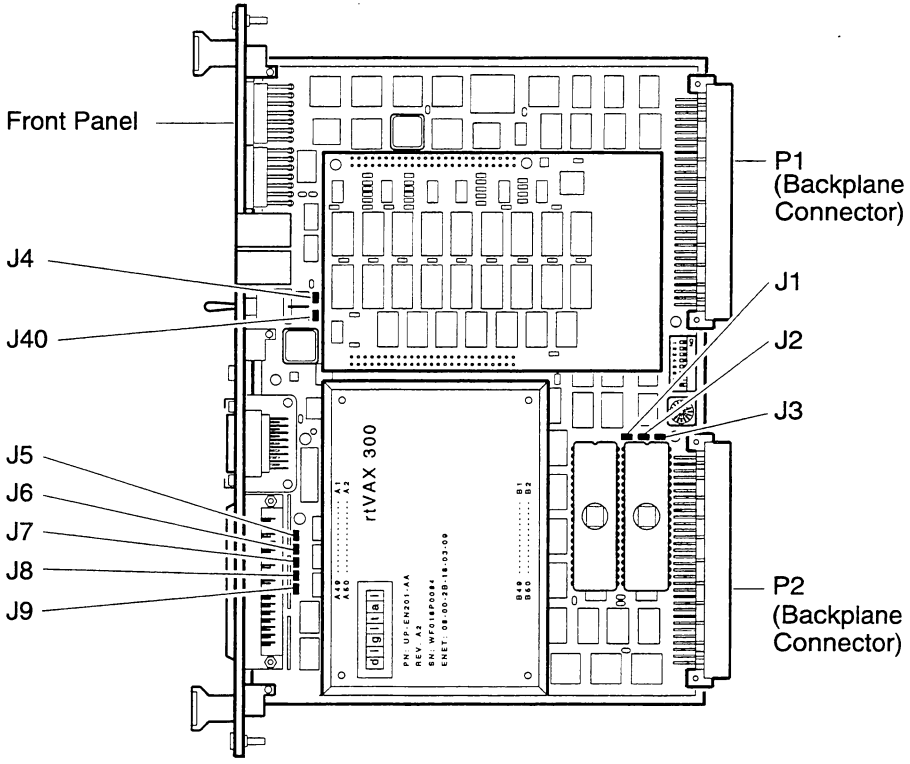
- The user ROM size

  The KAV30 can have 128K bytes, 256K bytes, 512K bytes, or 1M byte of user ROM. The KAV30 uses two erasable programmable read-only memories (EPROMs) to implement the user ROM. The J1 and J3 jumper posts determine the user ROM size (the jumper settings are the same for user ROM sizes of 128K bytes, 256K bytes, and 512K bytes). Table 2–1 describes how to configure these jumper posts.

Table 2–1 Settings for User ROM Size Jumper Posts

User ROM Size	J1 Jumper Posts	J3 Jumper Posts
128K bytes, 256K bytes, or 512K bytes	Jumper does not connect the posts	Jumper does not connect the posts
1M byte	Jumper connects the posts	Jumper connects the posts

Figure 2–1 KAV30 Jumper Post Locations



- The VMEbus RESET signal

  The VMEbus RESET signal resets the entire VMEbus system. If you want the KAV30 to drive the VMEbus RESET signal (that is, the signal is an output from the KAV30), ensure that a jumper connects the J7 jumper posts. If you want the KAV30 to respond to the VMEbus RESET signal (that is, the signal is an input to the KAV30), ensure that a jumper does not connect the J7 jumper posts. Note that the KAV30 can either drive this signal or respond to it, but it cannot both drive and respond to it. The KAV30 always drives this signal when you turn on the system.

- The SCSI bus TERMPWR signal

  A SCSI bus must have at least one source of power for its TERMPWR signal.

  If you want the KAV30 to supply power to the TERMPWR signal, ensure that a jumper connects the J5 jumper posts. If the KAV30 does not connect to a SCSI bus, or if you do not want to supply power to the SCSI bus, ensure that a jumper does not connect the J5 jumper posts.

- Break key asserts the VAX HALT signal

  The VAX HALT signal halts the KAV30 CPU.

  If you want to allow a break key assertion on a device connected to the auxiliary port to assert the KAV30 VAX HALT signal, ensure that a jumper connects the J6 jumper posts. If you do not want to allow a break key assertion on a device connected to the auxiliary port to assert the KAV30 VAX HALT signal, ensure that a jumper does not connect the J6 jumper posts.

- Ethernet device asserts the VAX HALT signal

  If you want to allow a device, which boots the KAV30 via the Ethernet (for example, a remote trigger), to assert the VAX HALT signal on the KAV30, ensure that a jumper connects the J8 jumper posts. If you do not want to allow a device, which boots the KAV30 via the Ethernet, to assert the VAX HALT signal on the KAV30, ensure that a jumper does not connect the J8 jumper posts.

- Power source for the battery backed-up devices

  The KAV30 has two battery backed-up devices: the battery backed-up RAM and the calendar/clock. The power source for these battery backed-up devices is called the **auxiliary power source**. You can configure the auxiliary power source as follows:

  - The KAV30 battery is the auxiliary power source
  - The VMEbus standby power supply is the auxiliary power source

The J9 jumper posts determine the auxiliary power source. There are three J9 jumper posts. Table 2–2 describes how to configure these jumper posts.

Table 2–2 Settings for Auxiliary Power Source Jumpers

Auxiliary Power Source	Two Posts Closest to J8	Two Posts Furthest from J8
VMEbus standby power supply	Jumper does not connect the posts	Jumper connects the posts
KAV30 battery	Jumper connects the posts	Jumper does not connect the posts

Caution

You must configure the KAV30 to provide standby power from the on-board battery or from an external standby power supply to prevent a malfunction of the KAV30.

Digital recommends that the auxiliary power source is always connected to select the on-board battery when the KAV30 is installed in your VME crate. In those cases where standby power is required and you have connected the auxiliary power source to standby power, you must supply the appropriate voltage to the standby power connection at the backplane. If you do not adhere to these guidelines, there is a possibility that the module will not operate correctly.

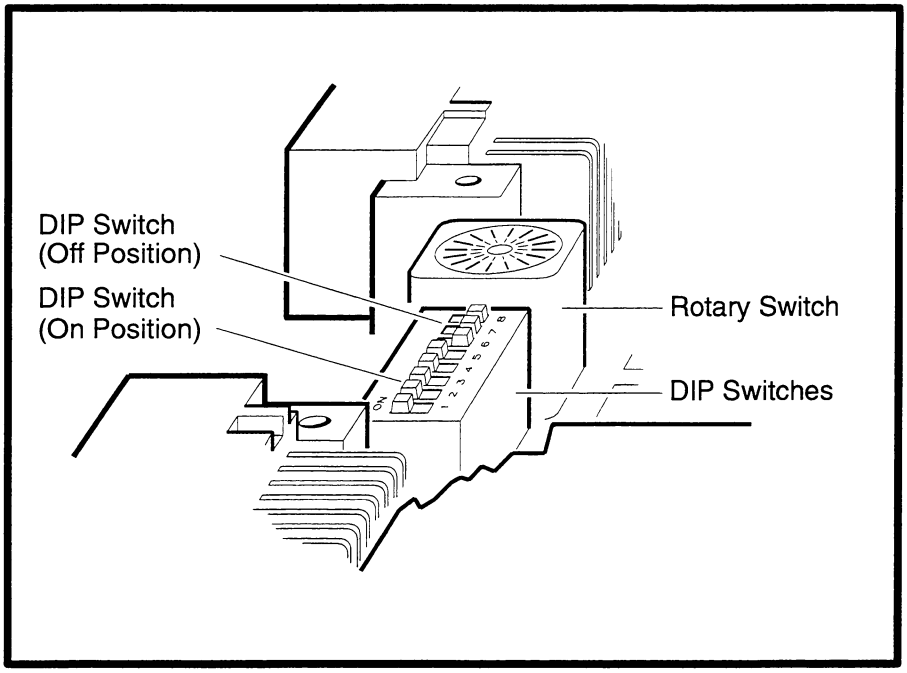
The module is shipped with the auxiliary power source jumper-connected to standby. You should change the jumper to the battery position prior to installing the module in your VME crate.

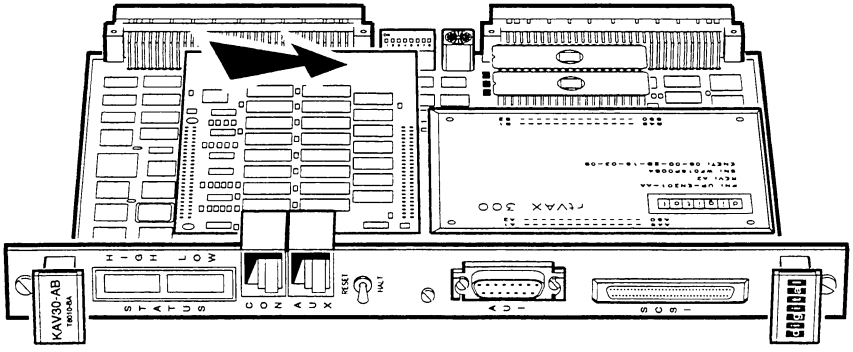
The KAV30 also contains jumper posts that are reserved for Digital. These jumper posts are J2, J4 and J40.

2.4.2 DIP Switches

A packet of eight DIP switches allows you to configure the KAV30. You can put each DIP switch in the on position or in the off position. Figure 2–2 shows a KAV30 DIP switch orientation. In Figure 2–2 the arrow shows the view from which the DIP switches are illustrated.

Figure 2–2 KAV30 DIP Switch Orientation





RE_EN06018A_90

You can use the DIP switches to configure the following:

- The boot source

  Switches 1 to 3 set the default KAV30 boot source. Table 2–3 lists the KAV30 boot source settings for these switches.

Table 2–3 Settings for Boot Source Switches

Switch 1	Switch 2	Switch 3	Boot Source
Off	Off	Off	Ethernet (EZA0)
On	Off	Off	Device connected to the auxiliary port that operates at 9600 bits per second (bits/s) (CSB2)†
Off	On	Off	Device connected to the auxiliary port that operates at 2400 bits/s (CSB1)†
On	On	Off	Device connected to the auxiliary port that operates at 1200 bits/s (CSB0)†
Off	Off	On	SCSI device or application program in the KAV30 user ROM (PRB1)‡
On	Off	On	User ROM (PRB0)
Off	On	On	Undefined
On	On	On	No boot

†The KAV30 performs a DECnet™ DDCMP™ boot.

‡The KAV30 first copies the contents of the user ROM to the system RAM.

- Whether or not to allow Ethernet trigger boots

  If you want to allow a node on the Ethernet to perform a remote KAV30 trigger boot, move switch 4 to the on position. If you do not want to allow a node on the Ethernet to perform a remote trigger, move switch 4 to the off position.

- The VMEbus arbiter

  If you want the KAV30 to act as the VMEbus arbiter, move switch 5 to the on position. If you do not want the KAV30 to act as the VMEbus arbiter, move switch 5 to the off position.

- Break key asserts the VAX HALT signal

  If you want a break key assertion on a device, which is connected to the console port, to assert the KAV30 VAX HALT signal, move switch 6 to the off position. If you do not want a break key assertion on a device, which is connected to the console port, to assert the KAV30 VAX HALT signal, move switch 6 to the on position.

- The VMEbus AC_FAIL signal

  If you want the VMEbus AC_FAIL signal to assert the VAX POWER_FAIL signal on the KAV30, move switch 8 to the off position. If you do not want the VMEbus AC_FAIL signal to assert the VAX POWER_FAIL signal on the KAV30, move switch 8 to the on position.

- Whether or not to boot after a fatal error

  Starting with EPROM revision A02, DIP switch 7 is defined as Attempt Boot on Fatal Error. All modules are shipped with this switch in the off position. The actions Halt on Fatal Error and Attempt Boot on Fatal Error are described in detail in Section 3.2. To identify the EPROM version and revision of your module, see the *KAV30 Hardware Release Notes*, order number AV-PJRKA-TE. The *KAV30 Hardware Release Notes* are shipped with every module.

  Note

  The implementation of the action Attempt Boot on Fatal Error differs from module revision levels A to B and higher.

2.4.3 Rotary Switch

Use the rotary switch to set the VMEbus A24 base slave address. To set the address, use a screw driver to turn the inner wheel of the rotary switch until the arrow points to the value you want. The A24 base slave address you choose must be unique to the VMEbus system.

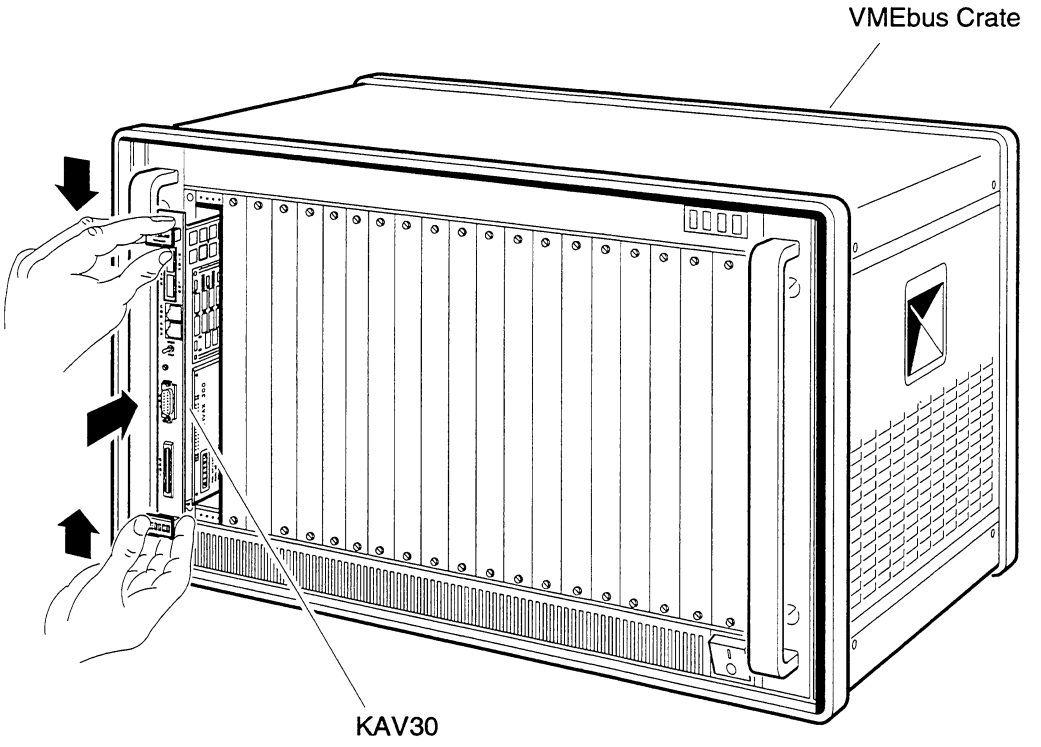
2.5 Inserting the KAV30 into a VMEbus Crate

This section describes how to insert the KAV30 into a VMEbus crate. Figure 2–3 shows how to insert the KAV30 into a VMEbus crate. To insert the KAV30 into a VMEbus crate, follow these steps:

1. Insert the KAV30 into the appropriate backplane slot. While holding the top and bottom extractor levers, lock the KAV30 in place by simultaneously

pushing down the top lever, pulling up the bottom lever, and pushing the KAV30 into the backplane slot (see Figure 2–3 for more information).

Figure 2–3 Inserting the KAV30 into a VMEbus Crate



2. Tighten the two front-panel screws that secure the KAV30 in the VMEbus crate.

Caution

If you do not tighten the two front panel screws, you cannot ensure that the KAV30 forms an electrical path with the crate.

2.6 Connecting Devices to the Serial Line Ports

This section describes how to connect devices to the KAV30 serial line ports. The devices that you connect to the KAV30 serial line ports must be product safety compliant with the local regulations, such as the UL1950, EN60950, CSA950 standards which include the SELV requirements.

Warning

If you connect devices that are not product safety compliant to the KAV30, you risk personal injury, fire, or damage to equipment.

You can connect DEC 423 compliant devices to the KAV30 serial line ports. For example, you can connect printers, terminals, modems, and computers to the serial line port.

To connect serial line devices to the KAV30, use a shielded DECconnect™ office cable with modified modular plug (MMP) connectors. Connect the ground cable on the DECconnect office cable to the front panel grounding screw and tighten the grounding screw.

Note

If you do not tighten the grounding screw, you lose essential mechanical support for the KAV30.

The devices that you connect to the console port have KAV30 console functionality.

Table 2–4 describes the default serial line port settings.

Table 2–4 Default Console and Auxiliary Port Settings

Setting	Value
Baud rate	9600
Number of data bits	8
Number of parity bits	0
Number of stop bits	1

To connect serial line devices to the KAV30, plug one end of the cable into the relevant port and the other end into the device that you want to connect.

2.7 Connecting Devices to the AUI Port

This section describes how to connect devices to the KAV30 AUI port. Connect devices to the AUI port only when the power switch on the VMEbus crate is in the off position.

Caution

If you connect devices to the AUI port when the power switch on the VMEbus crate is in the on position, you risk the following:

- Malfunction in the KAV30 and devices connected to the KAV30
- Damage to the KAV30 and devices connected to the KAV30

Connecting an Ethernet Network: You can connect the KAV30 to ThinWire™ or standard Ethernet networks. To connect to a ThinWire Ethernet network, you must have a DESTA transceiver. To connect to a standard Ethernet network, you must have a DELNI, H4000, or H4005 transceiver.

To connect an Ethernet network to the AUI port, use a straight-version Ethernet transceiver cable. Use a straight-version cable because an angled-version cable blocks the KAV30 SCSI port.

To connect a ThinWire Ethernet network to the AUI port, follow these steps:

1. Connect the DESTA transceiver to the ThinWire Ethernet network.
2. Connect the cable's male connector to the AUI port.
3. Connect the cable's female connector to a DESTA transceiver.

To connect a standard Ethernet network to the AUI port, follow these steps:

1. Connect a DELNI, H4000, or H4005 transceiver to the standard Ethernet network.
2. Connect the cable's male connector to the AUI port.
3. Connect the cable's female connector to the transceiver.

Connecting an Ethernet Loopback Connector: If you do not connect an Ethernet network to the AUI port, you must protect the AUI port from ESD. To protect the AUI port from ESD, connect an Ethernet loopback connector to the port.

Caution

If you do not protect the AUI port from ESD, you risk damaging the KAV30.

2.8 Connecting Devices to the SCSI Port

This section describes how to connect devices to the KAV30 SCSI port. The devices that you connect to the KAV30 SCSI port must be product safety compliant with the local regulations such as the UL1950, EN60950, and CSA950 standards that include the SELV requirements.

Warning

If you do not connect product safety compliant devices to the KAV30, you risk personal injury, fire, or damage to equipment.

Connect devices to the SCSI port only when the power switch on the VMEbus crate is in the off position.

Caution

If you connect devices to the SCSI port when the power switch on the VMEbus crate is in the on position, you risk the following:

- Malfunction in the KAV30 and devices connected to the KAV30
- Damage to the KAV30 and devices connected to the KAV30

Setting the KAV30 SCSI ID: Before you connect the KAV30 to an external SCSI bus, set the KAV30 SCSI ID. Choose a SCSI ID that is unique to the SCSI bus.

The KAV30 stores the SCSI ID in its battery backed-up RAM. The default SCSI ID is seven. You can use the KAV30 utility that begins at $2008'0050 to change the SCSI ID. However, when the battery backed-up RAM loses power, the SCSI ID is reset to seven. See Section 5.3 for information about the KAV30 utilities.

Connecting a SCSI Bus: To connect a SCSI bus to the SCSI port, use a Digital SCSI cable with a 68-pin female Honda connector at one end. Use the 68-pin Honda connector to connect to the SCSI port. See the *Small Computer Systems Interface Developer's Guide* for more information on connecting a SCSI bus to the KAV30. See Section D.3 for a description of the SCSI port.

Protecting the SCSI Port from ESD: If you do not connect a SCSI bus to the SCSI port, you must protect the SCSI port from ESD. To protect the SCSI port from ESD, fit the SCSI port ESD cover on the port.

Caution

If you do not protect the SCSI port from ESD, you risk damaging the KAV30.

2.9 Finishing the Installation

This section describes how to finish installing the KAV30. To finish installing the KAV30, follow these steps:

1. Insert conductive blank panels in the vacant backplane slots.

   This ensures that the VMEbus system meets the airflow requirements.

2. Set the power switches on the SCSI devices connected to the VMEbus system to the on position.

3. Set the power switches on the devices connected to the VMEbus system to the on position.

4. Set the power switch on the VMEbus crate to the on position.

5. Monitor the KAV30 diagnostic display.

When you supply power to the KAV30, the KAV30 executes its self-test. Executing the KAV30 self-test consists of executing all the KAV30 diagnostic tests. The self-test takes approximately 30 seconds to complete if the KAV30 has 4M bytes of system RAM, and approximately 2.5 minutes to complete if the KAV30 has 16M bytes of system RAM.

The KAV30 sends diagnostic test status information to the KAV30 front panel display and to the devices connected to the KAV30 console port. If the KAV30 passes all of the diagnostic tests, it displays 0 on its high status display and 5 on its low status display. See Section 3.1 for more information.

6. You can now use the VAXELN™ KAV Toolkit Extensions for VMS™ to configure the KAV30. See the *KAV30 Programmer's Reference Information* manual for more information.

Before you use a system containing the KAV30, you might have to satisfy the local regulations for such systems. Examples of actions that you must perform to satisfy the local regulations include the following:

- Obtain an electromagnetic interference (EMI) operation license for the system.
- Obtain a Postal, Telephone, and Telegraph (PTT) administration operation license, if you connect the system to public telephone networks.
- Supply appropriate instructions to the end user of the system. Instructions must include product safety information in the end user's local language.

3
Testing the KAV30

This chapter describes how to test the KAV30. It gives information on the following:

- Diagnostic tests (see Section 3.1)
- Diagnostic Halt on Fatal Error (see Section 3.2)
- Utilities (see Section 3.3)

3.1 Diagnostic Tests

This section describes the KAV30 diagnostic tests. The KAV30 includes a number of diagnostic tests that allow you to test the KAV30 hardware. See Appendix B for more information about the KAV30 diagnostic tests.

You can use any of the following methods to run the diagnostic tests:

- Supply power to the KAV30

  When you supply power to the KAV30, the KAV30 executes its self-test. Executing the KAV30 self-test consists of executing all the KAV30 diagnostic tests. The self-test takes approximately 30 seconds to complete if the KAV30 has 4M bytes of system RAM, and approximately 2.5 minutes to complete if the KAV30 has 16M bytes of system RAM.

- Reset the KAV30

  When you reset the KAV30, the KAV30 executes its self-test.

- Use the TEST console command

  You can use the TEST console command to execute a single diagnostic test. See Section 5.1.2 for more information about the TEST console command.

While the KAV30 executes a diagnostic test, it shows the relevant test number on the high status display and the relevant error number on the low status display. If the KAV30 finishes the self-test successfully, it displays a 0 on its high status display and a 5 on its low status display.

The KAV30 also sends status information to the device that is connected to the console port. The KAV30 diagnostic display is similar to the rest of Digital's VAX™ product line. The KAV30 displays the test number, a character indicating the status of the test, and two periods (..). The following characters indicate the status of the test:

- A period (.) indicates a test that is successful
- A question mark (?) indicates a test that fails
- An underscore (_) indicates a test that is disabled or that the item being tested is missing

Note

If a terminal is connected to the console port and the hold screen command is active on the terminal, then the self-test stops at diagnostic test number C. The self-test will not continue until you deactivate the hold screen command.

A typical KAV30 console diagnostic display for the self-test is as follows:

```
E...D...C...B...A...9...8...7...6...5...4...3...2...1...
```

The following table describes error messages that the KAV30 might display, and the actions that you must take if the KAV30 displays the messages.

Message	Reason for Message	User Action
D_	Diagnostic test D is disabled.	On initial power up (no battery connected or discharged during power down), the state of the memory flag (long or short memory test) is unpredictable. If you want to force short or extended memory test, you have to invoke the utility. See Section 5.3 and Appendix B for more information on disabling this diagnostic test.
3_	The auxiliary power supply is dead.	If you use the KAV30 battery as the source of the auxiliary power supply, see Table 3–1. If you use the VMEbus standby power supply, see the VMEbus system documentation. If you do not use either power supply, do not take any action.
2_	The KAV30 does not contain a SCSI controller.	If the KAV30 does not contain a SCSI controller, do not take any action. If the KAV30 contains a SCSI controller, contact your Digital representative.

The on-board self-test 3 tests the realtime clock chip and the on-board battery. The KAV30 may report problems when running this test. Table 3–1 contains information about the problems that may be detected by this test. The sub-test code is displayed on the lower front panel LED of the KAV30. The sub-test code is not included in the console printout.

Table 3–1 Diagnostic Test 3 Sub-Test Codes

Sub-Test Code	Meaning	User Action
8	Oscillator failed. RTC calendar clock and battery-backed-up area may be invalid.	Informational message only. No user action.
6	Battery is not connected, or battery is dead.	If the battery is connected, recharge the KAV30 for 30 minutes. If the problem persists contact your Digital representative.
5	Clock running too fast, or too slow, or not clocking.	Contact your Digital representative.

When you turn on the KAV30 in the VMEbus crate, the self-test fails, displaying a 3 on its high status display and an 8 on its low status display, in the following circumstances:

- When the J9 jumper posts are positioned so that the KAV30 battery is disconnected.
- When the KAV30 battery is not charged.
- If this is the first execution of the self-test since the J9 jumper posts were set to use the KAV30 battery as the auxiliary power source. In this case, turn off the VMEbus crate and then turn it on again to complete the self-test.

Note

To prolong battery life, the KAV30 is shipped with the battery disconnected.

If errors other than those listed in this section occur, contact your Digital representative.

3.2 Diagnostic Halt on Fatal Error

The current EPROM revision (see the *Hardware Release Notes, AV-PJRKA-TE*) allows the user to select whether the KAV30 halts after detecting a fatal error or continues to boot, if a boot source is selected, regardless of any fatal error. The diagnostic self-tests always run to completion (`test ..1`) before they halt or continue. You select the Halt on Fatal Error action using DIP switch 7.

If DIP switch 7 is set to the off position, which is the factory default setting, the following occurs:

- All the on-board self-tests run before the KAV30 halts. If a console terminal is connected to the KAV30 it indicates that the self-test halted at test 1.
- The KAV30 LEDs display the code for the first failing test.
- If a console terminal is connected to the KAV30, then a question mark (?) appears after each failing test. A dead or non-connected battery is displayed as an underscore character (_), which indicates that a device is missing. The diagnostics treat this as a fatal error and halt on test 1.
- When the on-board diagnostics detect a failing test they record details of the failing test in the battery-backed up RAM. The sample VAXELN Pascal program in Appendix C demonstrates how to retrieve information concerning the failing tests from the battery-backed up RAM.

If DIP switch 7 is set to the on position, the following occurs:

- The KAV30 runs through all on-board self-tests before attempting to boot, if a boot device is selected.
- If a console terminal is connected to the KAV30, then a question mark (?) appears after each failing test. A dead or non-connected battery is displayed as an underscore character (_), which indicates that a device is missing.
- When the on-board diagnostics detect a failing test they record details of the failing test in the battery-backed up RAM. The sample VAXELN Pascal program in Appendix C demonstrates how to retrieve information concerning the failing tests from the battery-backed up RAM.

Note

Any error reported by the diagnostics should be viewed as a failure of the KAV30. If the diagnostics detect errors and you continue to operate the KAV30, unpredictable application behavior may result.

3.2.1 Differences between KAV30 Revision Levels A and B

You can only use DIP switch 7 to select the Halt on Fatal Error action in KAV30 modules of the revision level A. This is due to firmware changes in the rtVAX300, which is incorporated in all KAV30 modules of revision level B and higher. KAV30 modules of revision level B and higher have the following functionality with DIP switch 7:

- If DIP switch 7 is in the off position, the functionality is as described in Section 3.2.
- If DIP switch 7 is in the on position, the KAV30 halts at the console prompt, after completing the self-test, whenever a fatal error occurs. In other words, there is no automatic boot possible after a fatal error.
- If a console terminal is connected to the KAV30 a question mark (?) is displayed after each failing test. A dead or non-connected battery is displayed as an underscore character (_), which indicates that a device is missing. The KAV30 treats this as a nonfatal error and continues to boot.

3.3 Utilities

This section describes the KAV30 utilities. The user ROM on the KAV30 that you receive contains eight utilities. You can use the utilities to perform a variety of functions, including testing and debugging the KAV30. The utilities allow you to test, without using the KAV30 software, how the KAV30 interacts with the other VMEbus modules. They also allow you to configure the VMEbus and VSB.

Before you use the utilities, you must understand all aspects of the following:

- The VMEbus
- The KAV30
- The devices with which the KAV30 communicates

See Section 5.3 for more information about the KAV30 utilities.

4

KAV30 Hardware

This chapter describes the KAV30 hardware. It gives information on the following:

- Overview of the KAV30 hardware (see Section 4.1)
- Status display (see Section 4.2)
- Reset/halt switch (see Section 4.3)

4.1 Overview of the KAV30 Hardware

This section gives an overview of the KAV30 hardware. Figure 4–1 shows a block diagram of the KAV30 hardware.

The KAV30 contains the following hardware:

- Battery

  The battery can back up the battery backed-up RAM and the calendar /clock. See Section 2.4 for information on configuring the KAV30 to allow the battery to back up the battery backed-up RAM and the calendar/clock.

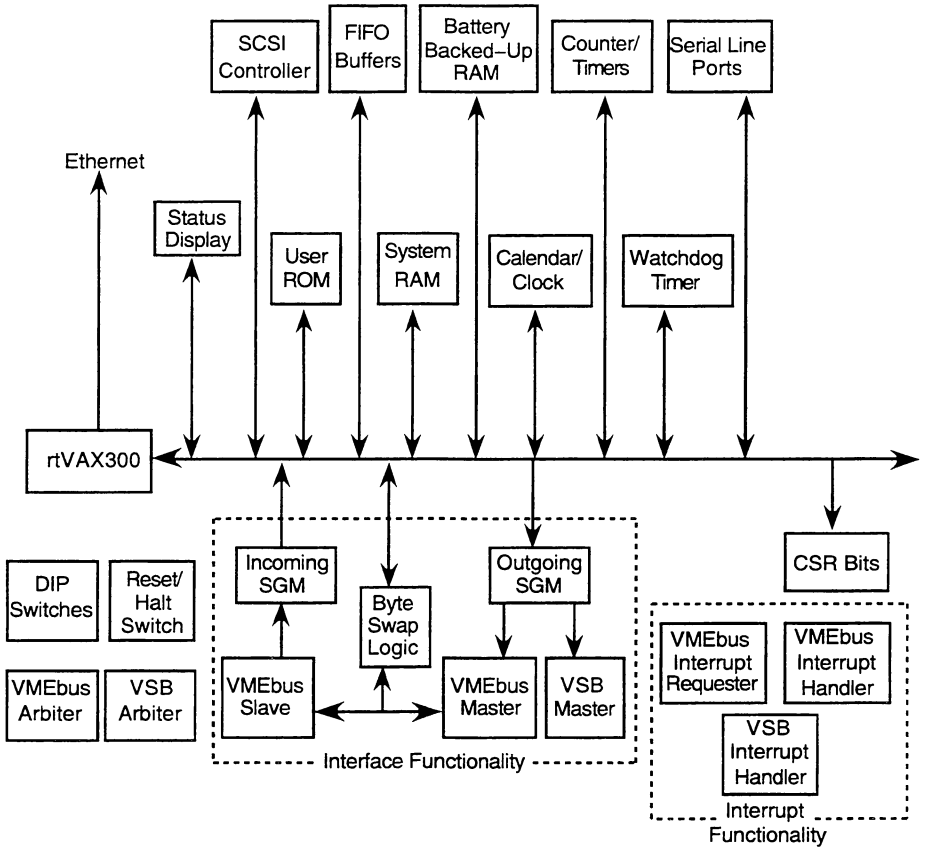
- Battery backed-up RAM

  There are 32K bytes of battery backed-up RAM. You can configure the KAV30 so that either the battery or the VMEbus standby power supply backs up the RAM. You can also configure the KAV30 so that the RAM has no backup power.

- Byte-swap logic circuitry

  The byte-swap logic circuitry allows the KAV30 (little-endian architecture) to exchange data with modules containing 68000® processors (big-endian architecture).

Figure 4–1 KAV30 Hardware Block Diagram



- Calendar/clock

  The calendar/clock keeps absolute time to a resolution of 10 milliseconds (ms). It can format this time into 12- or 24-hour mode, with the following counting functionality:

  - Day of week counting
  - Day of year counting
  - Automatic leap year counting

You can configure the KAV30 so that either the battery or the VMEbus standby power supply backs up the calendar/clock.

- Control and status register (CSR) bits

  The CSR bits consist of input bits and output bits. The CSR input bits report on the status of the KAV30 hardware, while the CSR output bits control the KAV30 hardware.

- Counter/timers

  There are fifteen 16-bit counter/timers. You can cascade two counter/timers to form one virtual 32-bit counter/timer. See the *KAV30 Programmer's Reference Information* manual for more information.

- First-in/first-out (FIFO) buffers

  The KAV30 uses hardware to implement four independently-operating FIFO buffers. You can access the FIFO buffers either in FIFO mode or in last-in/first-out (LIFO) mode.

  Each FIFO buffer has 255 32-bit entries. However, you can perform byte, word, longword, quadword, and octaword operations on these buffers. See the *KAV30 Programmer's Reference Information* manual for more information about the FIFO buffers.

- Front panel

  The KAV30 has a conductive aluminum grounded front-panel. The front side of the front panel is anodized. The reverse side of the front panel is chromatized.

- Hardware revision registers

  The hardware revision registers indicate the hardware revision of the KAV30.

- Incoming scatter-gather map (SGM) logic circuitry

  The incoming SGM allows other devices on the VMEbus to access the KAV30 system RAM, FIFO buffers, and VMEbus reset register. The KAV30 incoming SGM consists of 256 64K-byte pages.

  Each incoming SGM entry offers page write-protect functionality. Also, you can set each incoming SGM page to interrupt the CPU when it receives a read or write access.

- Outgoing SGM logic circuitry

  The outgoing SGM allows the KAV30 to access other devices on the VMEbus or VSB. The outgoing SGM consists of 3584 64K-byte pages of VMEbus or VSB address space.

Each outgoing SGM entry offers VMEbus address modifier, VSB address space, and page write-protect functionality.

- rtVAX™ 300 processor

  The rtVAX 300 processor is a 3-inch by 5-inch daughter card. It includes the following:

  - A CVAX™ microprocessor
  - A floating-point unit (FPU) coprocessor
  - A second-generation Ethernet controller (SGEC)
  - An interval timer
  - ROM
  - Control logic circuitry

  The CVAX microprocessor implements a central processing unit (CPU), which allows the KAV30 to run VAXELN applications. The CVAX microprocessor can store accesses in a 1 kilobyte, two-way associative, write-through memory cache. The KAV30 hardware maintains cache coherency for VMEbus, VSB, SCSI, and SGEC bus cycles.

  The KAV30 uses the FPU, the CVAX microprocessor, and the VAXELN emulated instruction trap handlers to execute VAX floating-point instructions.

  The SGEC allows the KAV30 to communicate with Ethernet networks. The SGEC can perform direct memory access (DMA) operations with the the KAV30 system RAM. However, the SGEC cannot move data from one system memory to another.

  The VAXELN kernel uses the interval timer to maintain the current time. The interval timer interrupts the CVAX microprocessor every 10 ms.

  The rtVAX 300 ROM contains firmware and diagnostic code. The rtVAX 300 firmware emulates a typical VAX console. The rtVAX 300 diagnostic code performs two functions:

  - It tests the rtVAX 300 hardware.
  - It checks the KAV30 user ROM for diagnostic code to test the remaining KAV30 hardware.

  The rtVAX 300 is the source of the data and address lines (DAL) bus, a 32-bit multiplexed bus. See the *rtVAX 300 User's Guide* for more information about the rtVAX 300 processor.

- SCSI controller

  The SCSI controller is optional. The KAV30 contains a SCSI controller only if you order a KAV30 that has one. You cannot install the SCSI controller as an upgrade.

  The SCSI controller can perform synchronous transfers at a rate of 2.5 megabytes per second and asynchronous transfers at a rate of 1.5 megabytes per second.

  The SCSI controller can perform DMA operations with the KAV30 system RAM. However, the SGEC cannot move data from one system memory to another.

  The SCSI controller contains an embedded Reduced Instruction Set Computer (RISC) processor. This RISC processor allows the SCSI controller to process lists of instructions, called scripts.

- Status display

  The status display shows the diagnostic and operation status information. See Section 4.2 for more information about the status display.

- System RAM

  Depending on the KAV30 that you order, it has either 4M bytes or 16M bytes of system RAM. The KAV30 uses a 4M byte dynamic random-access memory (DRAM) daughter card to implement 4M bytes of system RAM. The KAV30 uses a 16M byte DRAM daughter card to implement 16M bytes of system RAM.

  The KAV30 system RAM is parity-protected, with one parity bit for each byte of system RAM. The 4M byte DRAM daughter card consists of thirty-six 1-megabit chips. The 16M byte DRAM daughter card consists of thirty-six 4-megabit chips. The system RAM uses 32 chips to store data bits and four chips to store parity bits.

  The CVAX microprocessor, SGEC, SCSI controller, and VMEbus slave data interface logic circuitry can read from and write to the system RAM.

- Serial line ports

  The KAV30 has two DEC 423 compliant serial line ports: the *console port* and the *auxiliary port*. It uses a universal asynchronous receiver /transmitter (UART) to implement these ports. The default baud rate is 9600 baud. You can program the ports to have baud rates between 75 and 19 200 baud. However, when you supply power to the KAV30, the baud rate is reset to 9600.

- User ROM

  The KAV30 can have 128K bytes, 256K bytes, 512K bytes, or 1M byte of user ROM. The KAV30 uses two 16-bit EPROMs connected to 40-pin Joint Electron Device Engineering Council (JEDEC) sockets to implement the user ROM. You can upgrade or replace the user ROM. The user ROM you receive contains KAV30 firmware.

  You can program the user ROM with application programs. However, if you do, you must overwrite the SCSI boot user application program with the application programs. See Chapter 6 for more information.

- VMEbus arbiter functionality

  The KAV30 includes logic circuitry that implements VMEbus arbiter functionality. Use a DIP switch to enable the KAV30 VMEbus arbiter functionality. However, if you want the KAV30 to act as the VMEbus arbiter, you must place it in the leftmost backplane slot (slot 1) of the VMEbus system.

  The VMEbus arbiter can provide prioritized, round-robin, or single-level bus arbitration.

- VMEbus interrupt handler logic circuitry

  The KAV30 includes logic circuitry that can handle vectored and autovectored interrupt requests (IRQs) that it receives from the VMEbus. It can handle vectored interrupt requests on the VMEbus IRQ<6..1> lines. It can handle autovectored interrupt requests on the VMEbus IRQ<7..1> and POWER_FAIL lines.

- VMEbus interrupt requester logic circuitry

  The KAV30 includes logic circuitry that can generate vectored VMEbus IRQs. It can request interrupts on the VMEbus IRQ<7..1> lines.

- VMEbus master data interface logic circuitry

  The KAV30 includes VMEbus master data interface logic circuitry that supports the following:

  - A16, A24, and A32 addressing modes
  - D08, D16, and D32 data transfers
  - Read-modify-write cycles

- VMEbus slave data interface logic circuitry

  The KAV30 includes VMEbus slave interface logic circuitry that supports the following:

- VMEbus A24 and A32 addressing modes
- D08, D16, D32, and block mode data transfers
- Read-modify-write cycles

The VMEbus slave data interface can access the KAV30 system RAM, FIFO buffers, and the VMEbus reset register.

- VSB arbiter functionality

  The KAV30 includes logic circuitry that implements VSB arbiter functionality. Use the VAXELN KAV Toolkit Extensions for VMS to enable the KAV30 VSB arbiter functionality. However, if you want the KAV30 to act as the VSB arbiter, you must place it in the leftmost backplane slot (slot 0) of the VSB system.

- VSB interrupt handler logic circuitry

  The KAV30 includes logic circuitry that can handle autovectored IRQs that it receives on its VSB IRQ line.

- VSB master data interface logic circuitry

  The KAV30 includes VSB master interface logic circuitry that supports VSB ALTERNATE, SYSTEM, and I/O address spaces.

- Watchdog timer

  The KAV30 has a watchdog timer. If you enable the watchdog timer, it asserts the RESET signal when the KAV30 application enters an infinite loop. Application programs must periodically refresh the watchdog timer, otherwise the watchdog timer asserts the RESET signal. Use the VAXELN KAV Toolkit Extensions for VMS to enable, disable, or refresh the watchdog timer.

4.2 Status Display

This section describes the KAV30 status display. While the KAV30 is receiving power, it shows the diagnostic and operation status information on its status display. This status display shows a high-order digit above a low-order digit, with a decimal point on either side of each digit.

The KAV30 uses the digits on the status display to display its diagnostic status information. See Section 2.9 for more information on the KAV30 diagnostic status information.

The KAV30 uses the decimal points on the status display to indicate its operation status information. There are two decimal points on each display: one to the left of the digit and one to the right of the digit. The KAV30 lights the decimal points as follows:

- The KAV30 lights the decimal point to the left of the digit, on the low status display, if it is configured to act as the VMEbus arbiter.
- The KAV30 lights the decimal point to the right of the digit, on the low status display, if it is configured to act as the VSB arbiter.
- The KAV30 lights the decimal point to the left of the digit, on the high status display, if it connects to a SCSI bus on which there is a power supply to the TERMPWR signal. You can use the status of this decimal point to troubleshoot the SCSI bus. See Section 5.5 for more information.
- The KAV30 lights the decimal point to the right of the digit, on the high status display, if there is a VMEbus slave access to the KAV30. Because this pulse is not sufficient in length to allow the human eye to see the display, the KAV30 extends the pulse length to 0.5 seconds.

4.3 Reset/Halt Switch

This section describes the KAV30 reset/halt switch. The reset/halt switch is on the KAV30 front panel. You can use this switch to reset the KAV30, or halt the KAV30 CPU.

This switch remains in the center, unless you move it. While the switch is in the center, the KAV30 operates normally. However, you can move the switch to either of the following settings:

- Reset

  If you move the switch to this setting you reset the KAV30, and the KAV30 executes its diagnostic tests.

- Halt

  If you move the switch to this setting you halt the CPU, and the KAV30 starts its console program.

There is momentary contact between these switch settings. That is, the switch returns to the center when you release your finger.

5

Operating the KAV30

This chapter describes how to operate the KAV30. It gives information on the following:

- Console program (see Section 5.1)
- Booting the KAV30 (see Section 5.2)
- Using utilities (see Section 5.3)
- Testing the KAV30 auxiliary power supply (see Section 5.4)
- Troubleshooting the SCSI bus (see Section 5.5)

5.1 Console Program

This section describes the KAV30 console program. You can enter commands that control the operation of the KAV30 CPU via the console program. The console program runs on a terminal that is connected to the KAV30 console port. It is similar to the console programs supported by the rest of Digital's VAX product line. See Section 2.6 for more information on how to connect a terminal to the KAV30 console port.

5.1.1 Running the Console Program

To run the console program, set the KAV30 reset/halt switch to the halt position. When you run the console program, it displays the console prompt (>>>) on the terminal connected to the KAV30 console port. Enter console commands at the console prompt.

The KAV30 also runs the console program when one of the following conditions occurs:

- A program executing on the KAV30 CPU executes a HALT instruction.
- A KAV30 restart or boot operation fails, and the default recovery action is to halt the CPU.
- A program executing on the KAV30 CPU encounters a severe error.

5.1.2 Console Program Commands

This section describes the console program commands. The commands are the same as the rtVAX 300 console program commands. See the *rtVAX 300 Hardware User's Guide* for more information. The console commands are as follows:

- `B[OOT] [/<DATUM>] [<device-name>[:]]`

  This command loads and starts an application program on the KAV30.

  The `<DATUM>` qualifier is optional. It specifies a hexadecimal value representing the boot flags. The console program passes the `<DATUM>` information as a longword to the bootstrap program. If, for example, the `<DATUM>` qualifier has a value of 4, the VAXELN kernel debugger, if it is present on the VAXELN system, is invoked with the application program.

  The `<device-name>` parameter is optional. This parameter specifies the name of the device containing the application program. Table 5–1 describes the device names that you can use. You can type a colon (:) at the end of the device name, but it is not required.

Table 5–1 Boot Source Device Names

Device Name	Boot Source
EZA0	Ethernet
CSB0	Device† connected to the auxiliary port that operates at 1200 bits/s
CSB1	Device† connected to the auxiliary port that operates at 2400 bits/s
CSB2	Device† connected to the auxiliary port that operates at 9600 bits/s
PRB0	User ROM
PRB1	User ROM‡ (SCSI boot)

†The KAV30 performs a DECnet DDCMP boot.
‡The KAV30 first copies the contents of the user ROM to the system RAM.

If you do not specify the boot flag information or the device name, the console program uses default values. The default values depend on the KAV30 configuration.

- `B[OOT] [/<DATUM>] PRB1[:]`

  This command invokes the SCSI bootstrap program. The SCSI bootstrap program loads an application program from a SCSI device to the KAV30 system RAM and starts the application.

The SCSI bootstrap program searches each SCSI device for the [SYS0.SYSEXE]SYSBOOT.EXE file and loads this file (the user application) into the KAV30 system RAM, before starting the application. The SCSI bootstrap program first searches the device with the lowest SCSI ID, then the device with the next-lowest SCSI ID, and so on until it searches the device with the highest SCSI ID. If the SCSI bootstrap program encounters errors while searching a device, it ends the search of that device and continues the search with the next device.

The <DATUM> qualifier is optional. It specifies a hexadecimal value representing the boot flags. The console program passes the <DATUM> value as a longword to the SCSI bootstrap program. The following table describes the possible <DATUM> values:

Value	Action
00000002	Causes the bootstrap program to halt if it encounters errors, other than *Fatal drive error* and *Device not responding* errors, while searching a device.
00000010	The SCSI bootstrap program searches SCSI devices for a file called [SYSMAINT]DIAGBOOT.EXE and loads this file (the application program) into the KAV30 system RAM before starting the application.
00000100	The SCSI bootstrap program prompts you for the name of a file containing the application program.
n0000000	The SCSI bootstrap program searches the SCSI devices for a file called [SYSn.SYSEXE]SYSBOOT.EXE and loads the file (the application program) into the KAV30 system RAM before executing it.

You can combine boot flags. For example, if you set the <DATUM> qualifier to 00000102, the SCSI bootstrap program prompts you for the name of a file. The bootstrap program halts if it encounters errors.

The PRB1 parameter indicates that you want to invoke the SCSI bootstrap program. The SCSI bootstrap program is in the KAV30 user ROM. If you replace the SCSI bootstrap program with an application program, this command invokes the application program. You can type a colon (:) at the end of the device name, but it is not required.

- D[EPOSIT][/<QUALIFIER>] <ADDRESS> <DATUM>

  This command writes the specified data into a specified address. You can use the following optional qualifiers with this command:

Qualifier	Explanation
/B	A byte access
/W	A word access
/L	A longword access
/V	A virtual memory access
/P	A physical memory access
/I	An internal register access
/G	A general-purpose register access
/N:<COUNT>	Repeat the command <COUNT> times, where <COUNT> is an integer
/U	Access protected addresses

Note the following points about the qualifiers:

- If you do not specify qualifiers with a DEPOSIT command, the console program uses the previous set of qualifiers that you specified with this command.
- If you specify conflicting qualifiers, the console program does not perform the command and displays an error message.
- The /U and /N qualifiers affect only the command in which you type them.
- The /U qualifier allows you to access almost all the addresses. If you do not specify the /U qualifier, a protected DEPOSIT or EXAMINE command can access memory only in the PFN map, or physical addresses between 20000000 and 3FFFFFFF.

The <ADDRESS> parameter is a hexadecimal value representing a memory address. The DEPOSIT command supports the following symbolic addresses:

Symbol	Explanation
*	The address last referenced by an examine or deposit command.

Symbol	Explanation
@	The address last referenced by an examine or deposit command, with the exception of general register addresses.
+	The address immediately following the address last referenced by an examine or deposit command. For references to physical or virtual memory spaces, the address is the final address in the memory space plus the size of the last reference (1 for byte, 2 for word, 4 for longword).
-	The address immediately preceding the address last referenced by an examine or deposit command. For references to physical or virtual memory spaces, the address is the final address in the memory space minus the size of the last reference (1 for byte, 2 for word, 4 for longword).

The default <ADDRESS> value is equal to the + symbolic address.

The <DATUM> parameter is a hexadecimal value representing data that you want to place in the memory location. The default <DATUM> value is zero.

- E[XAMINE] [/<QUALIFIER>] [<ADDRESS>]

  This command displays the contents of the specified address in hexadecimal format.

  The supported qualifiers and the address specification are identical to those of the DEPOSIT command.

- HE[LP]

  This command displays all the console commands, supported parameters, and available options.

- I[NITIALIZE]

  This command initializes the CPU.

- R[EPEAT] <COMMAND>

  This command repeatedly executes the specified command. To stop executing the command, press Ctrl/C.

  The <COMMAND> parameter can be any valid KAV30 console command, with the exception of REPEAT.

- S[TART] <ADDRESS>

  This command executes instructions at a specified address. The address is within the context of the user's memory management mode (physical or virtual).

To execute this command, the Interrupt Stack Pointer (ISP) must contain a valid virtual or physical address. The console program pushes two longwords onto the interrupt stack. If the interrupt stack contains an invalid address, the console program displays the following message:

```
?04 ISP ERR
```

Note

The ISP is undefined after you switch on the power or reset the KAV30.

- `T[EST] <test-name>`

  This command executes a specified diagnostic test. The `<test-name>` parameter specifies a diagnostic test name. Before you execute this command, execute the UNJAM command to reset the KAV30. See Appendix B for information on the KAV30 diagnostic test names. See Section 3.1 for information on how to monitor the status of the KAV30 diagnostic tests.

  As with the self-test, in most cases the console program stops executing a diagnostic test and displays error information if it detects a serious error. If diagnostic tests complete successfully, the console program displays the console prompt. If diagnostic tests are disabled or the console program detects that the hardware function that you want to test is missing, the console program displays the following message:

  ```
  ?40 NOSUCHDEV
  ```

- `U[NJAM]`

  This command resets the system. It resets all the registers to zero, and resets all the logic circuitry to state zero.

- `<COMMAND> ! <COMMENT>`

  The exclamation point (!) prefixes a comment. If the console program encounters an exclamation point, it ignores the rest of the line.

5.1.3 Console Program Messages

The console program displays informational and error messages. Informational messages do not have message numbers. They consist only of text. However, error messages display a question mark (?), followed by a 2-digit hexadecimal number, followed by an abbreviated text message. The 2-digit hexadecimal number is the message number. See the *rtVAX 300 Hardware User's Guide* for an explanation of the console program error messages.

5.2 Booting the KAV30

This section describes how to boot the KAV30. To boot the KAV30, use one of the following methods:

- Execute a BOOT console command

  See Section 5.1.2 for information about the BOOT console command.

- Execute a programmed boot command

  Use software to enter a programmed boot command. The command you use depends on the programming language you use. See the *VAXELN Development Utilities Guide* for more information.

- Perform a remote trigger

  Perform remote triggers from a host VAX computer that is running the VMS operating system and is connected to the KAV30 via Ethernet. To perform a remote trigger, you must have the following:

  - A KAV30 that is configured to allow remote triggering. See Section 2.4 for information on configuring the KAV30.
  - A VAX host computer with a network database that is set up to access the KAV30.

    To set up the host computer network database to access the KAV30, use the network control program (NCP) on the host computer. Use the NCP program to define the KAV30 node name, address, hardware address, and load file. For example, to set the hardware address, node name, and load file for QNA-0, enter the following commands at the NCP prompt:

```
NCP> SET NODE KAV30 HARD ADDRESS 08-00-2B-16-39-45
NCP> SET NODE KAV30 SERVICE CIRCUIT QNA-0
NCP> SET NODE KAV30 LOAD FILE USER:[DAY]TEST.EXE
```

    However, before you boot the KAV30, ensure that the network services are enabled on the host computer. For example, to enable the network services on QNA-0, enter the following commands at the NCP prompt:

```
NCP> SET CIRCUIT QNA-0 STATE OFF
NCP> SET CIRCUIT QNA-0 SERVICE ENABLE
NCP> SET CIRCUIT QNA-0 STATE ON
```

    To perform a remote trigger, enter the TRIGGER NCP command on the host computer. See the NCP help utility for more information about the NCP commands.

- An Ethernet link between the KAV30 and the VAX host computer. See Section 2.7 for information on connecting the KAV30 to an Ethernet network.

See the VMS documentation for more information on the VMS commands listed in this section.

- Supply power to, or reset, the KAV30

  When you supply power to, or reset, the KAV30, it uses its default configuration to determine the boot options and automatically boots.

5.3 Using Utilities

This section describes how to use the KAV30 utilities. The user ROM on the KAV30 that you receive contains eight utilities. You can use the utilities to perform a variety of functions, including testing and debugging the KAV30.

To execute a utility, enter the KAV30 START console command and specify the start address of the utility. However, before you execute utilities, enter the following console commands to initialize the rtVAX 300:

```
>>> UNJAM
>>> INIT
```

Table 5–2 describes the KAV30 utilities. It gives the start address of each utility and describes the function that the utility performs.

Table 5–2 KAV30 Utilities

Start Address	Function
$2008'0040	Continuously runs the KAV30 self-test. Digital recommends that you execute the UNJAM console command before executing this utility. If the first running of the self-test does not complete, you must reset or switch off the power on the KAV30 and execute the utility again.
$2008'0044	Allows you to enable or disable the running of the comprehensive system memory diagnostic test (test number D) during the self-test. See Appendix B for more information about diagnostic test number D.
$2008'0048	Allows you to set the calendar/clock's date and time.
$2008'004C	Continuously shows the calendar/clock's date and time.
$2008'0050	Displays information about the SCSI targets on the SCSI bus, and allows you to set the KAV30 SCSI ID.

(continued on next page)

Table 5–2 (Cont.) KAV30 Utilities

Start Address	Function
$2008'0054	Tests the VMEbus and VSB.
$2008'0058	Displays information that helps you to use the utility that starts at address $2008'0054.
$2008'005C	Displays this table.

5.4 Testing the KAV30 Auxiliary Power Supply

This section describes how to test the KAV30 auxiliary power supply. The auxiliary power supply is the source of power for the KAV30 battery backed-up RAM and the KAV30 calendar/clock. To test the KAV30 auxiliary power supply, perform one of the following actions:

- Execute diagnostic test number 3.

  See Section 5.1.2 for information about executing KAV30 diagnostic tests. If the >>> prompt is not displayed, and the KAV30 displays a 3 on its high status display and an 8 on its low status display, the auxiliary power supply has failed. This is expected behavior when the KAV30 battery is the auxiliary power supply and the battery has been connected for the first time (see Section 3.1 for more information).

- Call the appropriate KAV30 system service. See the *KAV30 Programmer's Reference Information* manual for more information.

If the auxiliary power supply fails, the contents of the calendar/clock and battery backed-up RAM are unpredictable. The auxiliary power supply can fail under the following conditions:

- The battery is the auxiliary power supply and the battery is dead.
- The VMEbus standby power supply is the auxiliary power supply, and it has failed.
- Neither the battery nor the VMEbus standby power supply is the auxiliary power supply.

5.5 Troubleshooting the SCSI Bus

This section helps you to troubleshoot a SCSI bus. If you order a KAV30 that has a SCSI controller, the KAV30 contains a SCSI port. You can connect a SCSI bus to the SCSI port. The SCSI bus must have at least one source of power to its TERMPWR signal. Any device on the SCSI bus can supply power to this signal. You can use the KAV30 status display to troubleshoot the power supply to the TERMPWR signal.

You can troubleshoot the power supply to the TERMPWR signal only when the VMEbus crate that houses the KAV30 has its power switch in the on position. When the power switch is in the on position, the KAV30 uses the decimal point to the left of the digit on the high status display to indicate status information about the TERMPWR signal. Table 5–3 describes how you can use the status of the decimal point to troubleshoot the SCSI bus.

Table 5–3 Troubleshooting the SCSI Bus

Other Devices on the SCSI Bus	Jumper set to Supply TERMPWR	Status of Decimal Point	Explanation
No	No	Not lighting	The high status display is functioning correctly.
No	No	Lighting	The high status display is not functioning correctly.
No	Yes	Not lighting	The SCSI termination power fuse has blown.
No	Yes	Lighting	The SCSI termination power fuse is intact.
Yes	No	Not lighting	No other device on the SCSI bus supplies power to the TERMPWR signal.
Yes	No	Lighting	At least one other device on the SCSI bus supplies power to the TERMPWR signal.
Yes	Yes	Not lighting	The SCSI bus termination power fuse has blown and no device, other than the KAV30, supplies power to the TERMPWR signal.

(continued on next page)

Table 5–3 (Cont.) Troubleshooting the SCSI Bus

Other Devices on the SCSI Bus	Jumper set to Supply TERMPWR	Status of Decimal Point	Explanation
Yes	Yes	Lighting	At least one device on the SCSI bus supplies power to the TERMPWR signal.

6

Replacing the KAV30 User ROM

This section describes how to replace the KAV30 user ROM. It gives information on the following:

- Overview of the KAV30 user ROM (see Section 6.1)
- Removing the KAV30 from a VMEbus crate (see Section 6.2)
- Replacing the EPROMs (see Section 6.3)
- Programming the EPROMs (see Section 6.4)

6.1 Overview of the KAV30 User ROM

This section gives an overview of the KAV30 user ROM. The KAV30 that you receive contains 256K bytes of user ROM. The KAV30 uses two 16-bit wide EPROMs to implement the user ROM. Each EPROM is 128K bytes in size.

The first 96K bytes of the user ROM are reserved for Digital. These bytes contain the code for the KAV30 diagnostic tests and utilities. The user area of the user ROM starts at 96K bytes. The user area on the KAV30 that Digital ships contains SCSI boot code. You can replace this SCSI boot code with a user application. However, the user application code must begin at the start of the user area, and thus overwrites the SCSI boot code.

If you want to store a user application in the user area of the user ROM, you might need to replace the KAV30 EPROMs with larger capacity EPROMs. Table 6–1 describes the possible user ROM sizes. The user area on the KAV30 that Digital ships is 160K bytes (256K bytes minus 96K bytes) in size. However, the KAV30 accommodates EPROMs that provide a capacity of up to 1M byte of user ROM. Therefore, 904K bytes (1M byte minus 96K bytes) is the size of the largest user area, and also the largest user application, that you can have.

If you want to replace the KAV30 EPROMs, the new EPROMs must satisfy the following conditions:

- The access time must be 120 nanoseconds (ns) or less
- The disable time must be 50 ns or less
- The enable time must be 50 ns or less

Table 6–1 User ROM Sizes

User ROM	EPROMs	EPROM Part Number
128K bytes	Two 32K by 16-bit EPROMs	27C0512
256K bytes	Two 64K by 16-bit EPROMs	27C1024
512K bytes	Two 128K by 16-bit EPROMs	27C2048
1M byte	Two 256K by 16-bit EPROMs	27C4096

Note

The part numbers listed in Table 6–1 refer only to EPROMs supplied by Digital. If you use similar EPROMs made by other manufacturers, the part numbers are different. The KAV30 operates correctly, however, if the EPROMs are of the correct type and size.

On the KAV30, two JEDEC sockets, an even socket and an odd socket, allow you to connect the EPROMs. The EPROM you connect to the even socket supplies the low-order 16 data bits of each 32-bit longword that you store in the user ROM. The EPROM you connect to the odd socket supplies the high-order 16 data bits of each 32-bit longword that you store in the user ROM.

Note

Different revisions of the KAV30 module are shipped with different EPROMs. See the *KAV30 Hardware Release Notes* for information about the EPROMs that are shipped with each revision of the KAV30.

6.2 Removing the KAV30 from a VMEbus Crate

This section describes how to remove the KAV30 from a VMEbus crate. To remove the KAV30 from a VMEbus crate, follow these steps:

1. Back up all the software and data on the VMEbus system.

2. Set the power switches of the SCSI devices connected to the VMEbus system to the off position.
3. Set the power switches of the devices connected to the system to the off position.
4. Set the power switch on the VMEbus crate to the off position.
5. To protect the KAV30 from ESD, wear an antistatic wrist strap while handling the KAV30. Place the KAV30 on an antistatic mat.

Caution

If you do not wear an antistatic wrist strap while handling the KAV30, or if you do not place the KAV30 on an antistatic mat, you risk damaging the KAV30.

6. Unplug the connections to the KAV30.
7. Unscrew the two front panel screws that secure the KAV30 in the VMEbus crate.
8. Unlock the KAV30 by simultaneously pulling up the top extractor-lever, pushing down the bottom extractor-lever, and pulling the KAV30 out of the VMEbus crate.
9. Pull out and remove the KAV30 from the VMEbus crate.
10. If the battery is the auxiliary power supply, and you do not want to back up the battery backed-up RAM and calendar/clock, configure the battery jumper to prevent battery wastage. See Section 2.4.1 for more information.
11. Place the KAV30 in an antistatic bag.

Caution

If you do not place the KAV30 in an antistatic bag, you risk damaging the KAV30.

If you want to use the system from which you removed the KAV30, follow these steps:

1. Reconfigure the system's VMEbus and VSB backplanes. See the system documentation and the VSB documentation for more information on configuring the backplanes.

2. Insert a conductive blank panel in the backplane slot that you removed the KAV30 from.
3. Set the power switches on the SCSI devices connected to the system to the on position.
4. Set the power switches on the devices connected to the system to the on position.
5. Set the power switch on the crate to the on position.

6.3 Replacing the EPROMs

This section describes how to replace the KAV30 EPROMs. Figure 6–1 shows how to replace the KAV30 EPROMs.

Note

Some versions of the KAV30 are shipped with a piggyback socket on the EPROM sockets. The part number for the piggyback socket is FD-23361-01.

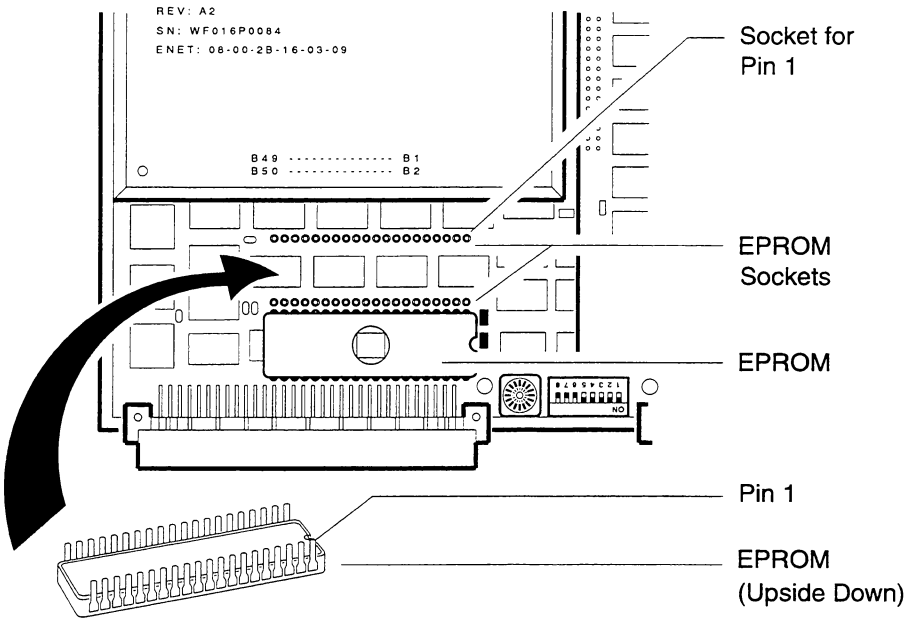
Caution

Never remove the piggyback sockets from the EPROM sockets. Instead, remove the EPROMs from the piggyback sockets.

To replace the KAV30 EPROMs, follow these steps:

1. Remove the KAV30 from the VMEbus crate. See steps 1 to 8 in Section 6.2 for more information.
2. Remove both EPROMs from the KAV30. Make a note of the odd and even EPROMs. Place the EPROMs to one side, you will need them later.
3. Program the new EPROMs with the KAV30 firmware and the application programs you want to include. See Section 6.4 for more information.
4. Insert the new EPROMs. See Figure 6–1 for more information. Make sure to insert the EPROMs so that pin 1 of the EPROM is in the socket for pin 1.

Figure 6–1 Replacing the KAV30 EPROMs



Caution

If you do not insert pin 1 of the EPROM into the socket for pin 1, you risk damaging the EPROM.

5. Insert the KAV30 into the backplane slot that you removed it from in step 1.
6. Turn on the VMEbus system. See steps 2 to 4 in Section 2.9 for more information.

6.4 Programming the EPROMs

This section describes how to program the EPROMs with code for the KAV30 diagnostic tests, the KAV30 utilities, and a user application. To program the EPROMs with code for the KAV30 diagnostic tests, the KAV30 utilities, and a user application, follow these steps:

Note

If you program the EPROMs with user application code, you overwrite the KAV30 SCSI boot code.

1. Convert the application programs into VAXELN images using the VAXELN System Builder with the destination set to ROM. See the *VAXELN System Development Guide* for more information about the VAXELN System Builder.
2. Convert the VAXELN image, created in step 1, into two files.

   One file contains the low-order 16-bits of each longword, and the other file contains the high-order 16-bits of each longword. This conversion is necessary because of the physical arrangement of the user ROM: the even EPROM supplies the low-order 16-bits, and the odd EPROM supplies the high-order 16-bits of each longword read from the user ROM.
3. Read the contents of the even EPROM that you removed from the KAV30 into the ROM burner memory.

   The even EPROM contains the low-order 16-bits of the longwords that make up the KAV30 firmware.

Caution

If you do not include the code for the KAV30 diagnostic tests and utilities in the new EPROMs, you risk causing a malfunction in the KAV30.

4. Append the file containing the low-order 16-bits of each longword in the application programs (see step 2) to the contents of the ROM burner memory, starting at 96K bytes.
5. Program the new even EPROM with the contents of the ROM burner memory.
6. Repeat steps 3, 4, and 5 for the odd EPROM.

The EPROMs now contain code for the KAV30 diagnostic tests, the KAV30 utilities, and a user application.

A

KAV30 Hardware Specifications

This appendix gives the KAV30 hardware specifications. It gives information on the following:

- Temperature and humidity specifications (see Section A.1)
- Power supply specifications (see Section A.2)
- Bus load specifications (see Section A.3)
- Timing specifications (see Section A.4)

A.1 Temperature and Humidity Specifications

This section gives the KAV30 hardware temperature and humidity specifications. Table A–1 lists the KAV30 hardware temperature and humidity specifications.

Table A–1 Temperature and Humidity Specifications

Parameter	Specification
Operating conditions:	
Temperature range	5°C to 50°C
Temperature rate of change	20°C/h
Relative humidity	10% to 95% RH (noncondensing)
Maximum relative humidity rate of change	10% RH/h
Maximum wet bulb temperature	32°C
Minimum dew point temperature	2°C
Heat dissipation	45 W (maximum)

(continued on next page)

Table A–1 (Cont.) Temperature and Humidity Specifications

Parameter	Specification
Nonoperating conditions:	
Temperature range	-40°C to 66°C
Relative humidity	95% RH (maximum)
Storage conditions:	
Temperature range	5°C to 50°C
Relative humidity	10% to 95% RH

Note

Digital strongly recommends that you operate the module in crates or environments with a forced cooling of at least 100 lfm. If you do not follow this advice, the MTBF of your module may degrade significantly.

A.2 Power Supply Specifications

This section gives the KAV30 power supply specifications. Table A–2 indicates the KAV30 maximum power supply specifications.

Table A–2 Maximum Power Supply Specifications

Supply	5°C	20°C	50°C
+5 V	9.5 A	9 A	7.9 A
+12 V	200 mA	200 mA	200 mA
-12 V	200 mA	200 mA	200 mA

These power specifications do not include the +12 V power requirement of devices that you connect to the AUI port. However, this current is limited by a fuse and by the capacity of the backplane connector. These power specifications include the +5 V power requirement of the SCSI terminators.

The typical power consumption for a KAV30 is 6.1A at 25°C for the +5 V power supply. This does not include the +5 V power requirement of the SCSI terminators.

A.3 Bus Load Specifications

This section gives the KAV30 bus load specifications. Table A–3 lists the KAV30 bus load specifications.

Table A–3 KAV30 Bus Load Specifications

Signal Name	Capacitive Load (pF)	I_{IL} (μA)	I_{OZL} (μA)	I_{IH} (μA)	I_{OZH} (μA)
AS	19	-1	1	-50	50
ACFAIL	5	-1	1	–	–
AM0 - AM5	18	-100	10	-50	50
A01 - A31	20	-1	1	50	50
BBSY	20	-100	20	50	50
BCLR	10	–	–	-50	50
BERR	20	-1	1	-50	50
BG0IN - BG3IN	18	-100	10	60	60
BG0OUT - BG3OUT	10	–	–	40	40
BR0 - BR3	21	-200	20	40	40
DS0 - DS1	17	-1	1	-50	50
D00 - D31	24	45	45	45	45
DTACK	20	-1	1	-5	5
IACK	30	120	30	-50	50
IACKIN	16	–	–	–	–
IACKOUT	16	–	–	–	–
IRQ1 - IRQ7	15	-20	20	-100	100
LWORD	17	-1	1	-50	50
SERCLK	–	–	–	–	–
SERDAT	–	–	–	–	–
SYSCLK	10	–	–	-50	50
SYSFAIL	30	-10	10	-10	10
SYSRESET	10	-1	1	-50	50
WRITE	17	-1	1	-50	50

A.4 Timing Specifications

This section gives the KAV30 timing specifications. Table A–4 lists the KAV30 system RAM access timing-specifications.

Table A–4 System RAM Access Timing-Specifications

Type of Access	Time (ns)
rtVAX 300	
Byte access	200
Word-aligned word access	200
Word access with bits <1:0> set to 11	300
Aligned longword access	200
Nonaligned longword access	300
Aligned quadword access	300
Nonaligned quadword access	400
Aligned octaword access	500
Nonaligned octaword access	600
SCSI Controller†	
Aligned longword access	400
VMEbus Slave Logic Circuitry†	
Aligned longword access	500
Block access	500

†The access times do not include the delay involved in obtaining control of the DAL bus. Typically, this delay takes between 100 ns and 1000 ns. However, this delay depends on what instruction the CVAX is currently executing, and can be much longer.

B

Diagnostic Tests

This appendix describes the KAV30 diagnostic tests. Table B–1 lists the KAV30 diagnostic tests. Supply the test name when you execute a test. The KAV30 displays the test number on its high status display and on the console terminal, when it executes a test. The KAV30 displays a test's error number on its low status display, when it tests that error condition.

Table B–1 KAV30 Diagnostic Tests

Test Name	Test Number	Error Number	Description
0	F	-	Tests that the rtVAX 300 power supply is correct and that the rtVAX 300 can write to the rtVAX 300 display register.
1	E	-	Tests that the rtVAX 300 ROM checksum is correct, that the high and low bytes of the rtVAX 300 ROM are the same version, and that the rtVAX 300 ROM test patterns are correct. It also tests the functionality of the rtVAX 300 LED registers.
2	D	-	Initializes and verifies the rtVAX 300 scratch memory. The rtVAX 300 scratch memory is random-access memory (RAM). The rtVAX 300 ROM code uses the rtVAX 300 scratch memory. The test uses a bit-pattern test, an address test, and a cleared to zero test. You must have 16K bytes of valid memory to pass this test.
3	C	-	Checks that the rtVAX 300 console responds to the console address, and tests the functionality of the console serial line unit (SLU).
4	B	-	Not implemented.
5	A	-	Verifies the operation of the rtVAX 300 FPU.

(continued on next page)

Table B–1 (Cont.) KAV30 Diagnostic Tests

Test Name	Test Number	Error Number	Description
6	9	-	Verifies that the rtVAX 300 interval timer interrupt signal exists, that the interval timer generates interrupts when enabled, and that it is accurate.
7	8	-	Tests the functionality of the rtVAX 300 Ethernet interface. Tests that the Ethernet network ID ROM contains a valid network identification, and that the Ethernet network checksum is correct.
8	7	-	Verifies the system RAM, SGM RAM, and the FIFO buffer RAM. It also tests the functionality of the FIFO buffer RAM write interrupt.
9	6	-	Tests the functionality of the four FIFO buffers, that the FIFO buffer entries are processed in the correct order, that the read operations from an empty FIFO buffer result in an error, and that the write operations to a full FIFO buffer result in an error.
A	5	-	Tests that the KAV30 CPU can correctly access the logic circuitry that implements the CSR bits.
B	4	-	Tests that the logic circuitry that implements the CSR bits can interrupt the KAV30 CPU.
C	3	-	Tests the auxiliary power supply and that the KAV30 CPU can correctly access the calendar/clock. Verifies the battery backed-up RAM and tests that the KAV30 CPU can correctly access the VMEbus interrupt logic circuitry.
D	2	-	Tests that the KAV30 CPU can correctly access the SCSI controller, and that the SCSI controller can correctly interrupt the KAV30 CPU.
E	1	-	Reserved for future use by Digital.
F	0	5	Indicates that the system is waiting for or executing a console command.
F	0	4	Indicates that the system is trying to boot.
F	0	3	Indicates that the boot host is found, or that a ROM footprint is located.
F	0	2	Indicates that the ROM or SLU is not present or is invalid.

(continued on next page)

Table B–1 (Cont.) KAV30 Diagnostic Tests

Test Name	Test Number	Error Number	Description
F	0	0	Indicates that control is passed to an Ethernet boot or to an external ROM code.

C

VAXELN Pascal Program

This appendix contains a sample Pascal Program that can be used to read diagnostic error information stored in the battery backed-up RAM on board the KAV30 module.

```
MODULE read_diag_error_info;
{*****************************************************************************}
{*                                                                           *}
{*              COPYRIGHT (C) 1992 DIGITAL EQUIPMENT CORPORATION             *}
{*                                                                           *}
{*   THIS SOFTWARE MAY BE COPIED WITHOUT FEE PROVIDED THAT THE  COPIES  ARE  *}
{*   NOT  MADE OR DISTRIBUTED FOR DIRECT COMMERCIAL ADVANTAGE,  THAT CREDIT  *}
{*   TO THE SOURCE IS GIVEN AND THAT THIS ENTIRE COPYRIGHT NOTICE IS INCLUDED.*}
{*                                                                           *}
{*   THE  INFORMATION  IN  THIS  SOFTWARE  SHOULD  NOT  BE  CONSTRUED  AS A  *}
{*   COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.                            *}
{*                                                                           *}
{*   DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE,    *}
{*   SUPPORT, OR RELIABILITY OF THIS SOFTWARE.  THIS SOFTWARE IS             *}
{*   DISTRIBUTED "AS IS."                                                    *}
{*                                                                           *}
{*****************************************************************************}

{ This Program can be used to read the diagnostic error information stored in
  non-volatile RAM. }

INCLUDE $get_message_text;

   PROCEDURE handle_error( status : INTEGER);

   VAR
      error_text : VARYING_STRING(255);

   BEGIN

      ELN$GET_STATUS_TEXT ( status,
                            [STATUS$IDENT,STATUS$FACILITY,STATUS$TEXT,
                            STATUS$SEVERITY],
                            error_text);

      WRITELN(error_text);
```

```
   END;

  PROGRAM read_diag_error_info( INPUT, OUTPUT);

CONST
   diag_error_page = %X20110600;
   mem_offset      = %X0FC;
   data_offset     = %X108;

TYPE

   byte = [BYTE] 0..255;
   error_info = PACKED RECORD
                   upper_led : [POS(0)] byte;
                   filler_1  : [POS(8)] ARRAY [1..3] OF byte;
                   lower_led : [POS(32)] byte;
                   filler_2  : [POS(40)] ARRAY [1..3] OF byte;
                END;
   count_info = PACKED RECORD
                   count : [POS(0)] byte;
                   filler : [POS(8)] ARRAY [1..3] OF BYTE;
                END;
   error_info_array = ARRAY [1..90] OF error_info;

VAR
   memory_ptr     : ^count_info;
   data_ptr       : ^error_info_array;
   page_ptr       : ^ANYTYPE;
   page_ptr_value : INTEGER;
   error_count    : INTEGER;
   i              : INTEGER;
   status         : INTEGER;

BEGIN

   { map page containing diagnostic error information }

   ALLOCATE_MEMORY( page_ptr, 1024, PHYSICAL := diag_error_page,
                    STATUS := status);
   IF NOT ODD(status) THEN
      BEGIN
         handle_error( status);
         EXIT
      END;

        page_ptr_value := page_ptr :: INTEGER;
   memory_ptr :: INTEGER := page_ptr_value + mem_offset;
   data_ptr :: INTEGER := page_ptr_value + data_offset;
```

```
    { how many errors do we have? }
    error_count := memory_ptr^.count DIV 8 - 1;
    IF error_count > 0 THEN
       BEGIN
          WRITELN('*** Diagnostics detected', error_count,' error(s)');
          WRITELN;
          WRITELN('*** The following test(s) failed ***');
          WRITELN;
          FOR i := 1 TO error_count DO
             WRITELN(data_ptr^[i].upper_led, data_ptr^[i].lower_led)
       END
    ELSE
       WRITELN('*** No errors detected by diagnostics ***')

END.

END;
```

D

KAV30 Port Connectors

This appendix describes the KAV30 port connectors. It gives information on the following:

- Console and auxiliary port connectors (see Section D.1)
- AUI port connector (see Section D.2)
- SCSI port connector (see Section D.3)

D.1 Console and Auxiliary Port Connectors

This section describes the console and auxiliary port connectors. The console and auxiliary port connectors have the same layout and pin specification. Both ports use a modified modular jack (MMJ) connector. These ports are collectively known as serial line ports. Figure D–1 shows the serial line port connector layout.

Figure D–1 Serial Line Port Connector Layout

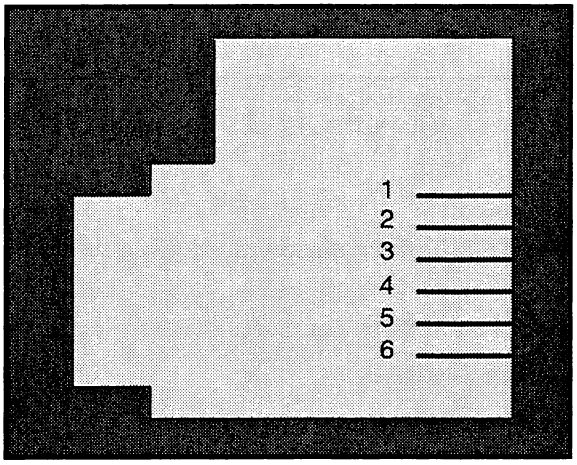


Table D–1 describes the serial line port connector pin specification.

Table D–1 Serial Line Port Connector Pin Specification

Pin	Signal Name	Explanation
1	+5 V	Ready out
2	TXD+	Transmit Data+
3	GND	Transmit Data-
4	RXD-	Receive data-
5	RXD+	Receive data+
6	Not used	Ready in

D.2 AUI Port Connector

This section describes the AUI port connector. Table D–2 describes the AUI port connector pin specification.

Table D–2 AUI Port Connector Pin Specification

Pin	Signal Name	Explanation
01	GND	Control In circuit Shield
02	COLL+	Control In circuit A
03	XMIT+	Data Out circuit A
04	GND	Data In circuit Shield
05	RCV+	Data In circuit A
06	GND	Voltage common
07	Not used	–
08	GND	Control Out circuit Shield
09	COLL	Control In circuit B
10	XMIT-	Data Out circuit B
11	GND	Data Out circuit Shield
12	RCV-	Data In circuit B
13	+12 V	Voltage Plus
14	GND	Voltage shield
15	Not used	–

D.3 SCSI Port Connector

This section describes the SCSI port connector. Figure D–2 shows the SCSI port connector layout. Table D–3 describes the SCSI port connector pin specification.

Figure D–2 SCSI Port Connector Layout

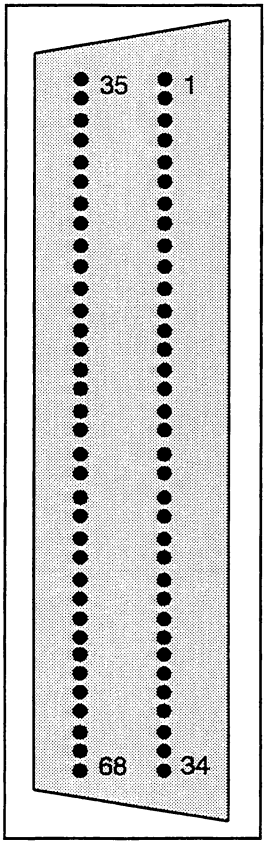


Table D–3 SCSI Port Connector Pin Specification

Pin	Signal Name	Explanation	Pin	Signal Name	Explanation
01	NC	No Connection	35	NC	No Connection
02	- DB(0)	Data 0	36	GND	Ground
03	GND	Ground	37	NC	No Connection
04	- DB(1)	Data 1	38	NC	No Connection
05	GND	Ground	39	NC	No Connection
06	- DB(2)	Data 2	40	GND	Ground
07	GND	Ground	41	NC	No Connection
08	- DB(3)	Data 3	42	GND	Ground
09	GND	Ground	43	NC	No Connection
10	- DB(4)	Data 4	44	NC	No Connection
11	GND	Ground	45	NC	No Connection
12	- DB(5)	Data 5	46	GND	Ground
13	GND	Ground	47	NC	No Connection
14	- DB(6)	Data 6	48	GND	Ground
15	GND	Ground	49	NC	No Connection
16	- DB(7)	Data 7	50	NC	No Connection
17	GND	Ground	51	NC	No Connection
18	- DB(P)	Data parity	52	NC	No Connection
19	ATN	Attention	53	NC	No Connection
20	- BSY	Busy	54	NC	No Connection
21	- ACK	Acknowledge	55	NC	No Connection
22	GND	Ground	56	GND	Ground
23	- RST	Reset	57	NC	No Connection
24	- MSG	Message	58	NC	No Connection
25	- SEL	Select	59	NC	No Connection
26	- C/D	Cmd/Data	60	NC	No Connection
27	- REQ	Request	61	NC	No Connection
28	- I/O	Input/Output	62	NC	No Connection

(continued on next page)

Table D–3 (Cont.) SCSI Port Connector Pin Specification

Pin	Signal Name	Explanation	Pin	Signal Name	Explanation
29	TERMPWR2	Term +5 V	63	NC	No Connection
30	GND	Ground	64	NC	No Connection
31	NC	No Connection	65	NC	No Connection
32	NC	No Connection	66	NC	No Connection
33	NC	No Connection	67	NC	No Connection
34	NC	No Connection	68	NC	No Connection

Glossary

The glossary defines some of the important terms used in this guide.

application program

A program that performs an end-user task.

AUI

Attachment unit interface. An IEEE 802.3 compliant compatibility interface.

auxiliary power supply

The source of power for the KAV30 battery backed-up RAM and KAV30 calendar/clock. The source of the auxiliary power supply can either be the KAV30 battery or the VMEbus standby power supply.

backplane

A printed circuit board at the rear of a crate which, by means of its attached connectors, mates with the modules and constitutes the crate segment.

back up

The process of making copies of the data stored on the disk, so that you can recover that data after an accidental loss.

big-endian device

A device based on the 68000 family of processors.

boot

To execute the boot loader program. This program is always present in the system when the power is on.

boot device

The device that contains the application program that is booted when the system starts up.

CPU

Central processing unit. The main unit of a computer that contains the circuits that control the interpretation and execution of instructions. The CPU holds the main storage, arithmetic unit, and special registers.

CSR bit

Control and status register bit. The CSR bits consist of both input bits and output bits. The CSR input bits report on the status of the KAV30 hardware, whereas the CSR output bits control the KAV30 hardware.

DAL bus

Data and address lines bus. A 32-bit multiplexed bus. The rtVAX 300 is the source of the DAL bus.

DECconnect

The Digital cabling system that extends Ethernet and terminal interconnections into offices and work areas.

DELNI

A local area network interconnect product that gives eight separate network interfaces from a single transceiver tap.

DESTA

A station adapter that acts as a ThinWire Ethernet transceiver. A DESTA adapter allows you to connect a station with a transceiver cable to ThinWire Ethernet.

diagnostics

The programs that detect and identify malfunctioning hardware.

DMA

Direct memory access. A method of accessing a device's memory without interacting with the device's CPU.

down-line load

To send a copy of a system image or other file over a line to the memory of a target node.

DRAM

Dynamic read-only memory. A read/write memory device in which data is retained by a continually circulating signal.

electromagnetic

Pertaining to the force fields generated through electrical processes.

EMI

Electromagnetic interference

EPROM

Erasable programmable read-only memory. EPROM is a type of read-only memory that can be erased, and so returns the device to a blank state.

ESD

Electrostatic discharge.

Ethernet

A type of local area network (LAN) based on Carrier Sense Multiple Access with Collision Detection (CSMA/CD).

FIFO

First-in/first-out. The order in which processing is performed. For example, a FIFO queue processes data on a first-come, first-served basis.

FIFO buffer

An area in which devices can store and retrieve data.

FPU

Floating point unit. A device that can process expressions that include decimal points.

ground

A voltage reference point in a system that has a zero voltage potential.

H4000

An Ethernet transceiver.

H4005

An Ethernet transceiver.

ISP

Interrupt stack pointer. The stack pointer for the interrupt stack. Unlike the stack pointers for process context stacks, which are stored in the hardware PCB, the interrupt stack pointer is stored in an internal register.

JEDEC

Joint Electron Device Engineering Council.

jumper

A short length of wire used to complete a circuit temporarily, or to bypass a circuit.

LIFO

Last-in/first-out. The order in which processing is performed. For example, a LIFO queue processes data on a last-come, first-served basis.

little-endian device

A device based on the Intel™ family of processors.

MMJ

Modified modular jack. A socket that accepts an MMP at one end and is attached to electrical circuits at the other.

MMP

Modified modular plug. A plug that connects to an MMJ.

module

A printed circuit board that contains electrical components and electrically conductive pathways between components. A module stores data or memory, or controls the functions of a device.

NCP

Network control program. The block that contains the necessary information to set up a virtual circuit or to accept or reject a request to set up a virtual circuit.

port

The physical connector.

printed circuit board

A circuit wired or etched on a phenolic or fiberglas material, complete with all the necessary components for performing a specific function.

PTT

Postal, Telephone, and Telegraph.

RAM

Random-access memory. A read/write memory device.

RISC

Reduced instruction set computer.

ROM

Read-only memory. A memory in which information is permanently stored at the time of production and is not alterable by computer instructions.

SCSI

Small computer systems interface. An interface designed for connecting disks and other peripheral devices to computer systems. SCSI is defined by an American National Standards Institute (ANSI) standard.

SELV

Safety Extra Low Voltage. According to UL 1950 (US) and EN 60950 (Europe); a SELV circuit is a secondary circuit, which is so designed and protected, that under normal and single fault conditions the voltage between any two accessible parts, or between one accessible part and the protective earthing terminal equipment for Class I equipment does not exceed a safe value.

Under normal conditions this limit is either 42.4 V peak, or 60 V dc. Under fault conditions higher limits are specified in this standard for transient deviation.

SGEC

Second-generation Ethernet controller.

SGM

Scatter-gather map. A means of allowing either of the following types of data transfer:

- From pages in memory that are not contiguous to contiguous blocks on a bus
- From contiguous blocks on a bus to pages in memory that are not contiguous

slot

A position where a printed circuit board can be installed into a backplane.

SLU

Serial line unit. A port you can use to connect serial line devices such as terminals and printers to the KAV30.

standard Ethernet

An IEEE 802.3 compliant Ethernet network composed of standard Ethernet cable as opposed to thin Ethernet cable.

system

A combination of system hardware, software, and peripheral devices that performs specific processing operations.

terminator

A special connector that gives electrical termination of a bus.

ThinWire

A trademark used to describe Digital's IEEE 802.3 compliant products used for local distribution of data communication.

transceiver

A device that gives a single physical connection between standard Ethernet and Ethernet communication equipment.

UART

Universal asynchronous receiver/transmitter. A UART receives or transmits data using serial lines.

VSB

VME subsystem bus.

802.3

An IEEE standard describing the physical and data link layers of a local area network based on bus typology and Carrier Sense Multiple Access/Collision Detect (CSMA/CD).

Index

A

B

C

D

E

T

U

V

W

digital