EK–SCSIS–OV–001

Small Computer System Interface

An Overview

Revision/Update Information: Version 1.0

digital equipment corporation
maynard, massachusetts

First Printing, December 1989

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation.

Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

Any software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license. No responsibility is assumed for the use or reliability of software or equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

© Digital Equipment Corporation 1989.

All rights reserved.
Printed in U.S.A.

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

The following are trademarks of Digital Equipment Corporation:

DEC	MicroVAX	VAX DOCUMENT
DECserver	PDP	VAXELN
DECstation	RSTS	VAXserver
DECsystem	RSX	VAXstation
DECUS	RT	VMS
DECwriter	ULTRIX	digital™
DIBOL	UNIBUS	
DIGITAL	VAX	

The following are third-party trademarks:

NCR is a registered trademark of the NCR Corporation. ANSI is a registered trademark of the American National Standards Institute, Inc.

S1259

This document was prepared with VAX DOCUMENT, Version 1.2.

Contents

About This Guide v

1 Digital and SCSI

Why Digital Selected SCSI 1–2
Background 1–3
Commitment to Open Systems 1–3

2 Evaluating SCSI

Development Considerations 2–1
Systems with SCSI 2–2
NCR 5380 SCSI Interface 2–3
System Interface Interconnect (SII) 2–4
Features Implemented 2–4
Customer Support Services 2–6

3 Understanding SCSI—An Overview

The SCSI Bus 3–1
SCSI Bus Phases 3–3
SCSI Bus Signals 3–5
SCSI Device Controllers 3–7
The Common Command Set 3–7

4 Operating System Support

VMS SCSI Device Support 4–1
VMS SCSI Third-Party Device Support Mechanisms 4–3
SCSI Bus to IEEE488 Bus 4–5
Planning Software Support for the IEEE488 Bus-Converter 4–5
IEEE488 Bus-Converter Class Driver Interface Data Flow 4–6
ULTRIX SCSI Device Support 4–8
VAXELN SCSI Device Support 4–9
VAXELN SCSI Third-Party Device Support Mechanisms 4–9
Disk Device Class Driver 4–10
Message Class Driver 4–10
User-Written Class Driver 4–10

A Associated Documents

Glossary

Index

Figures

3–1 SCSI Configuration 3–2
3–2 Bus Phases for INQUIRY Command 3–4
4–1 VMS SCSI Class/Port Interface 4–2
4–2 Data Flow from User to IEEE488 Device 4–7

Tables

2–1 Digital Systems and Associated Host Adapter 2–3
2–2 Digital's Implementation of SCSI 2–5
3–1 SCSI Bus Signals 3–6

About This Guide

Purpose of This Guide

This guide encourages third-party vendors to develop complementary Small Computer System Interface (SCSI) peripheral devices not currently available through Digital. Examples of peripheral devices include: copying, imaging, and communication devices; scientific test equipment; and any other device that increases the customer's productivity in the workplace.

To help vendors understand what is required to develop devices for Digital's SCSI-based systems, this guide provides an overview of Digital's implementation of the American National Standards Institute (ANSI), SCSI-2 specification.

The following materials will help the development process:

- *American National Standard for Information Systems—Small Computer System Interface-2 (SCSI-2)* specification. See Appendix A for ordering information.

Note *At this writing, the Small Computer System Interface is under development. The draft SCSI-2 specification (Revision 10b) should be the official guide to what is implemented by a third-party device.*

- The *Small Computer System Interface: A Developer's Guide*, a companion piece to this *Overview*.

  The *Small Computer System Interface: A Developer's Guide* specifies the technical requirements associated with Digital's implementation of SCSI. It should be used in conjunction with the SCSI-2 specification. The SCSI-2 specification used alone

does not guarantee device compatibility with Digital computer systems. See Appendix A for ordering information.

- The *VMS Version 5.3 SCSI Device Support Manual* describes the software interfaces the VMS operating system provides that allow third-party devices to be attached to VAXstation and MicroVAX systems.

Appendix A of this guide contains a list of software manuals and additional ordering information associated with Digital's implementation of the Small Computer System Interface (SCSI).

In the U.S., Digital documentation can be ordered by telephoning 1-800-DIGITAL. In other countries, contact your local sales office.

Who Should Use This Guide

This guide is intended for business, marketing and engineering representatives of third-party vendors who wish introductory information about Digital's implementation of the Small Computer System Interface.

The guide addresses several people within an organization. An executive may choose to give this guide to a technical person who will evaluate the opportunity of connecting to Digital's SCSI-based systems.

Structure of This Guide

This guide contains four chapters and one appendix.

Chapter 1 presents background information about Digital's association with SCSI.

Chapter 2 lists the Digital systems on which SCSI is implemented, describes the SCSI chips used on these systems, and provides suggestions for evaluating a peripheral device to be used on a Digital system.

Chapter 3 provides an overview of the Small Computer System Interface and a description of its major components.

Chapter 4 describes current operating system support.

Appendix A lists the manuals associated with Digital's implementation of the Small Computer System Interface.

The Glossary contains definitions of technical terms used in the manual.

Conformance to Standards

The ANSI SCSI-2 specification (revision 10b) should be the official guide to what features should be implemented on the device and how to implement them. Digital's requirements for third-party SCSI devices are based on the evolving SCSI-2 specification. SCSI-2 is compatible with the SCSI-1 standard; SCSI-2 clarifies much of the material in SCSI-1.

Digital has designed its hardware and software so that it adheres to the SCSI-2 specification and yet provides as much flexibility as possible. To work with Digital systems, third-party devices must implement all the mandatory features of the SCSI-2 interface. Third-party devices may implement optional features, as long as the implementation follows the SCSI-2 specification. Devices may implement vendor-specific features as long as the features are implemented in areas clearly designated as such by the SCSI-2 specification.

Development Considerations

Products licensed or sold by Digital Equipment Corporation have been tested by Digital. Products should operate with Digital computer systems if they are developed using Digital SCSI documentation and implemented according to Digital requirements as specified in this guide. The proper operation of products that have been developed by third-party vendors cannot be assured by Digital. Many third-party products require additional features that are not part of Digital's software. These products may not be fully functional with Digital's software as licensed and sold. Digital customers who use unsupported products may not qualify for Digital customer service programs.

Third-party peripheral devices may be attached to Digital SCSI-based systems; however:

- Performance and functionality with the operating system may not be optimized.
- Data integrity and reliability are potentially at risk due to loss of data, error rate, data corruption, and disabling of the bus.
- Quality of devices may not meet Digital standards.

A product that is not supplied by Digital may appear to work in a specific configuration; it may not work with future updates of Digital hardware and software.

Conventions

The following conventions are used in this guide:

boldface	Boldface type represents the introduction of a new term. New terms are defined in the Glossary.
Note	Notes provide information about the current topic and are of special interest to vendors of peripherals.
SCSI	The term SCSI is used wherever it is not necessary to distinguish between SCSI-1 and SCSI-2.
UPPERCASE	Uppercase text represents the names of signals, messages, and commands; for example, INQUIRY.

1

Digital and SCSI

Digital Equipment Corporation, a worldwide leader of integrated computer systems, is providing an opportunity to developers of Small Computer System Interface (**SCSI**) peripheral products to connect to Digital's SCSI-based systems. This will enable users of Digital products to choose the complementary peripheral devices they need to increase productivity in their place of business. The benefits extend to Original Equipment Manufacturers (**OEMs**), who can offer more packaging options to customers. SCSI is the solution to a customer's connectivity needs.

To date, the SCSI peripheral market has been primarily concentrated on the PC environment. Now, however, developers can take advantage of another segment of the market. The potential exists to make SCSI products available to a significant number of VMS operating system users. In the near future, Digital plans to extend this opportunity to ULTRIX users as well. Digital systems that support the SCSI bus are: MicroVAX 3100, VAXstation 3100, VAXstation 3520/3540, DECstation 2100/3100 and DECsystem 3100.

The Small Computer System Interface is an emerging standard that allows for flexibility in implementation by host system producers. Digital chose SCSI because of its wide acceptance throughout the industry. The industry is choosing SCSI because its architecture is easy to understand and because it requires fewer chips and is therefore cost-effective. Digital's current implementation of the standard includes the VMS Version 5.3 Small Computer System Interface (SCSI) Device Support software product, which provides a mechanism to connect with Digital's systems.

Digital is one of the first large computer companies to offer information to SCSI developers about its implementation of the ANSI **SCSI-2** specification. The information provided in this document is designed to assist you in making the decision to develop for Digital's SCSI-based systems. The presentation addresses several people within an organization. An executive may not need any other information than what appears in this chapter. The executive can then pass this document to a manager, who in turn may need the input of an engineer. All three can evaluate the opportunity individually or as a decision-making team.

Why Digital Selected SCSI

The Small Computer System Interface permits a customer to keep a computer current with changing technology. Digital is implementing SCSI to help customers connect their computer systems to the wide variety of **peripheral devices** currently available in the marketplace.

Digital's SCSI strategy is simple. Digital wants its customers to have:

- The resources they need to accomplish their work.
- Flexibility in defining their own low-end computing environment.
- The best platform for their SCSI devices, one that is easy to implement and simple to use.
- Protection for their financial investment. With SCSI, customers do not have to add special hardware to their systems in order to connect a peripheral device.

Digital wants customers and third-party vendors to have access to Digital's SCSI development and debugging tools. Customers and vendors can write their own **class driver** in order to tightly integrate their SCSI-based devices into the operating system environment.

Developers of SCSI peripheral devices benefit too—SCSI provides a cost-effective means of personalizing a product to make it compatible with Digital systems.

Background

The X3T9 committee of the American National Standards Institute (**ANSI**) has published a set of requirements for the Small Computer System Interface (SCSI) entitled: *American National Standard for Information Systems—Small Computer System Interface-2 (SCSI-2)*. The specification defines the electrical and functional requirements for connecting physically small computers to peripheral devices that use the SCSI bus. The requirements provide a standard means through which third-party peripheral devices can connect to Digital computer systems.

Commitment to Open Systems

Digital is committed to achieving the integration of applications across multivendor platforms. The ability to reach this goal depends on the degree to which standards are adopted throughout the computer industry. Digital understands the advantages to using standard interfaces and protocols to manage information. Digital is a member of the X3T9 committee of ANSI and works actively to help bring SCSI solutions to the customer. Digital follows an established, consistent, and conscientious process of standards development and compliance.

2

Evaluating SCSI

Digital's operating systems provide mechanisms for the attachment of third-party devices to Digital systems. Depending upon the peripheral device to be supported, the requirements of the application programs that will access the device, and programming resources available, a third-party vendor may elect to use either an application level interface or to write a device-specific driver.

To write a device driver, programmers should have detailed knowledge of the operating system and the ANSI SCSI-2 specification.

Engineers and system programmers who are responsible for implementing third-party device support will need the information contained in the Digital manual *Small Computer System Interface: A Developer's Guide* as well as operating system documentation.

Development Considerations

Before you decide to connect a third-party peripheral device to a Digital system, you should consider the following:

1. Ensure that the hardware features of the Digital computer system and the SCSI peripheral device are compatible.
2. Ensure that the electrical and data link levels are compatible.
3. Determine if the same commands are interpreted by both products. The peripheral device may recognize more commands than the Digital computer system.

4 Prepare to write an application-level driver that follows operating system rules. Remember, the same operating system driver will not work on all machines—a driver needs to be adapted for multiple operating systems. See Chapter 4 for information on the VMS third-party device support product.

5 After you have written a software driver, determine if you can connect your **SCSI device** to each of the Digital systems listed in Table 2–1. Try to run applications on the Digital systems with your peripheral device attached. Ensure that your peripheral device is fully functional. Determine if all error conditions are being handled correctly.

6 Hardware and software driver support is the responsibility of the third-party vendor. Be prepared to provide customer support for your device.

Systems with SCSI

Digital is a leading seller of workstations based on the Complex Instruction Set Computer (**CISC**) and Reduced Instruction Set Computer (**RISC**) architectures. Digital delivers general-purpose and distributed computing solutions, networking, and application integration. Now, Digital adds SCSI as a solution to the customer's connectivity needs.

The MicroVAX 3100 and VAXstation 3100 systems are uniprocessing systems, and provide access to one or two SCSI buses, each of which are under control of an NCR 5380 SCSI controller chip.

The DECstation 2100 and 3100 systems are uniprocessing systems, and provide access to a single SCSI bus by means of Digital's **System Interface Interconnect (SII)** chip.

The VAXstation 3520/3540 systems are multiprocessing systems that provide access to a single SCSI bus by means of the SII chip.

Table 2–1 lists the Digital systems with a **SCSI port**, the **host adapter** used with each system, and the expected data transfer rate.

Table 2–1 Digital Systems and Associated Host Adapter

System	Host Adapter	Maximum Asynchronous Transfer Rate[1]	Maximum Synchronous Transfer Rate[1]
VAXstation 3100 and MicroVAX 3100	NCR 5380	1.5 MB/s	Not Implemented
DECstation 2100/3100	DEC SII	2 MB/s	4+ MB/s
VAXstation 3520/3540	DEC SII	2 MB/s	4+ MB/s

[1]Optimized rates are shown

Note *Maximum asynchronous transfer rate is a function of cable length, number of peripheral devices, and device propagation delays.*

SCSI-based systems use a host adapter as the information path between host system memory and the SCSI bus. The information path extends not just between the host adapter and a device controller on the SCSI bus, but the path runs all the way from host memory to the peripheral device performing the operation. The host adapter permits the exchange of command, data, and status information between the host and the peripheral device.

NCR 5380 SCSI Interface

The NCR 5380 SCSI interface chip is a 40-pin N-channel metal oxide semiconductor (NMOS) device designed to work with the SCSI bus as defined by the ANSI SCSI-2 specification. The chip is used in Digital's VAXstation 3100 and MicroVAX 3100 systems as the host adapter, and functions as either an **initiator** or **target**.

The NCR 5380 chip communicates with the system microprocessor as a peripheral device. The chip is controlled by reading and writing several internal registers addressed as standard or memory-mapped I/O. Direct Memory Access (**DMA**) transfers require a minimum of processor intervention, because the NCR 5380 controls the necessary handshake signals.

System Interface Interconnect (SII)

The Small Computer System Interface is the same on the DECstation 2100 and 3100 workstations and the VAXstation 3520/3540 systems. The interface is based on a gate array developed by Digital, and referred to as the System Interface Interconnect (SII).

The SII gate array manages the following activities:

- SCSI bus selection
- Bus to memory transfers
- Disconnect and reconnect commands

The SII gate array does not directly drive the SCSI bus. The SII, together with the NCR 8310 (or 8311) SCSI transceiver chip, drives the SCSI bus. The SII gate array supports synchronous data transfers of 4.0+ megabytes per second. The processor monitors general operations through the SII internal control and status registers.

The SII communicates with the system microprocessor as a peripheral device. The chip is controlled by reading and writing several internal registers addressed as standard or memory-mapped I/O. Direct Memory Access (DMA) transfers require a minimum of processor intervention, because the SII controls the necessary handshake signals.

Features Implemented

Table 2–2 lists the major features in Digital's implementation of the Small Computer System Interface.

Table 2–2 Digital's Implementation of SCSI

FEATURE	Currently Not implemented	IMPLEMENTED
Data transfer width	16 and 32 bit-wide SCSI	8-bit SCSI
Transfer mode	Fast data transfer	1.5 MB/s (async), 4+ MB/s (synch)
Drivers	Differential	Single-ended drivers and receivers
Termination	Internal termination	External termination of last device on bus
Cable	25 meters long	6 meters long
	68-conductor B cable	50-conductor A cable
Initiators	Multiple initiators	Single initiator with multiple targets
Performance	Command queuing (tagged commands)	
		Disconnect and reconnect
	Soft Reset to clear devices from bus	Hard Reset to clear devices from bus. Also, ABORT, BUS DEVICE RESET.
	Dynamic reconfiguration of devices on bus	
Command set (class driver)		Common Command Set for disks
Operating system support for SCSI devices not supplied by Digital	ULTRIX application level interface. VAXELN application level interface.	VMS Version 5.3 third-party device development support
Operating system support for SCSI disk and tape drives supplied by Digital		ULTRIX Workstations Software V2.1 for MicroVAX 3100, DS2100, DS3100, VS3100, and VS3520/3540 systems
		VMS driver support on MicroVAX 3100, VS3100, and VS3520/3540 systems

Customer Support Services

SCSI has been widely adopted by manufacturers for a variety of peripheral devices. The SCSI specification is so broad in scope that not all implementations are compatible. Many peripheral devices require features that are not part of Digital's hardware and/or software, and as such, may not be fully functional with Digital's software as licensed and sold. Because of this, Digital does not offer direct third-party vendor operation and remedial services. The vendor is responsible for ensuring that the product works with the Digital systems listed in Table 2–1, so that Digital customers who buy the product may qualify for Digital's customer service programs.

Products sold by Digital Equipment Corporation are tested by Digital and receive remedial service. Products that have not been tested by Digital may qualify for remedial support through one or more of the Customer Support Services currently offered by Digital.

Digital provides service to its customers as follows:

- **Vendor Equipment Services (VES)** is the Digital group responsible for developing and managing support plans and programs to service selected vendor products. DECservice, Basic Service, and Digital Servicenter Agreements on vendor products attached to Digital processors are available through DECompatible Service agreements.
  - Category A products are treated as if they are Digital products with worldwide support.
  - Category B covers vendor products supported only in specific countries, such as Australia.
  - Category C customers have vendor equipment that is not included in category A or B. A Digital branch can choose to service the vendor equipment if there is a sufficient quantity of similar equipment within the local area to justify the investment.

  Vendor Equipment Services selects high-volume, non-competitive products based upon field demand, which complement Digital product offerings.

- The **Vendor Application Service (VAS)** program expands the number of software products that Digital supports by qualifying the products engineered by vendors. For the most part, the service available to the customer is identical to what is offered by currently available software product services for standard Digital products. A partnership service arrangement between Digital and the vendor is designed to increase market penetration, to better utilize support staff, and to enhance customer satisfaction.
- Digital offers a **Desktop Service Program** to help customers install, repair, and integrate desktop products from Original Equipment Manufacturers (OEMs). The program consists of maintenance, start-up services, direct-access advisory services, and the assembly of hardware at customer sites away from the actual desktop where the equipment will be used. The program is available to customers in major metropolitan areas with a minimum of 100 units. (Digital defines ''units'' as personal computers, terminals, and printers.)

Contact your Digital sales representative for more information on the Customer Support Services.

3

Understanding SCSI—An Overview

The Small Computer System Interface is a parallel I/O bus and protocol that permits the connection of a variety of peripherals including disk drives, tape drives, modems, printers, scanners, optical devices, test equipment, and medical devices to a host computer.

The SCSI Bus

The SCSI **bus** connects all parts of a computer system so that they can communicate with each other. The bus frees the host **processor** from the responsibility of I/O internal tasks.

The SCSI protocol is a peer-to-peer relationship: one device does not have to be subordinated to another device in order to perform I/O activities. A total of eight devices can be connected to the bus simultaneously. Only two of these devices can communicate on the bus at any given time.

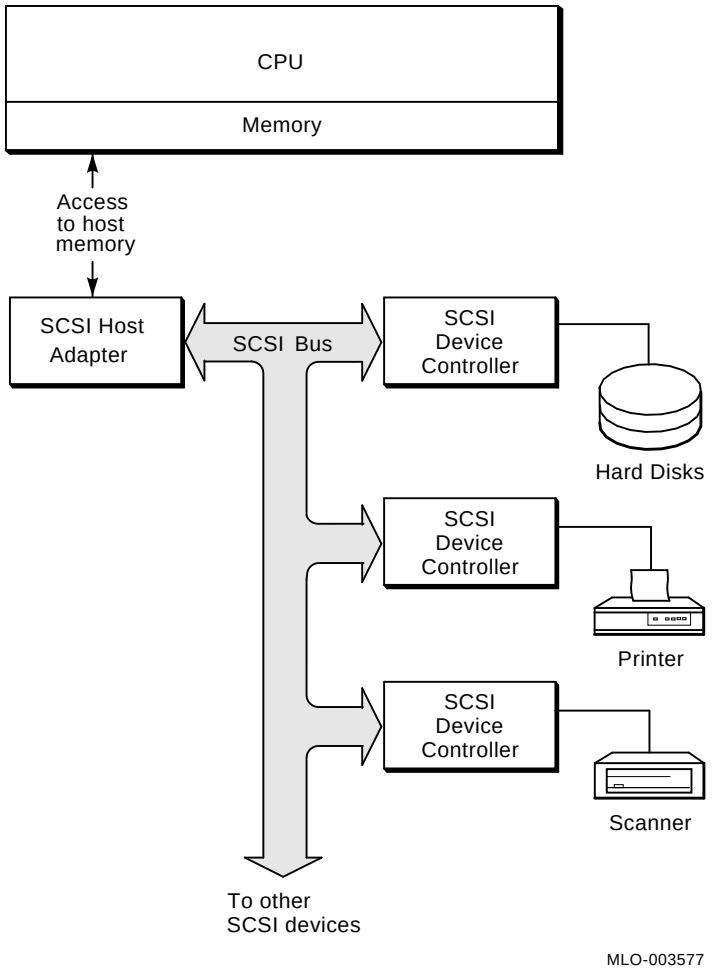
Note *The word ''device'' refers to the SCSI host adapter or a device* ***controller*** *connected to the SCSI bus; it does not refer to the peripheral device.*

A unique **SCSI device ID** (7–0) is assigned to the SCSI host adapter and to each device controller.

One of the devices on the bus must be the initiator (usually the host adapter). The other device is the target (usually the peripheral device controller). When two devices communicate on the bus, one device initiates the communication to the target, and the target performs the task. SCSI devices usually have a fixed role as an initiator or a target, although some devices can perform both roles.

The SCSI interface on Digital systems is a single-ended configuration with a cable length of up to six meters. The SCSI cable is always terminated on the SCSI host adapter module; an external terminator must be provided for the last peripheral device placed on the bus. Figure 3–1 shows a typical SCSI configuration.

Figure 3–1 SCSI Configuration

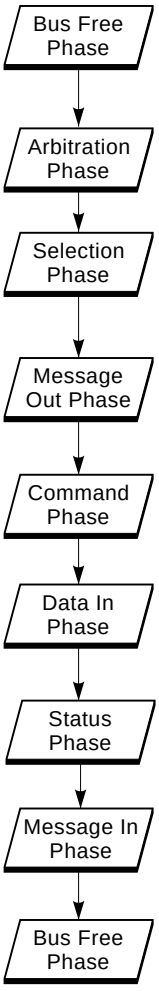


SCSI Bus Phases

The Small Computer System Interface bus can be time-shared, which results in greater usage of bus bandwidth. This is how it works: while one device is using the bus, other devices may be active and performing internal activities. Devices do not use the bus unless they are involved in data transfer or have **status** to report. Devices may **disconnect** from the bus while time-consuming activities internal to the device are occurring. As soon as a device is ready to resume communication, the device can **arbitrate** for the bus (when the bus is free) to reattach to the host. System performance is significantly increased when devices disconnect and **reconnect** to the bus.

During the bus phases (Figure 3–2), devices must first contend for access to the bus. Then a physical path is established between the initiator and target. Remember, the SCSI bus cannot be in more than one phase at a time.

Figure 3–2 Bus Phases for INQUIRY Command



On a busy system, the SCSI bus may be free for as little as 1.2 µs while there is no device requesting the bus. Or, it could remain in the bus free state indefinitely.

A device can arbitrate and win the bus in 3.6 µs or less. Devices that lose arbitration can retry when the bus is free.

The Selection phase can occur in 580 ns. If the target does not respond, the bus is free in about 250 ms. The target controls the bus after the Selection phase.

This is the first Information Transfer phase in the connection. It allows the initiator to send an IDENTIFY message to the target. Messages are always transferred asynchronously.

Six INQUIRY command bytes are transferred asynchronously from the initiator to the target.

The target responds with INQUIRY data. The data is transferred synchronously if both the target and the initiator have previously established a synchronous data transfer agreement.

The target sends a single status byte asynchronously.

The last information that is transferred in the connection is typically the COMMAND COMPLETE message.

The target releases all signals within 10 µs of the initiator's acknowledgement of the previous message. The initiator releases any signals it may be driving within 800 ns of its detection of the BSY signal.

MLO-003607

SCSI Bus Signals

The SCSI bus uses eighteen signals. Nine are control signals used to develop logical bus phases, and nine are data signals, including parity, for messages, commands, status, and data.

The state of the SEL, BSY, and I/O signals and the sequence of the phases determine when the Bus Free, Arbitration, Selection, and Reselection phases are entered. Table 3–1 describes the nine control signals.

- The Selection or Reselection phase can be entered only from the Arbitration phase.
- The Arbitration phase can be entered only from the Bus Free phase.
- The Bus Free phase can be entered from any of the other phases (although some transitions are caused by errors).

To determine the current phase, you need to know information about the previous phase and the state of the signals. The initiator and target drive these signals to change from one phase to another phase.

Table 3–1 SCSI Bus Signals

Signal	Description
BSY (BUSY)	Bus is being used.
SEL (SELECT)	Initiator selects a target or a target reselects an initiator.
C/D (CONTROL/DATA)	A signal driven by a target to indicate whether or not control or data information is on the data bus.
I/O (INPUT/OUTPUT)	A signal driven by a target to control the direction of data movement on the data bus.
MSG (MESSAGE)	A signal driven by a target during the Message phase.
REQ (REQUEST)	A signal driven by a target to request a REQ/ACK data transfer handshake.
ACK (ACKNOWLEDGE)	A signal driven by an initiator to acknowledge a REQ/ACK data transfer.
ATN (ATTENTION)	A signal driven by an initiator to indicate the Attention condition (initiator has a message for the target).
RST (RESET)	A hard Reset condition.
DB(7–0,P) (DATA BUS)	Eight data-bit signals, plus a parity-bit signal that form a Data Bus. DB(7) is the most significant bit and has the highest **priority** during the Arbitration phase. Bit number, significance, and priority decrease downward to DB(0).

SCSI Device Controllers

A SCSI device controller for peripherals can be thought of as a small independent computer with its own on-board memory and processor. It may either be separate from or embedded in a peripheral device. It works like a computerized traffic light, regulating the flow of data to and from devices on the SCSI bus. The device controller communicates with the SCSI bus on one end, and communicates with the peripheral device at the other end. It monitors the data for transmission errors and, when necessary, functions as a translator, changing data from serial format to parallel format so that it is understood by the host computer. SCSI device controller boards operate with disk drives, printers, tape drives, optical devices, scanners, and other peripherals.

The Common Command Set

In an effort to clarify many of the optional features of SCSI, leading disk vendors have developed software for disks around a common set of features, commands, and messages. The common features have been incorporated into an industry-standard format: a common command syntax called the **Common Command Set (CCS)**. CCS is a subset of 18 commands that adds control over device formatting and error recovery.

Digital has designed its hardware and software so that it adheres to the SCSI-2 specification and yet provides as much flexibility as possible. Digital follows the ANSI SCSI-2 specification, which includes the Common Command Set. Developers who create a software interface to Digital systems must implement all the mandatory features of the SCSI-2 interface. In addition, developers may implement optional features, as long as the implementation follows the SCSI-2 specification. Vendor-specific features may be implemented as long as the features are implemented in areas clearly designated as such by the SCSI-2 specification.

4

Operating System Support

Digital is committed to providing SCSI connectivity on its workstations and operating systems, as follows:

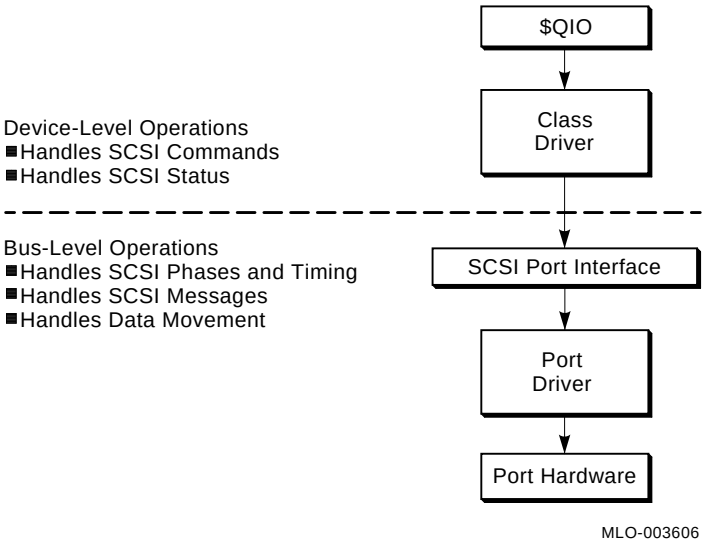
- The VMS operating system supports the SCSI bus on the MicroVAX 3100, VAXstation 3100, and the VAXstation 3520/3540 systems. A third-party vendor can develop an interface to a SCSI device using the VMS generic SCSI class driver.
- The ULTRIX operating system supports the SCSI bus on the MicroVAX 3100, VAXstation 3100, VAXstation 3520/3540, DECstation 2100, DECstation 3100, and DECsystem 3100 systems. See the ULTRIX Software Product Description for a list of SCSI devices that can be used on these systems.
- VAXELN Version 4.1 provides SCSI disk and generic device support on the VAXstation 3100 and MicroVAX 3100 systems.

VMS SCSI Device Support

The VMS operating system employs a class/port driver architecture to communicate with devices on the SCSI bus. The class/port design allows the responsibilities for communication between the operating system and the device to be cleanly divided between two separate driver modules (see Figure 4–1).

The SCSI **port driver** transmits and receives SCSI commands and data. It knows the details of transmitting data from the local processor's **SCSI port** hardware across the SCSI bus. Although the SCSI port driver understands SCSI bus phases, protocol, and

Figure 4–1 VMS SCSI Class/Port Interface



timing, it has no knowledge of which SCSI commands the device supports, what status messages it returns, or the format of the packets in which this information is delivered. The port driver is a communications path; when it is directed by a SCSI class driver, it forwards commands and data from the class driver onto the SCSI bus to the device. On any given MicroVAX or VAXstation system, a single SCSI port driver handles bus-level communications for all SCSI class drivers that may exist on the system.

The SCSI class driver acts as an interface between the user and the SCSI port, translating an I/O function as specified in a user's $QIO request to a SCSI command targeted to a device on the SCSI bus. Although the class driver knows about SCSI command descriptor buffers, status codes, and data, it has no knowledge of underlying bus protocols or hardware, command transmission, bus phases, timing, or messages. A single class driver can run on any given MicroVAX or VAXstation system, in conjunction with the SCSI port driver that supports that system. The VMS operating system supplies a standard SCSI disk class driver and a standard SCSI tape class driver to support its disk and tape SCSI devices.

VMS SCSI Third-Party Device Support Mechanisms

The VMS operating system provides the following three mechanisms to allow a SCSI device that is not supplied by Digital to be attached to a MicroVAX or VAXstation system. The implementor of support for the vendor-developed SCSI device can select the most appropriate method, based on the capabilities of the device, the needs of its end users, and available programming resources.

- The design of the standard VMS disk class driver allows it to control most disk drives that conform to the ANSI SCSI-2 specification. Because the ANSI SCSI-2 specification is not as well defined for tapes as for disks, the standard VMS tape class driver may or may not work with a specific tape drive not supplied by Digital.

Note *The ANSI SCSI-2 specification allows some flexibility in certain implementation details and omits other details. As a result, implementations of the SCSI standard may differ from manufacturer to manufacturer and from device to device. Although the VMS operating system allows the use of devices that are not supplied by Digital with the standard VMS disk and tape class drivers, the fact that a specific device may or may not operate correctly in this manner does not imply VMS operating system support of the device.*

Digital cannot guarantee that a vendor-developed device that currently works with a VMS class driver will continue to work with subsequent releases of the VMS operating system.

To ensure that devices that are not supplied by Digital will work properly in a VMS environment, Digital encourages the use of an established and supported VMS interface, such as those described below.

- An application program can send commands to, receive status from, and exchange data with a device on the SCSI bus by using the VMS generic SCSI class driver. The VMS operating system defines a special Queue I/O Request ($QIO) system service interface that allows an application to pass SCSI command packets to the device through the generic SCSI class driver and the VMS SCSI port driver.
- A third-party SCSI class driver, in conjunction with the VMS SCSI port driver, can supply the level of support most closely tailored to the capabilities of the device. By writing a SCSI class driver, a system programmer can implement

device-specific error handling and a simple, robust $QIO interface.

Because the VMS operating system provides a special set of macros that initialize the SCSI port and transfer commands and data to a SCSI device, the programmer of a SCSI class driver can focus on coding details related to device capabilities. The VMS operating system further facilitates the writing of a SCSI class driver by including the online sources of a template SCSI class driver.

When selecting between writing an application program that uses the VMS generic SCSI class driver and writing a third-party SCSI class driver, the implementor of SCSI device support should consider the following factors:

- Because the VMS generic SCSI class driver provides access to the SCSI device from application code, the programmer of an application that uses it must be familiar with a high-level language, have some I/O programming skill, and have a thorough understanding of VAX MACRO and VMS driver internals.
- The VMS generic SCSI class driver uses a fixed $QIO interface to the SCSI port, requiring an application to pass a SCSI command descriptor block to the device by means of a single I/O function, IO$_DIAGNOSE. By contrast, a product-specific SCSI class driver can define a unique $QIO interface that conceals the details of SCSI command format from application programs.
- A programmer typically can develop an interface to a SCSI device more quickly by using the VMS generic SCSI class driver than by developing a third-party SCSI class driver. Because device drivers are tightly integrated into the VMS operating system and run in a privileged mode at high **IPL**, coding errors in the driver can result in system crashes. Because the VMS generic SCSI class driver is an established system interface, a programmer using it can spend less time integrating the code into the operating system and more time working on the interface.
- A vendor-developed SCSI class driver can write entries to an error log buffer, thus allowing the programmer to use the VMS Error Log Utility as a debugging aid.

- A vendor-developed SCSI class driver can implement error recovery mechanisms that are closely associated with the abilities of the device. It can service a device error within the single $QIO request that initiated the transaction to the device.

  Because the generic SCSI class driver has no knowledge of specific device errors, an application using that driver must manage device-specific errors itself. To service an error returned on a single transaction, the application must issue additional $QIO requests and initiate further transactions to the device.

- The SCSI **asynchronous event notification (AEN)** protocol is available only to vendor-developed SCSI class drivers.

For information on how to program to the VMS generic SCSI class driver's $QIO interface and how to write a third-party SCSI class driver, see the *VMS Version 5.3 SCSI Device Support Manual*.

SCSI Bus to IEEE488 Bus

The IEEE488 Bus-Converter (IEZ11) is a hardware and software product that permits the connection of test equipment, measurement instruments, and other IEEE488-compatible peripheral devices to either MicroVAX 3100 or VAXstation 3100 systems by means of an IEEE-specific VMS SCSI class driver. The IEEE488 Bus-Converter meets both the ANSI X3T9.2 SCSI-2 requirements and the IEEE488-1978 standard. The implementation decisions made during the development of this product may be helpful to vendors who would like to develop similar support for their own SCSI products on a VMS system.

Planning Software Support for the IEEE488 Bus-Converter In order to provide software support for the IEEE488 Bus-Converter, Digital developed a VMS SCSI488/D class driver. A user-written specific driver can often maximize performance in time-critical applications.

The VMS interface to the generic class driver allows a programmer to construct SCSI command packets in an application program and pass them (and other relevant information) to the device by means of a single I/O function directed to the class driver. The class driver developed to support the IEEE488 Bus-Converter associates each SCSI command supported by the device with a separate I/O function.

Planning the engineering effort for the production of a SCSI class driver involves the following tasks:

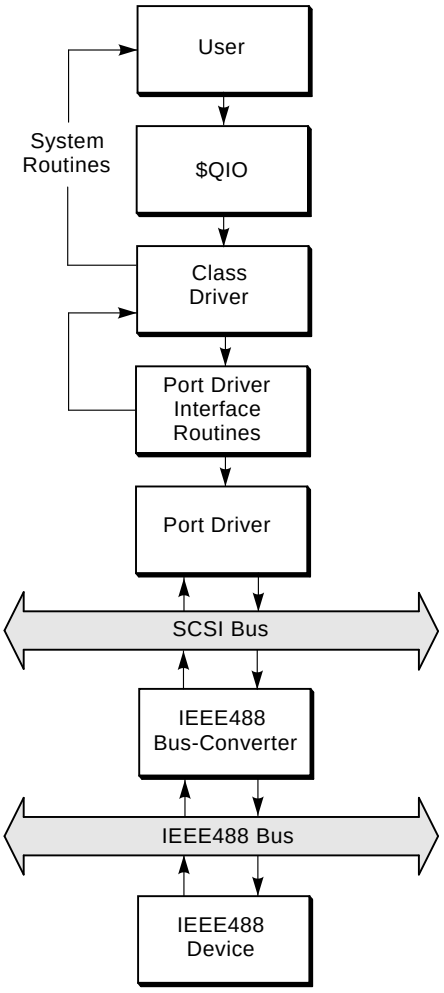
- Ascertaining the level of programming expertise required. Familiarity with the VAX MACRO assembler and some knowledge of VMS internals are essential to writing a VMS device driver. Documentation and training are available to help programmers become competent in these areas.
- Understanding the SCSI device's architecture and capabilities. The IEEE488 Bus-Converter responds to commands that configure the IEEE488 bus, control device activity on the bus, transfer data on the bus, and determine the status of devices on the bus.
- Defining the SCSI commands supported by the device.
- Outlining a user I/O interface that reflects the device's command set. The class driver that supports the IEEE488 Bus-Converter responds to a set of 24 I/O functions that reflect the capabilities of its command set.
- Determining whether the device has any special features.

IEEE488 Bus-Converter Class Driver Interface Data Flow When the IEEE488 Bus-Converter is operational in a MicroVAX 3100 or VAXstation 3100 system, the following events occur:

1. A user issues a VMS Queue I/O Request ($QIO) system service to the converter, supplying an appropriate function code.
2. The SCSI class driver for the converter receives the I/O request, interprets the function code, constructs the appropriate SCSI command, and delivers the command to the VMS SCSI port driver.
3. The VMS SCSI port driver sets up the appropriate environment for a data transfer if the command warrants it, and delivers the command to the device on the SCSI bus. The port driver always acts as the initiator that sends the commands. The converter is always the target that interprets the commands.
4. The IEEE488 Bus-Converter extracts information from the SCSI commands and puts the commands (and data, if applicable) onto the IEEE488 bus.

5 The IEEE488 peripheral device receives the commands and data. Figure 4–2 illustrates the flow of commands (and data) from the user request, through the port interface to the SCSI bus, onto the IEEE488 Bus-Converter, and then to the IEEE488 device.

Figure 4–2 Data Flow from User to IEEE488 Device



6 The SCSI port driver returns status (and data, if applicable) to the class driver.

7 The class driver interprets status and returns status (and data, if applicable) to the user program that issued the I/O request.

Additional information about this product can be obtained from your Digital sales representative or from product documentation. See Appendix A.

ULTRIX SCSI Device Support

ULTRIX Workstations Software, Version 2.1 (UWSV2.1), supports all Digital SCSI-based systems and their associated Digital SCSI disks and tapes. The systems supported include: the VAXstation 3100, MicroVAX 3100, DECstation 2100, DECstation 3100, DECsystem 3100, and VAXstation 3520/3540.

The ULTRIX SCSI development group is committed to open interfaces, and believes that the Common Access Method (CAM) charter is meeting this goal. The CAM interface, when available, will allow:

- Third-party vendors to code to an open interface standard
- Applications to be portable across different platforms
- Customers to develop their drivers once, port to any vendor's operating environment, and thereby provide for quick turn-around in the market with minimal support effort for each new system platform being introduced.

The ULTRIX development group plans to pursue a program for open SCSI that will be based on the standard SCSI interface.

VAXELN SCSI Device Support

The VAXELN product, Version 4.1, provides SCSI disk and generic device support on the VAXstation 3100 and MicroVAX 3100 systems.

Two SCSI drivers implement this support:

- SCSI disk device driver
- SCSI message class driver

The supplied disk device driver supports the following SCSI devices on the VAXstation 3100 platform:

RZ22	52-Mbyte	3.5-inch	Winchester	half-height
RZ23	104-Mbyte	3.5-inch	Winchester	half-height
RZ55	332-Mbyte	5.25-inch	Winchester	full-height
RX23	1.4-Mbyte	3.5-inch	diskette drive	half-height
RRD40	577-Mbyte	4.7-inch	compact disk	half-height

The VAXELN product employs a class/port driver architecture to communicate with devices on the SCSI bus. This architecture provides a clean definition of responsibility between the class driver and the port driver. The class driver is responsible for formatting the commands, interpreting status, and managing user data. The port driver monitors and controls the SCSI bus phase changes and sends and receives SCSI path control messages.

Drivers written using this class/port structure allow the class driver to be hardware independent, while creating a standard interface to the port drivers. This standardized interface permits class drivers to be developed independent of the underlying port hardware and software.

VAXELN SCSI Third-Party Device Support Mechanisms

The VAXELN Version 4.1 product provides three mechanisms that allow a vendor-developed SCSI device to be attached to the VAXstation 3100 or MicroVAX 3100 system:

- Disk device class driver
- Message class driver
- User-written class driver

The implementor of support for a vendor-developed device can select the most appropriate method, based on the capabilities of the device, user needs, and available programming resources.

Disk Device Class Driver The design of the standard VAXELN disk device class driver allows it to control most disk device drives that conform to the ANSI SCSI-2 specification.

Note *The ANSI SCSI-2 specification allows some flexibility in certain implementation details and omits other details. As a result, implementations of the SCSI standard may differ from manufacturer to manufacturer and from device to device. Although the VAXELN product allows the use of devices that are not supplied by Digital with the standard VAXELN disk driver, the fact that a specific device may or may not operate correctly in this manner does not imply VAXELN support of the device. Digital cannot guarantee that a vendor-developed device that currently works with a VAXELN class driver will continue to work with subsequent releases of the VAXELN product.*

To ensure that vendor-developed devices work properly in a VAXELN environment, Digital encourages the use of an established and supported VAXELN interface, such as those described in the next sections.

Message Class Driver The VAXELN product provides a generic message class driver to enable vendor-developed devices to communicate on the SCSI bus. The application program uses supplied subroutines in order to communicate with the device on the SCSI bus.

User-Written Class Driver An application, using a user-written vendor-developed SCSI class driver can communicate with a device that is not supplied by Digital. The user-written class driver is coupled with the port driver supplied by the VAXELN product. The user-written class driver can be tailored for the specific characteristics of the drive, including error recovery.

A

Associated Documents

The manuals associated with Digital's implementation of the Small Computer System Interface (SCSI) are listed below.

- *American National Standard for Information Systems—Small Computer System Interface-2 (SCSI-2)*, X3T9.2/89-042 specification

  The SCSI-2 specification (Revision 10b) is a draft of a proposed standard. Copies of the draft document may be purchased from Global Engineering Documents, 2805 McGaw, Irvine, CA 92714, (800) 854-7179 or (714) 261-1455. Refer to document X3.131-198X.

- *American National Standard for Information Systems—Small Computer System Interface (SCSI)*, X3.131-1986 specification

  The document can be ordered from the American National Standards Institute, Inc., 1430 Broadway, New York, NY, 10018. This document is known as the SCSI-1 standard.

- *Small Computer System Interface: A Developer's Guide*, Order No. EK–SCSIS–SP

  Defines Digital Equipment Corporation's implementation of the ANSI SCSI-2 draft requirements. It defines the mechanical, electrical, and functional requirements for interconnecting Digital's small computers and intelligent peripheral devices. (Available on compact disc.)

  This document can be ordered from Digital Equipment Corporation, P.O. Box CS2008, Nashua, NH 03061. For an internal order, write Publishing and Circulation Services (P&CS), NRO3-1/W3, Digital Equipment Corporation, Northboro, MA 01532.

- *VMS Version 5.3 Small Computer System Interface (SCSI) Device Support Manual*, Order No. AA–PAJ2A–TE

  Describes the mechanisms the VMS operating system provides that allow a SCSI device that is not supplied by Digital to be attached to certain VAXstation and MicroVAX systems.

  This document can be ordered from Digital Equipment Corporation, P.O. Box CS2008, Nashua, NH 03061. For an internal order, write the Software Supply Business (SSB), Digital Equipment Corporation, Westminster, MA 01473.

- *VMS Device Support Manual*

  Describes the components of a VMS device driver and the basic rules to which device drivers not supplied by Digital must adhere.

- *VMS I/O User's Reference Manual*

  Contains information necessary to issue I/O requests from an application program to I/O device drivers supplied as part of the VMS operating system.

- *VMS System Dump Analyzer Utility Manual*

  Provides assistance in investigating system failures.

- *VMS Delta/XDelta Utility Manual*

  Provides information on debugging VMS device driver code.

- *VAX MACRO and Instruction Set Reference Manual*

  Describes the VAX MACRO assembly language. VMS device drivers are written in VAX MACRO.

- *VMS System Services Reference Manual*

  Describes the high-level language interface to the I/O subsystem of the VMS operating system.

- *IEZ11 Software Installation Guide*, Order No. AA–PA9DA–TE

- *IEZ11 User's Guide*, Order No. AA–NU45A–TE

- *IEZ11 Hardware Installation Guide*, Order No. AA–PA9EA–TE

For additional information, refer to:

NCR Corporation. *SCSI: Understanding Small Computers*. Englewood Cliffs, N.J.: Prentice-Hall, 1990.

Glossary

ANSI

American National Standards Institute.

arbitrate

To select one device from several devices that are seeking access to the SCSI bus concurrently.

assertion

The process of driving a signal low, to the true state (0 V–0.5 V).

asynchronous event notification (AEN)

A SCSI protocol that allows a SCSI device (usually a target) to inform the processor (usually the initiator) that an event has occurred asynchronously with respect to the processor's current stream of execution.

bus

A flat or twisted-pair cable composed of individual identical circuits. A computer's SCSI port and SCSI peripheral devices can be installed anywhere along the cable.

bus free

No SCSI device is actively using the bus; therefore, the bus is available for use.

byte

A binary character string made up of eight bits.

central processing unit (CPU)

The part of the computer system that controls the interpretation and execution of instructions.

CISC

Complex Instruction Set Computer.

class driver

A component of the SCSI class/port architecture that acts as an interface between the user and the SCSI port. In the VMS implementation, the SCSI class driver translates I/O functions as specified in a user's $QIO request to a SCSI command targeted to a device on the SCSI bus.

Common Command Set (CCS)

A subset of eighteen commands in the SCSI command set. CCS is a proposal before the ANSI X3T9.2 subcommittee.

connect

The function that occurs when an initiator selects a target to start an operation. A connection can only occur between an initiator and a target.

controller

A computer module that interprets signals between the host and a peripheral device.

CPU

See **central processing unit**.

device

The general name for any unit connected to the system that is capable of receiving, storing, or transmitting data. *See also* **controller**.

device driver

A set of routines and tables that the system uses to process an I/O request for a particular device type.

disconnect

The action that occurs when a SCSI device releases control of the SCSI bus, allowing it to go to the Bus Free phase.

DMA

Direct Memory Access.

host adapter

A device that connects a host system to the SCSI bus. The device usually performs the lower layers of the SCSI protocol and normally operates as an initiator. This function may be integrated into the host system.

initiator

A SCSI device (usually a host system) that requests an operation to be performed by another SCSI device (a target).

I/O process

A process consisting of one initial connection and zero or more reconnections, all pertaining to a single command.

interrupt priority level (IPL)

The level at which a software or hardware interrupt is generated on a VAX system. There are 32 interrupt priority levels: IPL 0 is the lowest, 31 is the highest. The levels arbitrate contention for processor service. For example, a device cannot interrupt a processor if the processor is currently executing at an IPL greater than the IPL of the device's interrupt request.

OEM

Original Equipment Manufacturer.

open system

A vendor-neutral computing environment that is compliant with international standards, permits system and network interoperability or software applications portability, including consistency of data and human access, and satisfies one or more of a business's functional requirements.

peripheral device

A physical device that can be attached to a SCSI device, which in turn connects to the SCSI bus. It provides the CPU with additional memory storage or communications capability. Examples of peripheral devices are: magnetic disks, printers, optical disks, and magnetic tapes.

port driver

A component of the VMS SCSI class/port architecture that transmits and receives SCSI commands and data. It knows the details of transmitting data from the local processor's SCSI port hardware across the SCSI bus.

priority

The ranking of devices on the bus during arbitration.

processor

A functional section of hardware in a computer that changes instructions into a form the computer understands, and executes these instructions.

reconnect

The act of reviving a nexus to continue an I/O process. A target reconnects to an initiator by using the Reselection and Message In phases after winning arbitration. An initiator reconnects to a target by using the Selection and Message Out phases after winning arbitration.

RISC

Reduced Instruction Set Computer.

SCSI

An interface designed for connecting peripheral devices to computer systems. SCSI is defined by an American National Standards Institute (ANSI) standard; in this guide, SCSI refers to either SCSI-1 or SCSI-2.

SCSI-1

ANSI Small Computer System Interface (X3.131–1986).

SCSI-2

Proposed ANSI Small Computer System Interface-2 (X3.131–198x).

SCSI device

A host computer adapter, a peripheral controller, or intelligent peripheral that can be attached to the SCSI bus.

SCSI device ID

The bit-significant representation of the SCSI address referring to one of the signal lines DB(7–0).

VMS systems: Typically, both the MicroVAX 3100 and VAXstation 3100 systems are assigned device ID 6 and assert at DB(6); the VAXstation 3520/3540 CPU is assigned device ID 7 and asserts at DB(7).

ULTRIX systems: The VAXstation 3100, VAXstation 3520/3540, DECstation 2100 and 3100 systems, as well as the DECsystem 3100, are assigned device ID 6, and the ULTRIX kernel asserts at DB(7). ULTRIX customers can reconfigure the SCSI ID to their needs.

SCSI port

Software: the channel that controls communications to and from a specific SCSI bus in the system. Hardware: The name of the logical socket at the back of the system unit to which a SCSI device is connected.

SII

System Interface Interconnect, a SCSI controller chip.

status

One byte of information sent from a target to an initiator upon completion of each command.

System Interface Interconnect

A gate array developed by Digital that manages the SCSI bus and the following activities: SCSI bus selection, bus to memory transfers, and DISCONNECT and RECONNECT commands.

target

A SCSI device that performs an operation requested by an initiator.

Index

A

ANSI
  acronym for, 1–3
  X3T9 committee, 1–3

B

Bus phases
  Arbitration, 3–4
  Bus Free, 3–4
  Command, 3–4
  Data In, 3–4
  Message In, 3–4
  Message Out, 3–4
  operation of, 3–3
  Selection, 3–3
  Status, 3–3
Bus signals
  control, 3–5
  data, 3–5
  driven by initiator and target, 3–5
  table of, 3–6

C

Class driver
  VMS, 4–2
Common Command Set
  description of, 3–7
Customer Support Services
  desktop program, 2–7
  vendor application service, 2–7
  vendor equipment services, 2–6

D

Development considerations, vii, 2–1
Device
  definition of, 3–1
Device controller
  description of, 3–7
Device ID
  assignments, 3–1
Device support, 4–1
  See also VMS
Digital
  member of X3T9 committee of ANSI, 1–3
Documentation
  materials for developers, v
  where to order, A–1
Driver
  evaluating SCSI, 2–1
  experience needed, 2–1
  requirements for writing, 2–1
Driver, writing
  documentation needed, 2–1

E

Evaluating SCSI
  development considerations, 2–1

G

Generic SCSI class driver
  compared to SCSI third-party class driver, 4–4

H

Host adapter
  description of, 2–3

I

IEEE488 bus
  planning software support, 4–5
  to SCSI bus, 4–5
Initiator
  host adapter, 3–1

O

Open systems
  Digital's commitment to, 1–3
Operating system support
  ULTRIX, 4–1, 4–8
  VAXELN, 4–1, 4–9
  VMS, 4–1

P

Peripherals
  disk drives, 3–1
  medical devices, 3–1
  modems, 3–1
  optical devices, 3–1
  printers, 3–1
  scanners, 3–1
  test equipment, 3–1
Port driver
  VMS, 4–1

S

SCSI
  acronym for, 1–3
  benefits of, 1–2
  development considerations, 2–1
  Digital's strategy, 1–2
  Digital's systems with, 2–3
  features implemented by Digital, 2–4
  I/O bus, 3–1
  single-ended configuration, 3–2
SCSI bus
  configuration, 3–1
  figure of, 3–2
  peer-to-peer relationship, 3–1
  to IEEE488 bus, 4–5
SCSI chips
  Digital SII gate array, 2–4
  NCR 5380 chip, 2–3
SCSI connectivity
  DECstation 2100, 4–1
  DECstation 3100, 4–1
  DECsystem 3100, 4–1
  MicroVAX 3100, 4–1
  VAXstation 3100, 4–1
  VAXstation 3520/3540, 4–1
Service
  Digital policy, 2–6
  third-party support, 2–6
Standards, vii
  Digital's commitment to, vii, 1–3
System Interface Interconnect (SII), 2–4

T

Target
  device controller, 3–1
Third-party SCSI class driver
  compared to SCSI generic class driver, 4–4
Third-party SCSI device
  VMS support mechanisms, 4–3 to 4–5
Third-party support
  ULTRIX, 4–8

Third-party support (Cont.)
VAXELN, 4–9
VMS, 4–1

V

VAXELN
class/port driver architecture, 4–9
disk device class driver, 4–10
message class driver, 4–10
third-party device support mechanisms, 4–9
user-written class driver, 4–10
VMS
class/port driver architecture, 4–1
class driver, 4–2
device support mechanisms, 4–3
port driver, 4–1
third-party support, 4–1