EK–KDB50–UG–PRE

KDB50 Disk Controller User Guide

digital

EK-KDB50-UG-PRE

PRELIMINARY EDITION
KDB50
DISK CONTROLLER
USER GUIDE

Digital Equipment Corporation
Colorado Springs, Colorado

Preliminary Edition, July 1985

Copyright c 1985
by Digital Equipment Corporation
Printed in U.S.A.
All Rights Reserved

The material in this manual is for informational purposes and is subject to change without notice. DIGITAL assumes no responsibility for any errors which may appear in this manual.

Digital Equipment Corporation does not grant licenses to make, use or sell equipment as described in this manual, and makes no claim that use of its products with those of other manufacturers will not infringe on existing or future patent rights.

This equipment generates, uses, and may emit radio frequency energy. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC rules for operation in a commercial environment. If this equipment is operated in a residential area, the user, at his own expense, may be required to take corrective measures.

The following are trademarks of Digital Equipment Corporation, Maynard, Massachusetts:

DEC	DECnet	RQDX
DECUS	DECsystem-10	RC25
HSC50	DECSYSTEM-20	VT
digital	DECwriter	KDB50
PDP	RA80	RSTS
UNIBUS	RSX	UDA50
VAX	MASSBUS	VMS
RA81	RA60	KDA50-Q

CONTENTS

FIGURES

TABLES

PREFACE

This guide describes how to install, checkout, and operate the KDB5Ø disk controller. It is intended for the equipment user and includes detailed equipment specifications in the first chapter.

This guide does not cover maintenance and servicing. Information on these subjects is provided separately in the KDB5Ø Disk Controller Service Manual. For further information, refer to the Related Documentation section included in Chapter 1.

CHAPTER 1
GENERAL INFORMATION

1.1 KDB5Ø DISK CONTROLLER INTRODUCTION

The KDB5Ø is part of the Digital Storage Architecture (DSA) family of intelligent disk controllers. It interfaces any combination of up to four DSA disk drives to a host CPU operating on a VAXBI bus. Since the KDB5Ø is part of the DSA family, the KDB5Ø:

1. Handles most of the I/O management traditionally performed by the host.

2. Allows the host to view the disk subsystem as one contiguous string of error-free sectors known as logical blocks.

3. Takes over the traditional host concern of disk geometry (cylinder, track, and block).

4. Communicates with the host via the host bus (VAXBI) and Mass Storage Control Protocol (MSCP).

5. Communicates with the disk drives via the SDI bus and SDI protocol.

1.1.1 KDB5Ø SUBSYSTEM

Figure 1-1 shows the basic KDB5Ø subsystem configuration. The KDB5Ø communicates with the disk drives using the SDI bus and SDI protocol. It communicates with the host using the VAXBI bus and MSCP.

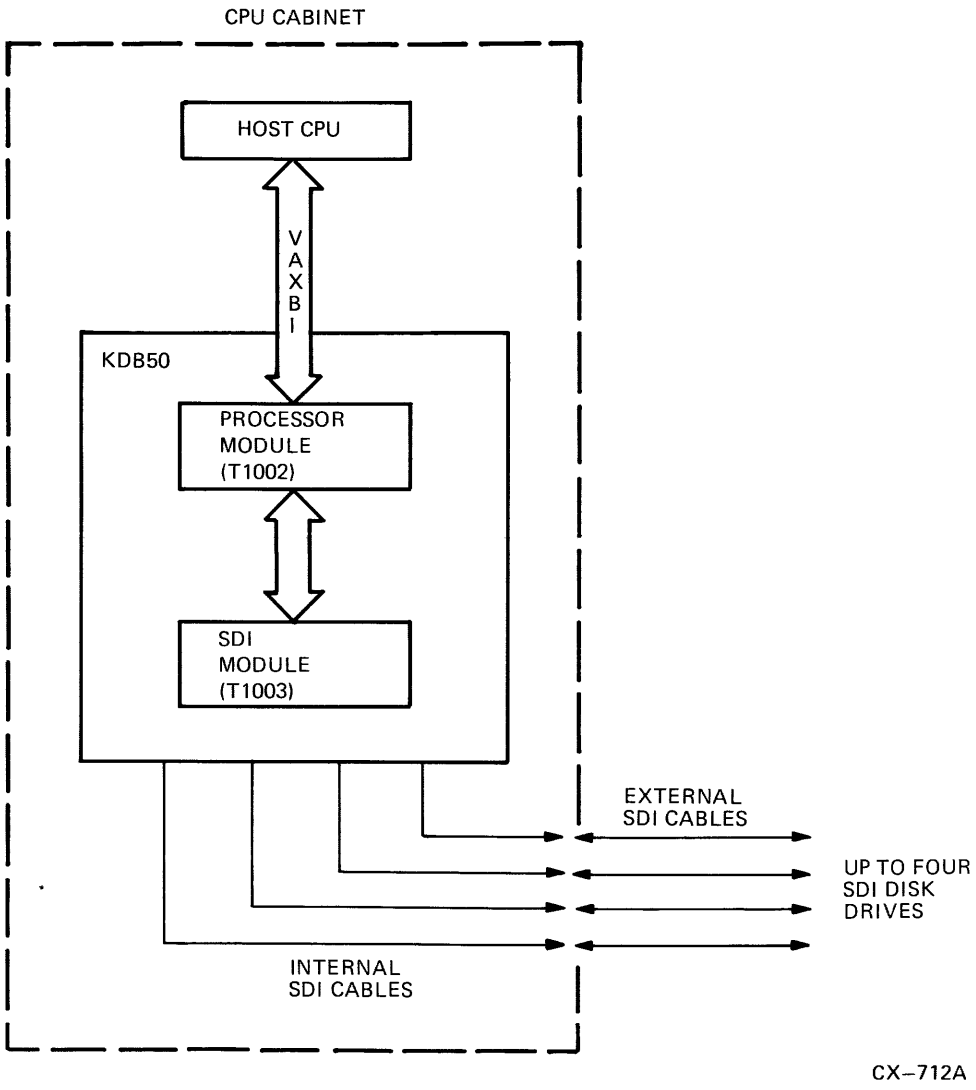


Figure 1-1 KDB5Ø Subsystem Configuration

1.1.2 MASS STORAGE CONTROL PROTOCOL

MSCP is a set of rules that hosts and intelligent controllers use to communicate. MSCP hides device-dependent requirements, such as disk geometry and error recovery strategies, from the host. Because these requirements are invisible to the host, one class driver can replace several different device drivers. For example, instead of supplying a device driver for each type of disk drive in a subsystem, MSCP supplies a disk class driver that is able to communicate with all the possible SDI disk drives.

The following description further illustrates the function of MSCP:

> To request an I/O operation, the host constructs an MSCP message and sends it to the controller. The MSCP message contains the drive address, the function to be performed, the starting logical block number, and the amount of data requested.
>
> When the controller receives the request, it independently performs all drive management and data movement, as well as any necessary error recovery. Upon command completion, the subsystem gives status information by sending the host an MSCP response message.

1.1.3 SDI Bus Interface

The SDI bus is the connection between the KDB5Ø and the disk drive(s). As a radial bus, it supplies a separate internal SDI cable to the CPU cabinet I/O bulkhead for each disk drive (up to four). The KDB5Ø is electrically attached to these cable(s) by a separate port and services each of these ports independently and with equal priority. In short, all communication between the KDB5Ø and the disk drive(s) follows the SDI communication protocol. The SDI bus and protocol:

- Provide real-time sector and index pulses to the controller from the drive.
- Provide error detection when:
  1. For data transfers (other than formatting operations), the controller verifies the correct read/write head has reached the correct disk address before attempting the transfer.
  2. For data writes to the disk, the controller generates an Error Correction Code (ECC) and an Error Detection Code (EDC). It then appends these codes to the data. During data reads from the disk, the controller checks the EDC and ECC for consistency.
  3. For all control and data transmissions, the receiving device constantly monitors the SDI transmission for format errors.

1.1.4 VAXBI Bus Interface

The VAXBI bus is a synchronous and interlocked connection between the KDB5Ø and the host system (Processor, memory, and I/O). The VAXBI is synchronous because of its two system clocks (distributed to all VAXBI nodes) which provide synchronous operation for all transactions. The VAXBI is interlocked because one transaction must complete before the next transaction may start. In addition, the VAXBI features:

- 13.3 Mbytes/second bus bandwidth maximum.
- Bus arbitration, address, and data transmissions that are time multiplexed over 32 data lines.
- High degree of data integrity.
- Bus error detection and reporting for all VAXBI nodes.
- Distributed arbitration--no central arbitrator required.
- Slot interchangeability.
- Standardized interface for all designs.
- One gigabyte address capability.
- Up to 16 full master-slave-interrupt type nodes.
- Worst-case design analysis.
- Powerup and requested self-test on all nodes.

1.1.4.1 VAXBI System Configurations - The VAXBI system may be configured in many different ways, depending on customer needs. Figures 1-2 and 1-3 show the KDB5Ø in two possible configurations.

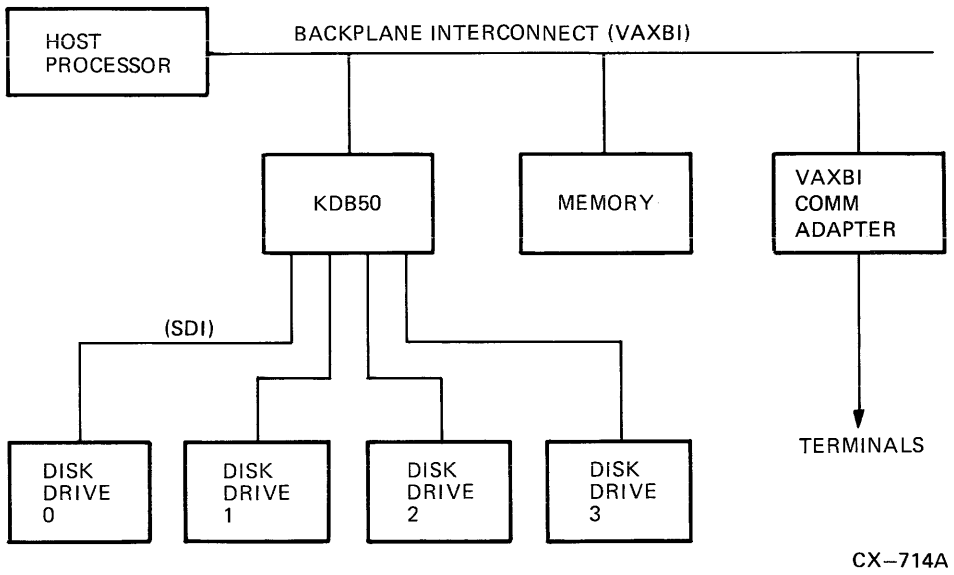


Figure 1-2 Configuration A

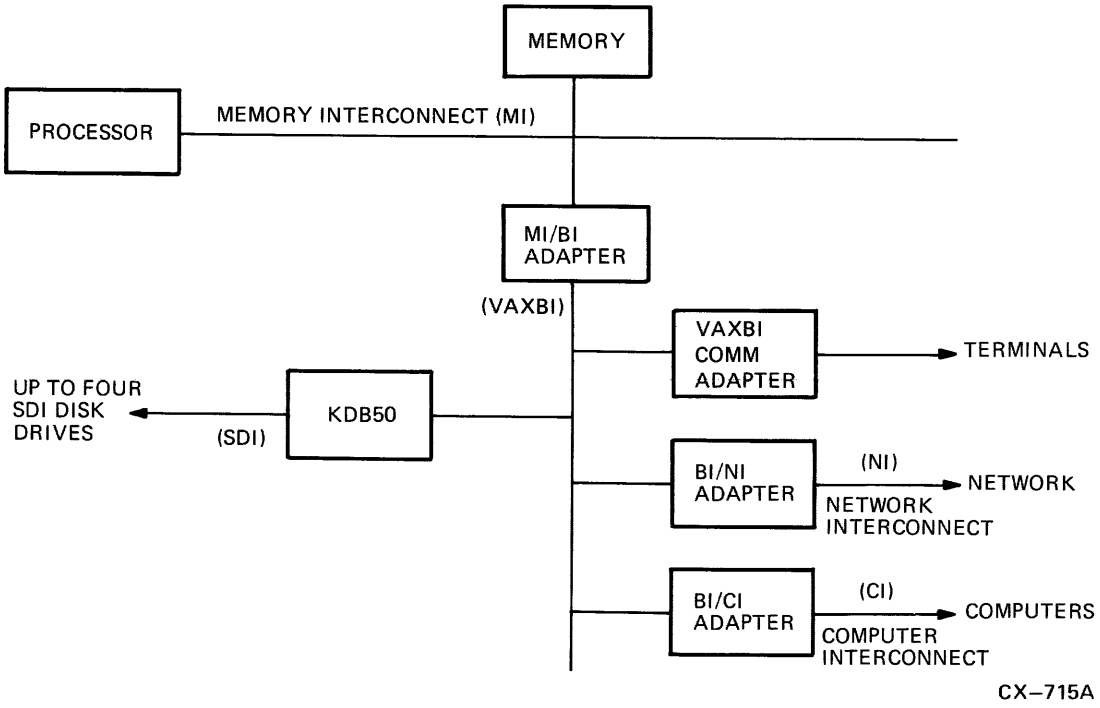


Figure 1-3 Configuration B

1.2 KDB5Ø BI MODULES

Two BI modules make up the KDB5Ø: the Standard Disk Interface (SDI) module (T1ØØ3) and the Processor module (T1ØØ2). These BI modules are modules that are sized according to European standards. Figure 1-4 shows the BI module physical dimensions.

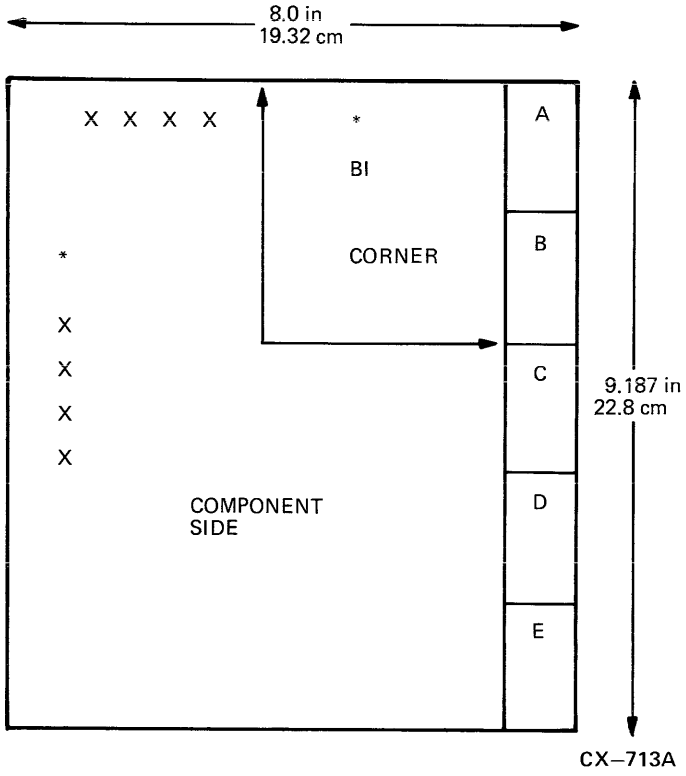


Figure 1-4 BI Module Physical Dimensions

NOTE

In Figure 1-4, x = the location of the red diagnostic LED indicators and * = the location of the required VAXBI yellow LED indicators.

Using circuitry on these two modules, as well as microprogrammed control, the KDB5Ø interfaces with the VAXBI bus and the SDI bus. The following sections describe the hardware on both the SDI and Processor modules.

1.2.1 SDI Module

The Standard Disk Interface (SDI) module (T1ØØ3) is the communication interface between the KDB5Ø Processor module and the disk drives. Some of the characteristics of the SDI module are that it:

- Contains a 16k-word high-speed buffer (RAM data buffer).
- Generates parity for the RAM data buffer.

- Detects RAM data buffer parity errors.
- Converts data between the parallel format of the KDB50 internal bus (DBUS <15:00>) and the serial format of the SDI bus.
- Generates the Reed-Solomon Error Correction Code (ECC) for disk write data.
- Checks the ECC for disk read data.
- Provides the electrical interface to the SDI bus.
- Detects pulse errors on the SDI bus.

1.2.1.1 **RAM Data Buffer** - The 16k-word high-speed RAM data buffer increases system performance by supplying temporary data storage during data transfers between the disk drive, the KDB50, and the host. Because this buffer can store up to 41 blocks of data, the possibility of a buffer-full condition is reduced. This, in turn, lessens the possibility that data will be missed due to insufficient temporary storage. (When data is missed, the disk must make another full revolution to retrieve it. This extra revolution decreases system performance.) By reducing the possibility of missed data due to a buffer-full condition, the 16k-word high-speed buffer increases system performance.

1.2.1.2 **RAM Parity Generation And Parity Error Detection** - The SDI module contains circuitry that generates and checks parity for the RAM data buffer. This ensures that the parity of the data read from the RAM buffer has not changed since it was written. In turn, this ensures that single bit and massive data errors are detected, thus providing data integrity.

1.2.1.3 **Data Conversion** - Data is converted between DBUS <15:00> and the SDI bus in the following two stages:

1. Between parallel format and serial Non-Return-to-Zero (NRZ) format.
2. Between serial NRZ format and serial SDI format.

Both the parallel format and the serial NRZ format use TTL (+5.Ø volt and Ø.Ø volt) levels to represent data. However, parallel format transfers data 16 bits at a time, while serial format transfers data one bit at a time.

The serial SDI format is also designed so that any NRZ data pattern can be represented as alternating positive and negative pulses. Because any data on the SDI must consist of alternating pulses, the receiving device (disk drive on one end, KDB5Ø on the other) detects transmission errors when it observes two adjacent pulses with the same polarity.

Figure 1-5 shows the various stages of data conversion between the SDI bus and DBUS <15:ØØ>. Notice that DBUS <15:ØØ> is common to both the KDB5Ø SDI module and the KDB5Ø Processor module. For simplicity, Figure 1-5 shows only one port to the SDI bus. However, there may be up to four ports.

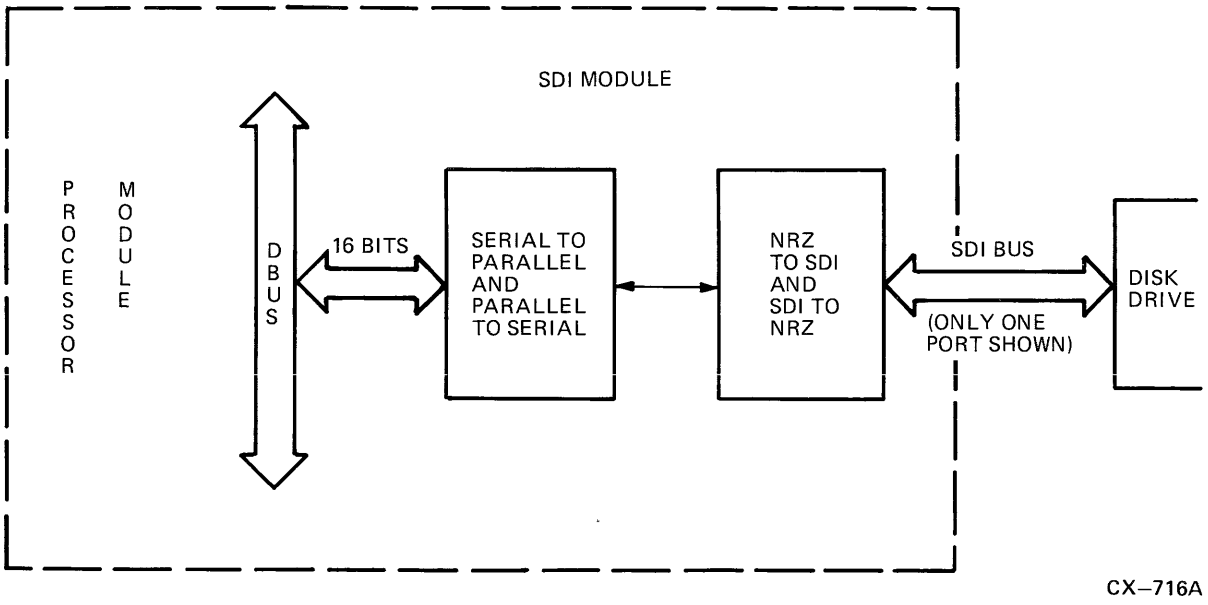


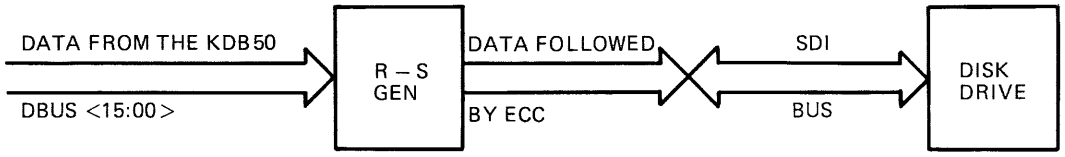
Figure 1-5 SDI To DBUS <15:ØØ> Datapath

1.2.1.4 Reed-Solomon Error Correction Code (ECC) - Error correction codes are used to correct errors in data. By performing certain mathematical calculations, they make it possible to correct portions of a data block that have been altered. (Altered data occurs when conditions such as media problems or circuit problems are experienced.) The ECC consists of 17 ten-bit symbols representing the 512 data and Error

Detection Code (EDC) word bytes of information the block contains.

The ECC code is generated and checked by a custom DIGITAL component known as the R-S GEN. This is done for each sector (block) of data sent and received from the disk drive. For example, before data goes to the disk drive, it passes through the R-S GEN which generates a unique ECC for that block of data and stores this ECC data with the block of data. When the data is subsequently read from the disk, it again passes though the R-S GEN. However, this time, the R-S GEN recalculates the ECC and compares it to the ECC read from the disk. The R-S GEN then detects an error if the recalculated ECC does not compare with the actual ECC read from the disk. (Figure 1-6 illustrates ECC generation and Figure 1-7 illustrates ECC checking.)

If the R-S GEN detects an ECC error, it notifies the KDB50 Processor of the error. Using a microcode algorithm, the Processor module corrects the data, if possible, before sending it to the host. If the error cannot be corrected, the data is not sent to the host, and the KDB50 notifies the host of an uncorrectable ECC error. The correction capability of this algorithm is up to one 180-bit burst per block.



THE R–S GEN GENERATES AN ECC FOR THE DATA WHICH IS GOING TO THE DISK DRIVE.

CX–717A

Figure 1-6 ECC Generation

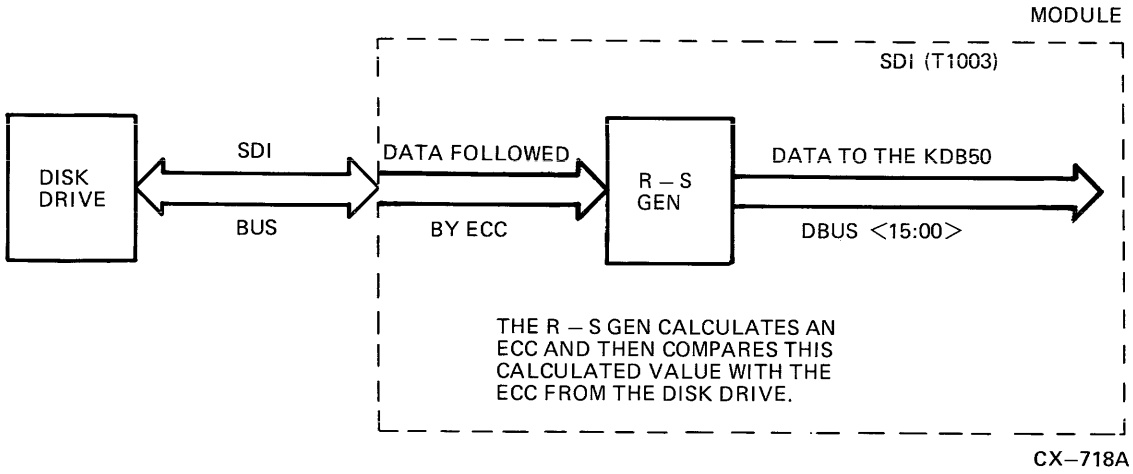


Figure 1-7 ECC Checking

1.2.2 Processor Module

The Processor module (T1ØØ2) is the control portion of the KDB5Ø. It:

- Performs all KDB5Ø interaction with the VAXBI bus via two custom DIGITAL chips: the Backplane Interconnect Interface Chip (BIIC) and the VAXBI Chip Interface Adapter Interface (BCAI).
- Contains a dual microprocessor consisting of two 12-bit sequencers and a shared 16-bit ALU.
- Contains the Control Store Read Only Memory (CROM) for the microprocessor.
- Provides parity checking for the CROM.
- Preserves parity on data from RAM to the VAXBI bus and from the VAXBI bus back to RAM.

1.2.2.1 BIIC And BCAI - The following section describes the interface between the VAXBI and DBUS <15:ØØ>. The acronyms used are:

- BIIC -- Backplane Interconnect Interface Chip.
- BCI -- Backplane Interconnect Chip Interface.
- BCAI -- Backplane Interconnect Chip Interface Adapter Interface.

The BIIC and the BCAI provide an interface between the VAXBI bus, the BCI bus, and the KDB50 DBUS and control lines. The BIIC provides the interface between the VAXBI bus and the BCI bus, while the BCAI provides the interface between the BCI bus, the KDB50 DBUS and control lines. Figure 1-8 shows the BIIC and BCAI configuration on the KDB50 Processor module.

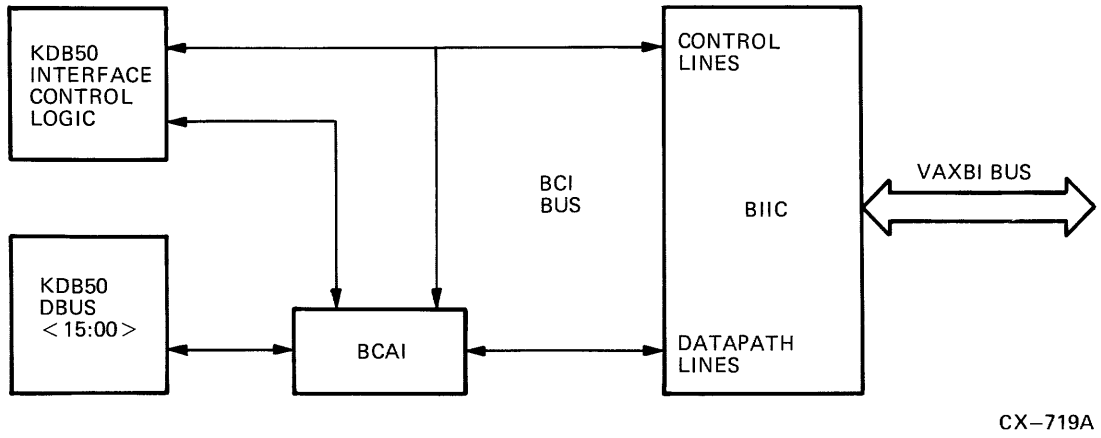


Figure 1-8 BIIC And BCAI Configuration On The KDB50 Processor Module

1.2.2.1.1 **BIIC Features** - The following list describes some of the BIIC features. The BIIC:

- Handles all aspects of the VAXBI arbitration protocol.
- Implements all VAXBI command types.
- Provides a flexible and convenient interrupt system.
- Provides address decoding and matching.
- Provides the five required VAXBI registers, two BIIC control registers, and three KDB50-specific registers.
- Provides all connections for the VAXBI bus (except two clock signals and three control signals).
- Provides all the necessary signals for the BCI bus at TTL levels.
- Allows the KDB50 to read or write BIIC registers via loopback mode which does not use the VAXBI bus datapath.
- Provides extensive error checking (examples are data parity, data compare, protocol).

- Performs a self-test on powerup and on request.
- Permits node retries on most transient VAXBI errors.
- Operates on a single +5.Ø volt source.
- Does not affect the VAXBI bus when the BIIC is powered down.

1.2.2.1.2 BCAI Features - The BCAI is a DIGITAL custom IC containing a dual-ported register file which acts as a buffer file between the BIIC and the KDB5Ø's internal data bus. In addition, the BCAI contains a DMA Master Port through which the KDB5Ø implements the BCAI. The DMA Master Port consists of two octal-word DMA data buffers, a command/address register with increment capability, and a next-page frame register. (The DMA data buffers are protected via byte parity.)

1.2.2.2 Dual Microprocessor - The dual 16-bit microprocessor consists of several bit-slice components. By itself, each bit-slice component has a limited function and a narrow data path (4-bits). However, when connected in parallel, they create two 12-bit sequencers and a 16-bit ALU. Each sequencer provides a program counter, stack, and stack pointer, while the ALU provides the arithmetic and logic capabilities.

The two sequencers, known as the BI Processor (BPROC) and the Drive Processor (DPROC), operate alternately and execute individual microprograms from CROM (refer to Section 1.2.2.3). For example, while the BPROC is using the ALU to perform an operation, the DPROC is fetching its next instruction from CROM. Similarly, while the DPROC is using the ALU to perform an operation, the BPROC is fetching its next instruction from CROM. Figure 1-9 shows a simplified block diagram of this dual microprocessor scheme.

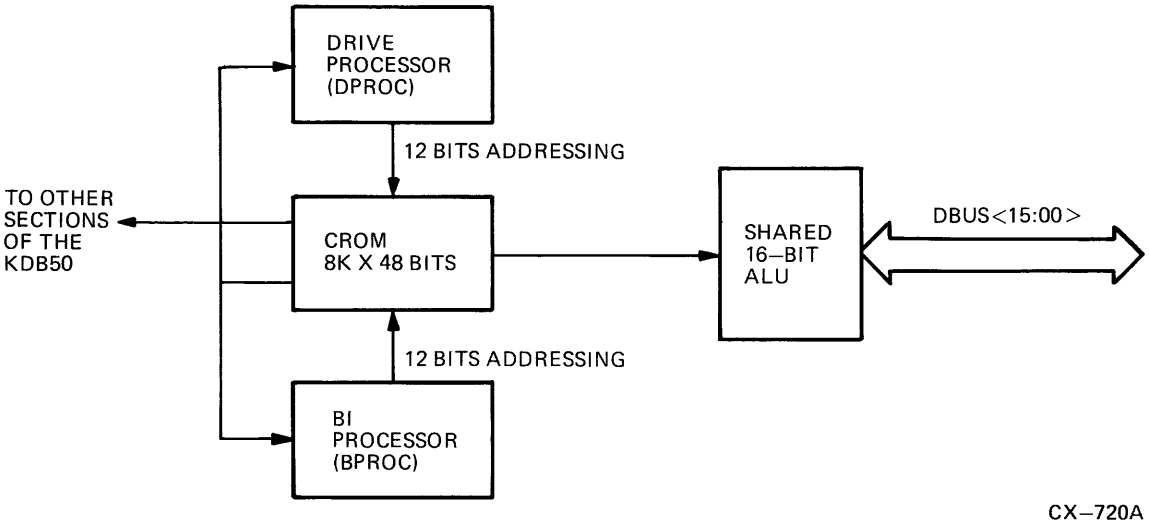


Figure 1-9 KDB5Ø Dual Microprocessor Scheme

1.2.2.3 Control Store Read Only Memory (CROM) - The CROM is an 8k x 48-bit Read Only Memory (ROM) containing microcode for both the BPROC and the DPROC. One half of the CROM is dedicated to the BPROC, and the other half is dedicated to the DPROC.

A CROM parity circuit constantly monitors the CROM data for odd parity. If parity is not odd, the parity circuit assumes a corrupted microword has been accessed and halts all KDB5Ø operation. The KDB5Ø then flags the host of an error and stays in a wait state until re-initialized by the host. Before initialization can complete, the KDB5Ø examines the entire contents on the CROM for parity errors.

1.3 KDB5Ø FUNCTIONAL MICROCODE

The KDB5Ø microcode is divided into two functional streams: The BI control stream which manages the controller to host interface, and the drive control stream which manages the controller to disk drive interface. Figure 1-1Ø illustrates the basic functional division of the two control streams in a system.

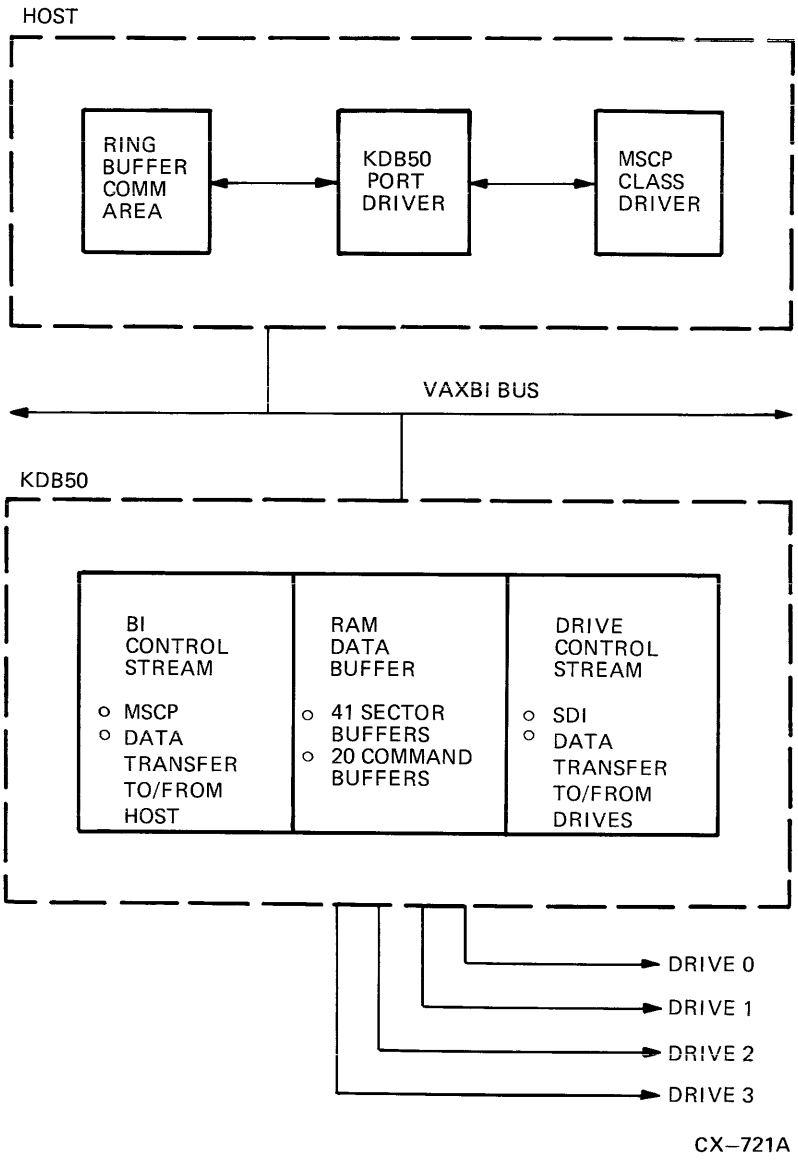


Figure 1-10 VAXBI And SDI Control Stream Functional Division

1.3.1 BI Control Stream

The BI control stream:

- Handles the transfer of commands from the host and responses to the host.
- Exchanges information packets with the host in memory.
- Validates each packet from the host.
- Executes MSCP immediate commands.

- Generates the KDB5Ø response packets for transmission to the host.
- Analyzes the drive packets and performs the following functions:
  - Decodes the logical block number (LBN) to cylinder, group, track, and sector (block) information.
  - Optimizes seek selection from the outstanding commands.
  - Allocates data buffer space.
  - Computes and stores parameters for each block transfer.
  - Performs packet error detection.
  - Performs memory mapping for mapped requests.
  - Transfers data between the host and the KDB5Ø internal RAM data buffer.
- Performs ECC error correction.
- Polls the command queue at the completion of each command.
- Performs initialization.
- Initiates drive control stream packet executions.
- Monitors host activity.
- Performs attention and log message transmission to the host.
- Performs the appropriate error handler for VAXBI bus errors.

1.3.2 Drive Control Stream

The drive control stream:

- Monitors ATTENTION from the drives. When drive attention has been detected, the drive control stream gets the drive status, compares it with the previous status, and takes appropriate action.
- Constructs and sends packets to the disk drives. The packets may be the result of a host request or in response to a drive-attention condition.
- Receives and validates packets from the drives.
- Monitors the drive status flags from the BI control stream. The drive status flags are used for communication between the BI control stream and the drive control stream.
- Performs the following tasks as required by the drive status flags:
  - Initiates read, write, seek, and head select packets to the drive.
  - Reads and verifies the block header.
  - Performs data transfers between the internal RAM and the disk drive.
  - Updates drive status and buffer-use flags.
  - Performs bad block revectoring.

1.4 KDB5Ø SPECIFICATIONS

The KDB5Ø Disk Controller Specifications are described in Table 1-1.

Table 1-1 KDB5Ø Specifications

Characteristics	Specifications
Physical components	KDB5Ø Processor module (T1ØØ2) KDB5Ø SDI module (T1ØØ3) KDB5Ø backplane I/O assembly (7Ø-22492-Ø5 (8 ft.), 7Ø-22492-Ø6 (12 ft.)) KDB5Ø I/O bulkhead (74-31369-Ø1)
Power consumption	73.6 watts nominal
Heat dissipation	Approximately 268.8 BTU/hour
Electrical voltage and current requirements	1Ø.4 amps at +5.Ø volts, 2Ø milliamps at +12.Ø volts, 3ØØ milliamps at -2.Ø volts, 3.1 amps at -5.2 volts
Operating temperature range	5 degrees C to 5Ø degrees C (41 degrees F to 122 degrees F), with a temperature gradient of 2Ø degrees C/hour (36 degrees F/hour)
Operating relative humidity range	1Ø% or less to 95%, with a maximum wet bulb temperature of 32 degrees C (9Ø degrees F) and a minimum dew point of 2 degrees C (36 degrees F)
Operating altitude range	Sea level to 24ØØ meters (8ØØØ ft). Reduce the maximum allowable operating temperature by 1.8 degrees C/1ØØØ meters (1 degree F/1ØØØ feet) for operation above sea level
Mounting restrictions	Mounts in two BI module slots in BI cage box H94ØØ-AA

1.5 DIGITAL CUSTOMER SERVICE CONTRACT OPTIONS

You can upgrade your CPU system smoothly and efficiently and maintain optimum performance by taking advantage of one of the following service options.

1.5.1 Hardware Services

Add-on and upgrade services get you started. We strongly recommend that your new CPU upgrade be installed by qualified DIGITAL field service technicians. Installation includes:

1. Pre-installation evaluation to ensure a suitable site environment, including power, temperature and humidity.

2. Physical connection of equipment and verification of full system functionality.

Maintenance services keep you going. Once your upgrade has been installed by DIGITAL field service, the entire upgraded system will be eligible for coverage by one of the following DIGITAL comprehensive service contracts:

1. DECservice - A comprehensive on-site service providing a program of preventive maintenance, committed response times, continuous effort until the problem is solved, installation of the latest engineering changes, and automatic escalation for complex problems.

2. Basic Service - An economical full-service coverage providing priority status second only to DECservice calls, as well as the preventive maintenance on-site services described for DECservice.

For less comprehensive but equally reliable service, you can choose Per Call Service, Carry-In Service or DECmailer Service.

1.5.2 Software Services

If your need is to analyze or upgrade your current system, or develop or implement software, DIGITAL offers a service to meet your needs. The following services will be of particular interest to you as you plan to add-on or upgrade:

1. Computer Performance Service - Helps you develop growth plans by identifying add-on or upgrade options before problems begin.

2. System Start-up Service Packages - Provides fixed-cost training for your staff, as well as one year of support services.

3. Consulting Services - Provides software programming or project manager expertise on a resident, per-call, or fixed-price basis. Your choice.

Any DIGITAL service option you select will provide you with high quality, reliable service from one of the largest service organizations in the industry. For more information, call your local DIGITAL field service office.

1.6 RELATED DOCUMENTATION

DIGITAL customers may order the following KDB50 related manuals:

- KDB50 DISK CONTROLLER USER GUIDE (EK-KDB50-UG)
- KDB50 DISK CONTROLLER SERVICE MANUAL (EK-KDB50-SV)

External DIGITAL Customers:	The User Guide and Service Manual may be ordered directly from Digital Equipment Corporation, P.O. Box CS2008, Nashua, New Hampshire 03061, or by calling toll free: 800-258-1710.
Internal DIGITAL Customers:	The User Guide and Service Manual may be ordered directly from Publication and Circulation Services, 10 Forbes Road, Northboro, Massachusetts 01532 (RCS Code: NR12, Mail Code: NR03/W3).

Outside the United States, consult local DIGITAL offices.

CHAPTER 2
INSTALLATION

2.1 KDB5Ø INSTALLATION

A certain amount of planning and preparation is necessary before installing the KDB5Ø. The following paragraphs outline some of the items that should be considered.

In addition to the required BI-based system +12.Ø and +5.Ø voltages, ensure that the BI cage also has -2.Ø and -5.2 available voltages. These voltages are required to power the ECL logic on the SDI module.

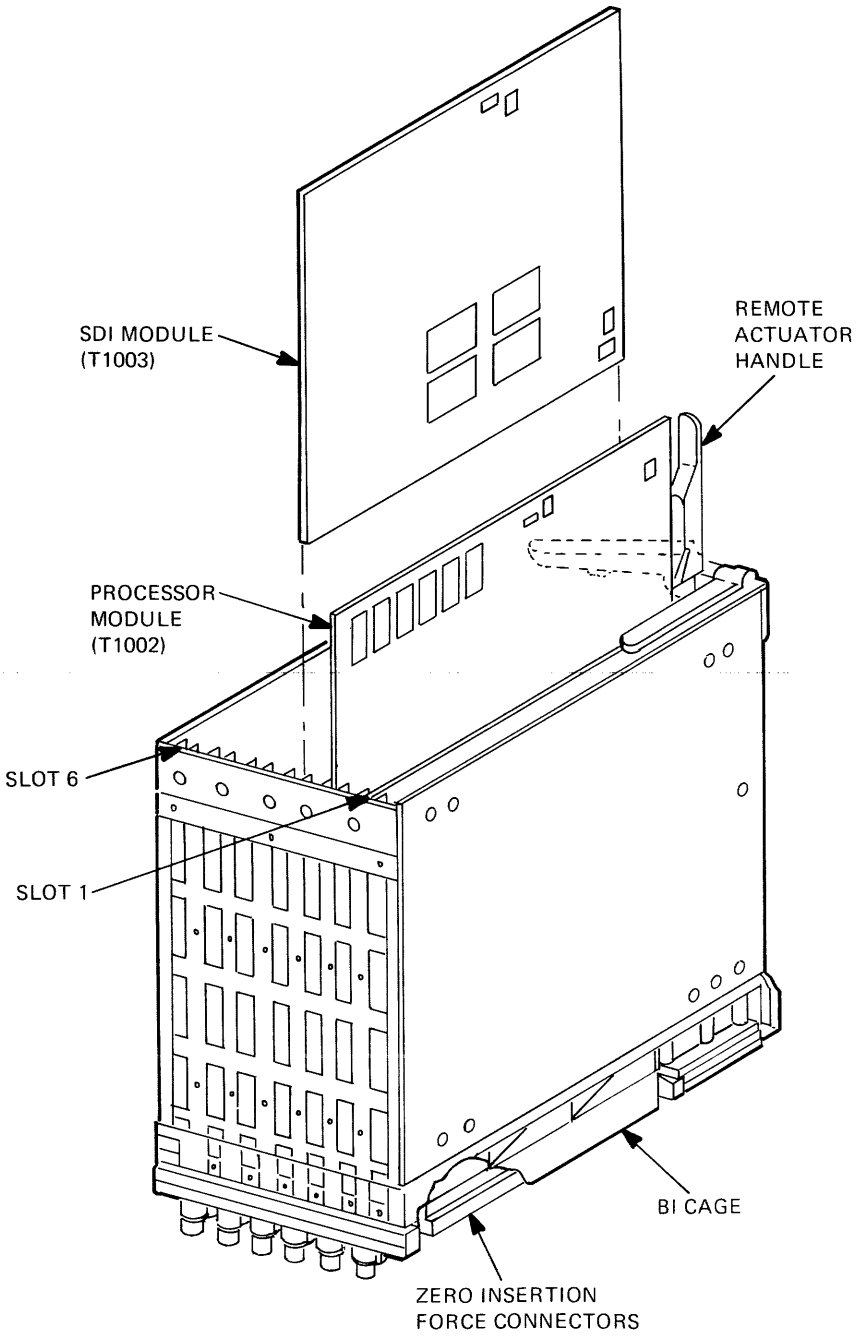
WARNING

> Before installing the KDB5Ø, make sure that the system power is off. Also, to ensure electro-static protection, wear either an anti-static wriststrap or heelstrap.

To install the KDB5Ø:

1. Gain access to the BI cage.

2. Insert the two KDB5Ø modules into adjacent slots in the same BI cage (Figure 2-1). The KDB5Ø SDI module should be placed in the higher numbered (or left) slot because of the configuration of the KDB5Ø backplane I/O assembly. However, the location of the two modules can be reversed if you interchange the section C transition segment with the internal SDI cables in the transition header assembly (Figure 2-2).



NOTE: MODULE POSITIONS ARE FOR DEMONSTRATION PURPOSES ONLY.

CX-722A

Figure 2-1 Inserting KDB50 Modules

3. Secure the modules in the BI cage by closing the remote actuator handle for each KDB5Ø module slot (Figure 2-1). Ensure that the handle closes all the way, signifying that the modules are properly inserted. The ZIF (Zero Insertion Force) connectors provide the electrical connection between the KDB5Ø backplane I/O assembly and the KDB5Ø modules.

4. Attach the KDB5Ø backplane I/O assembly to the backplane of the BI cage using the following procedure (Figure 2-2):

NOTE

Later versions of the KDB5Ø backplane I/O assembly should have molded cables.

a. Gain access to the BI backplane.

NOTE

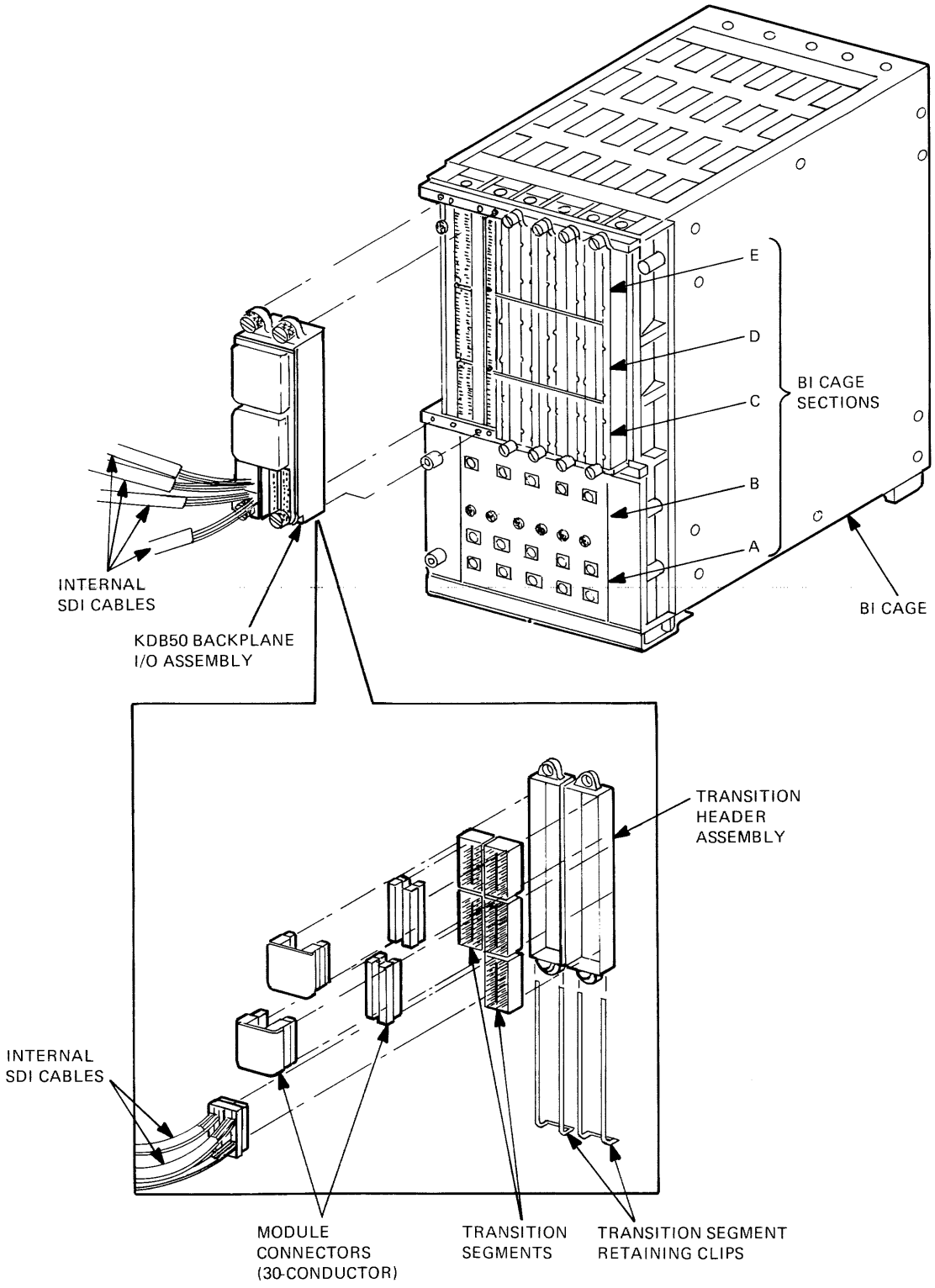
Choose a unique node ID plug from your system set of plugs. Insert this node ID plug in the area that corresponds to the KDB5Ø Processor module slot.

b. Remove the dust covers from the BI backplane.

c. Attach the keyed KDB5Ø backplane I/O assembly to the backplane sections that correspond to the KDB5Ø module slots.

CAUTION

To ensure that all screws are tightened evenly, only use the torque screwdriver (29-27381-ØØ) provided with the system. Tighten each screw only half-way, initially.



NOTE: KDB50 I/O BULKHEAD ASSEMBLY POSITION IS FOR DEMONSTRATION PURPOSES ONLY.

CX-723A

Figure 2-2 Attaching The KDB5Ø Backplane I/O Assembly

d. Return the BI cage to its original position.

e. Refer to your CPU user guide or installation manual to determine the proper internal SDI routing.

NOTE

Route the internal SDI cables separately from the power cables.

5. Attach the I/O bulkhead on the VAXBI-based system cabinet using the following procedure:

   a. Bring the internal SDI cables through the opening of one of the available option bulkhead panels of the VAXBI-based system backframe.

   b. Screw the internal SDI cables (J1 through J4) into the small holes on either side of each port on the I/O bulkhead (Figure 2-3). You will need to turn two of the keyed internal SDI cables one-half of a complete turn (180 degrees) to complete this task.

   c. Mount the I/O bulkhead on the option bulkhead panel.

6. Attach the keyed, external SDI cables to the VAXBI cabinet I/O bulkhead (Figure 2-4).

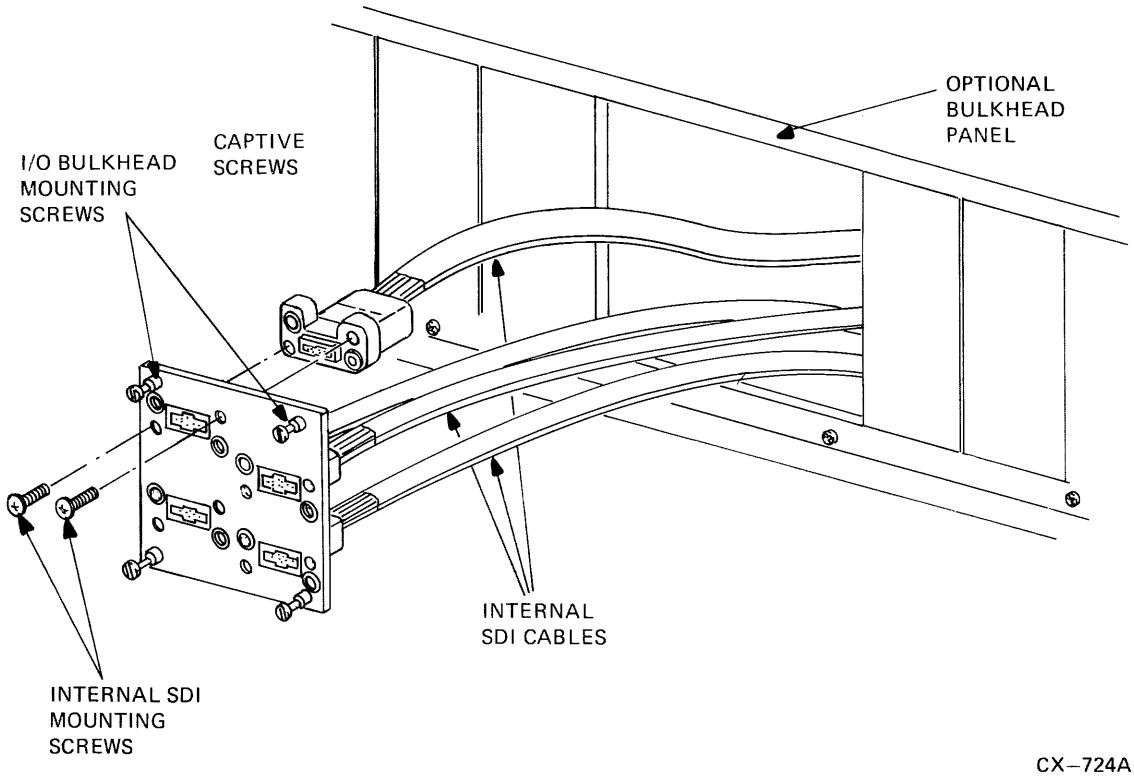


Figure 2-3 Internal SDI Cable Installation

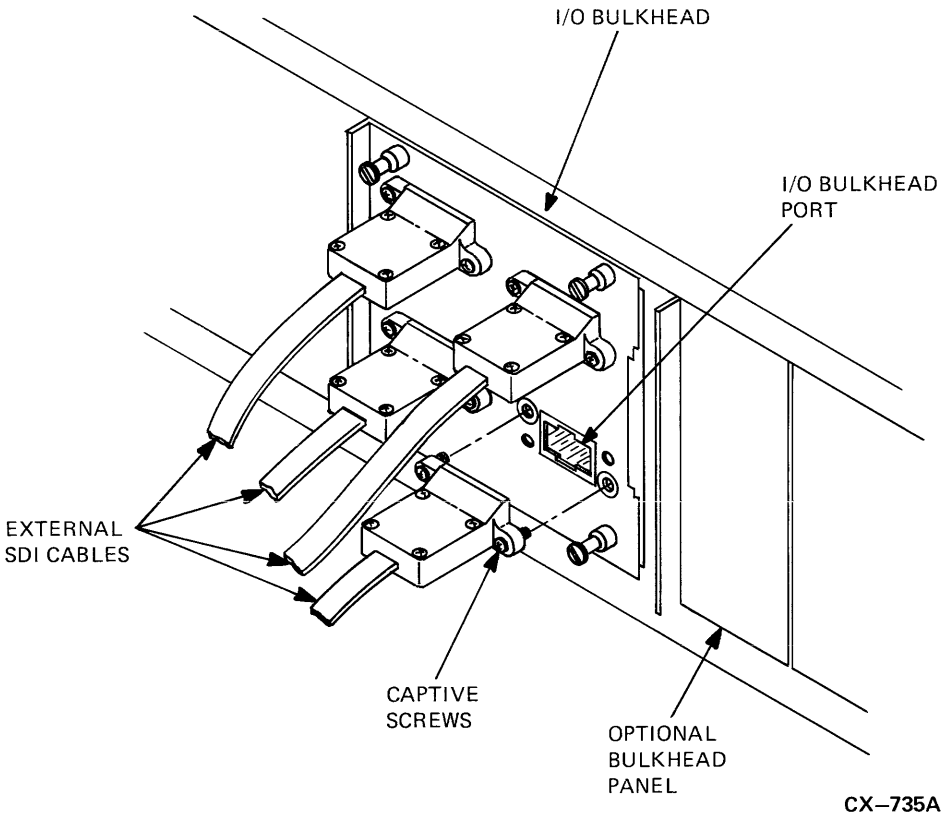


Figure 2-4 External SDI Cable Installation

7. Power-up the system: The selftest on the KDB5Ø module set is activated upon power-up. At this time, two single yellow LED indicators show the current state of the KDB5Ø. Each yellow LED has the same function to enable viewing from different locations. Refer to Figure 2-5 for the location of these LEDs.

   Check that the yellow LEDs on each KDB5Ø module turn on after about ten seconds.

   If the yellow LEDs do not turn on, see Section 2.2.1.

2.2 FIELD ACCEPTANCE TEST PROCEDURE

The field acceptance and test procedure for the KDB5Ø Disk Subsystem has two parts:

1. The KDB5Ø Disk Controller selftest

2. The KDB5Ø Subsystem Diagnostics

2.2.1 KDB5Ø Disk Controller Selftest

ROM based diagnostics run upon powerup or anytime the KDB5Ø is initialized by the host. At this time, two sets of four red LED indicators on each KDB5Ø module should display a cycling pattern in the LEDs. Each set of red LEDs has the same function to enable viewing from different locations. (Refer to the comment following Table 2-1 (*), and to Figure 2-5 which shows the location of each set of four red LEDs on the KDB5Ø modules.) The cycling pattern in the LEDs signifies the completion of a successful KDB5Ø selftest. If, however, the KDB5Ø red LEDs do not display the cycling pattern after power is applied, look at the LED code in Table 2-1 to locate the problem.

Table 2-1 LED Error And Symptom Codes

T1ØØ2 LEDs 8 4 2 1	T1ØØ3 LEDs 8 4 2 1	Error Symptoms	Most Likely Failure
NOTE: 1 = LED ON, Ø = LED OFF, x = LED may be ON or OFF			
Ø Ø Ø 1	x x x x	Hex 1; undefined	Undefined
Ø Ø 1 Ø	Ø Ø Ø Ø	Hex 2; microcode stuck in init step 2	T1ØØ2 or software
Ø Ø 1 1	Ø Ø Ø Ø	Hex 3; microcode stuck in init step 3	T1ØØ2 or software
Ø 1 Ø Ø	Ø Ø Ø Ø	Hex 4; microcode stuck in init step 4 or VAXBI bus timeout error	T1ØØ2 or host inactive
Ø 1 Ø BLINK	Ø Ø Ø Ø	Hex 4/5; test complete Normal display for operating KDB5Ø	No problem
Ø 1 1 Ø x x x x	x x x x Ø 1 1 Ø	Hex 6; undefined	Undefined
Ø 1 1 1 x x x x	x x x x Ø 1 1 1	Hex 7; undefined	Undefined
1 Ø Ø Ø	Ø Ø Ø Ø	Hex 8; wrap bit 14 set in SA register	T1ØØ2 or software
1 Ø Ø 1 Ø Ø Ø Ø	Ø Ø Ø Ø 1 Ø Ø 1	Hex 9; board one error	T1ØØ2
1 Ø 1 Ø 1 Ø 1 Ø	Ø Ø Ø Ø 1 Ø 1 Ø	Hex A; board two error	T1ØØ3
1 Ø 1 1 x x x x	x x x x 1 Ø 1 1	Hex B; undefined	Undefined
x x x x 1 1 Ø Ø	1 1 Ø Ø x x x x	Hex C; Timeout error, check error code in SA register. Refer to KDB5Ø Service Manual	Many causes

Table 2-1 (cont.)

T1Ø02 LEDs 8 4 2 1	T1Ø03 LEDs 8 4 2 1	Error Symptoms	Most Likely Failure
1 1 Ø 1 x x x x	x x x x 1 1 Ø 1	Hex D; RAM parity error	T1Ø03
1 1 1 Ø x x x x	x x x x 1 1 1 Ø	Hex E; ROM parity error	T1Ø02
1 1 1 1	1 1 1 1	Hex F; sequencer error	T1Ø02
Cycling pattern	Cycling pattern	None	No problem *
		The cycling pattern continues beyond the start of the initialization process. The KDB5Ø is not responding to the host CPU.	T1Ø02

* During a cycling pattern, LEDs begin flashing on the T1Ø02 module and then progress to the T1Ø03 module. The LEDs flash one at a time, starting at the LSB (least significant bit) and progressing through the MSB (most significant bit). The flash goes on and off for approximately a quarter-second, and then repeats at about a seven-second rate. However, the pattern is executed so fast that it looks like all the LEDs are flashing at the same time.

The LEDs normally cycle while the KDB5Ø is waiting for the host to start the initialization process. At this time, the KDB5Ø responds to the initialization and the cycling pattern stops. This normally occurs in about seven seconds if the system software is prepared to establish a connection with the KDB5Ø.

When two codes are given for the same error, both indicate the same failure.

2.2.2 KDB5Ø Subsystem Diagnostics

CAUTION

The diagnostic printout identifies the KDB5Ø software and hardware revision levels. Determine if the stated revision levels are compatible. The locations of the hardware revision level jumpers are shown in Figure 2-5.

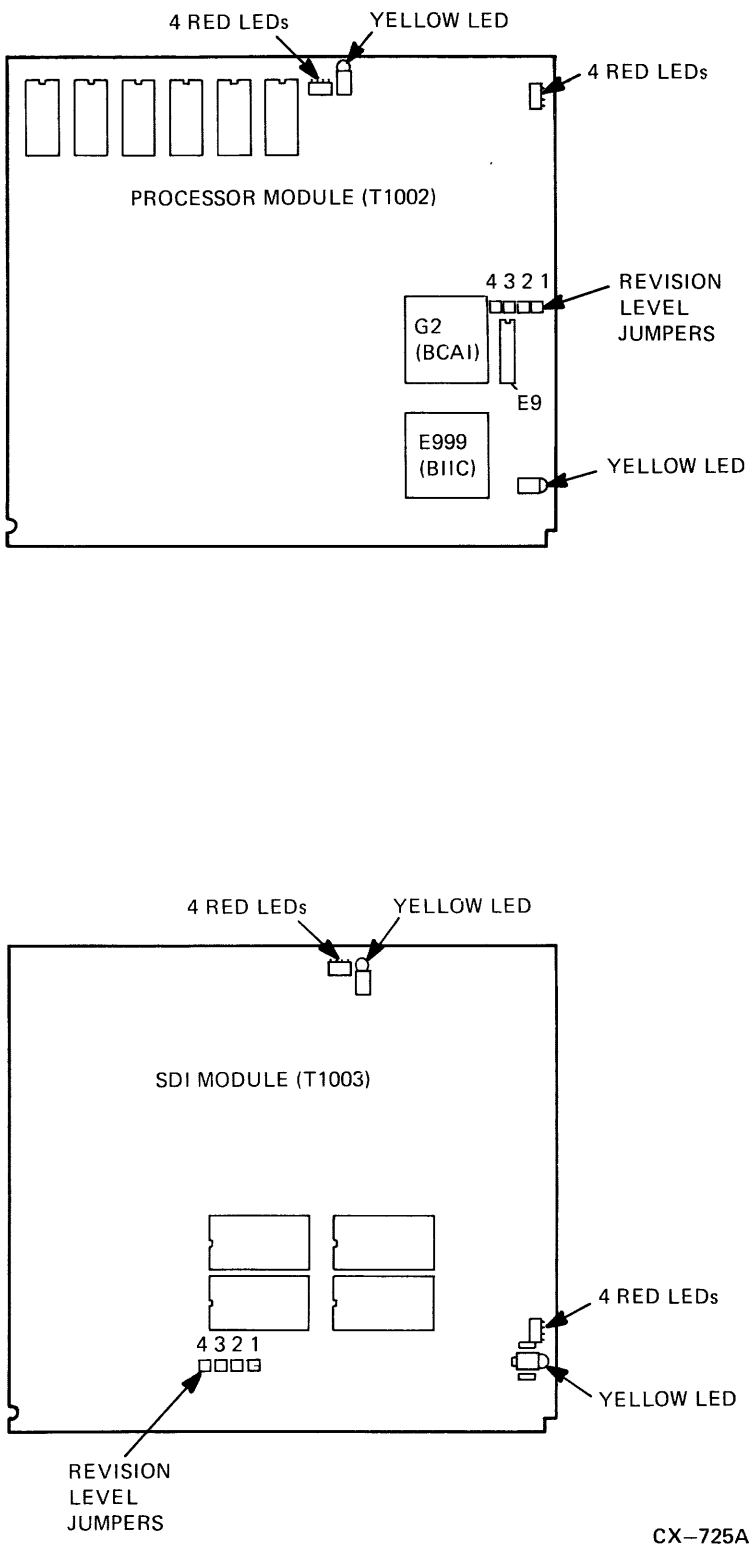


Figure 2-5 KDB50 Module LEDs And Revision Level Jumpers

NOTE

These diagnostics are subject to change due to future revisions. Always follow the program directions that are displayed on your terminal. Refer to the software documentation for detailed descriptions and other information.

The KDB5Ø subsystem diagnostics are used with the KDB5Ø controller. The following paragraphs describe these diagnostics.

NOTE

If the diagnostic program reports errors, refer to the KDB5Ø Disk Controller Service Manual (EK-KDB5Ø-SV).

- EVRLB (KDB5Ø Disk Drive Formatter Utility--uses DM code)

CAUTION

When EVRLB is run, it erases customer data on the disk. EVRLB should only be used after it has been established that a hardware problem has created false entries in the RCT (Replacement Control Table).

EVRLB, when run, will:

1. clear the RCT
2. move the FCT (factory bad block control table) to the RCT
3. write and verify each sector and replace (or revector) as necessary

- EVRLF (UDA5Ø/KDB5Ø Basic Subsystem Diagnostic--uses DM code)

  EVRLF consists of the following three tests:

  1. Test #1 VAXBI Bus addressing test

     This test checks to make sure that each addressing line can be driven to both one and zero (that the KDB5Ø addresses unique locations as each line is toggled). The test then does large transfers to and from memory with known patterns.

  2. Test #2 Disk-Resident diagnostic test

     This test issues a DIAGNOSE command to each disk drive selected for testing. Each drive should accept the DIAGNOSE command and respond. Failure status and all data from the disk drive will be sent to the host for error reporting. An interactive dialogue allows an operator to manually select any other diagnostic options available in the disk drive.

  3. Test #3 Disk function test

     This test checks out all disk functions as defined by the SDI. The test will perform such functions as DRIVE CLEAR, READ, WRITE, SEEK, FORMAT, and RECALIBRATE. All writing and formatting will only be done on the diagnostic cylinders of the disk drives. An extensive test of the positioning capability will be performed before attempting any format operation. Positional verification will be implemented before any formatting will be allowed.

- EVRLG (KDB5Ø Disk Exerciser Test--uses DM code)

  This exerciser will test the disk drives selected by issuing random seeks and read/write transfers. Parameters can be specified to use diagnostic or host cylinders, restrict testing to specific areas of the disk, control length of test, and do error logging.

  EVRLG will duplicate all the firmware error recovery mechanisms, thereby thoroughly checking out the KDB5Ø firmware.

Before running the KDB5Ø Subsystem Diagnostics, the diagnostic supervisor must be loaded and started. If you need assistance in this task, refer to the appropriate System Manual.

The following procedure describes how to load the KDB5Ø Subsystem Diagnostics:

```
DS>Load EVRnn
DS> AT
Device type?  KDB5Ø
Device Link?  HUB
Device Name?  DUA (or DUB)
```

NOTE

> The presence of B at the third character position in this string indicates the existence of a second UDA5Ø/KDA5Ø-Q/KDB5Ø controller.

Node ID ?

The response range is Ø to 15. This value is on the node ID plug that is attached to the BI cage backplane for the KDB5Ø Processor module.

BR Level ?

Type in 5. This is the VAXBI interrupt level assigned to the KDB5Ø.

```
DS> AT
Device type?  RAnn
Device link?  DUA (or DUB)
Device Name?  DUAØ (or DUBØ)
DS>
```

An RA6Ø must be attached by using the following sequence:

```
DS> AT
Device type?  RA6Ø
Device link?  DUA (or DUB)
Device Name?  DJAØ (or DJBØ)
```

NOTE

Notice the J instead of the U. This is how VMS identifies the drive as removable.

```
DS>Select DUA,DUAØ,DJAØ
DS>Run (or Start) EVRnn
.Program: EVRnn - VAX nnnn UDA5Ø/KDB5Ø Basic Subsystem Diagnostic
revision Ø.Ø
```

2.3 SYSTEM AND SOFTWARE CONSIDERATIONS

The following sections describe common system and software concerns.

2.3.1 System Clock Or Timer

Some aspects of diagnostic and/or functional usage of a KDB5Ø depend upon the ability of the host processor to time-out on an operation.

2.3.2 Error Logs

The KDB5Ø has the ability to return information to the operating system for inclusion into an error log. These entries may include specific information on the operation of the KDB5Ø, its attached drives, or other parts of the system (examples are host processor, memory, software) which may be important in diagnosing problem sources.

NOTE

In order to support the fault tolerant and error recovery features of the KDB5Ø, the host Operating System must support Bad Block Replacement and error logging.

Some reports contained in the error log, however, may represent changes in the configuration or operation of your system that are only informational in nature and do not represent the occurrence of an actual error condition. Examples of this may be:

- Completion of the initialization sequence between the port driver and the KDB5Ø.
- Attention messages pertaining to the availability of a disk drive (which may be the result of changing a drive's unit number).

For further directions, refer to the appropriate system documentation section.

2.3.3 Drive Numbering

DSA/SDI drives that can be connected to the KDB5Ø can usually be given a unit number ranging from Ø to 254. However, some operating systems do not support this entire range of unit numbers and may only support the range Ø to 7. Consult operating system documentation for accepted drive numbers. Consult drive documentation for unit number programming instructions.

The unit numbers assigned to drives attached to a KDB5Ø do not imply any priority or other special property: all drives are treated equally by the KDB5Ø. The only requirement is to avoid duplicating drive unit numbers. If these numbers are duplicated, the KDB5Ø will not permit a drive to be accessed. This situation may be corrected by assigning unique drive numbers.

Unit numbers can usually be easily changed at the drive, although this is recommended only when the intended drive is offline to the KDB5Ø (not mounted by the operating system).

CHAPTER 3
KDB50 PROGRAMMER INFORMATION

3.1 KDB50-SPECIFIC PROGRAMMING INFORMATION

All register addresses supplied in this chapter are of the form bb + nn, where:

1. bb = 20000000H + (2000H x Node ID)

2. nn varies depending on the register

For example, if the KDB50 is node 7, then the BI Control and Status (BICSR) register (bb + 04) is at address:

20000000H + (2000H x 7) + 04 = 2000E004H.

The following information is necessary to write software for the KDB50:

- The KDB50 initializing and polling register address is bb + F2.

- The KDB50 allows the host to set the vector address. This address is used for both error interrupts (generated by the BIIC) and user interface interrupts (generated by the BPROC microcode stream).

  NOTE

  The vector address varies with different system configurations.

- The KDB50 has a command limit value of 21. This includes 20 MSCP commands plus 1 immediate-only command.

- The KDB50 supports only 512-byte disk formats.

- The KDB50 supports MSCP and UQSSP.

- The diagnostic option capabilities available on the KDB5Ø are purge/poll and wrap.
- The KDB5Ø supports maintenance read and maintenance write to and from the KDB5Ø RAM.
- The KDB5Ø supports last fail log packets.

3.2 BIIC REGISTERS

The KDB5Ø uses 9 BIIC registers which are divided into three groups: five VAXBI-required registers, two BIIC control registers, and two KDB5Ø specific registers. The following sections describe these registers as implemented on the KDB5Ø.

NOTE

Refer to your System Reference Manual for more detailed information.

3.2.1 VAXBI Required Registers

The VAXBI-required registers are the minimum set of registers that any node needs to operate on the VAXBI bus. The required registers are:

1. DEVICE REGISTER (DTYPE)
2. CONTROL AND STATUS REGISTER (BICSR)
3. BUS ERROR REGISTER (BER)
4. ERROR INTERRUPT CONTROL REGISTER (EINTRCSR)
5. INTERRUPT DESTINATION REGISTER (INTRDES)

The following paragraphs describe the VAXBI-required registers.

3.2.1.1 DTYPE - The DTYPE register (address bb + ØØ) identifies the type of node and its revision level. The upper 16 bits (31:16) provide the microcode and BI module revision level information. The lower 16 bits (15:ØØ) provide the device-type information. Figure 3-1 shows the format of the DTYPE register.

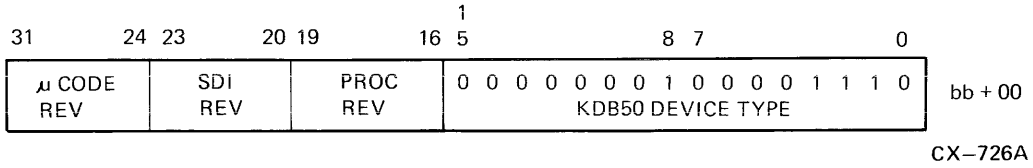


Figure 3-1 KDB5Ø DTYPE Register Format

3.2.1.2 BICSR - The VAXBI control and status register (BICSR) (address bb + Ø4) provide status and error indicators along with the NODE ID. Figure 3-2 shows the format of the BICSR register.

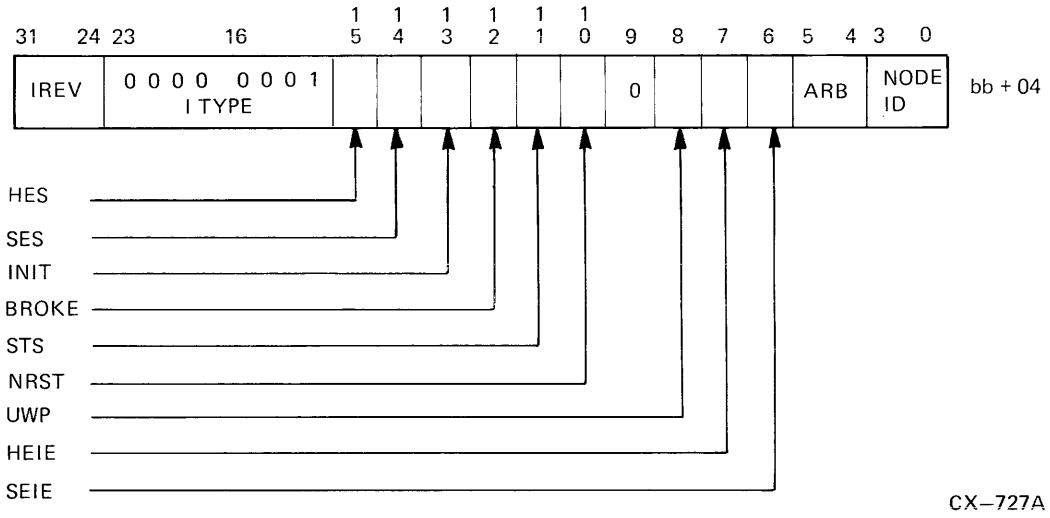


Figure 3-2 KDB5Ø BICSR Register Format

3.2.1.3 BER - The bus error register (BER) (address bb + Ø8) provides VAXBI transaction error status. Unless otherwise noted, all bits in the bus error register can be set during VAXBI and loopback transactions. Bits <3Ø:16> are hard error bits, and bits <2:Ø> are soft error bits. Bit <3>, the User Parity Enable (UPEN) bit, is not an error bit; it indicates the BIIC parity mode. (The BIIC contains all necessary logic to set the bus error register bits.) Figure 3-3 shows the format of the bus error register.

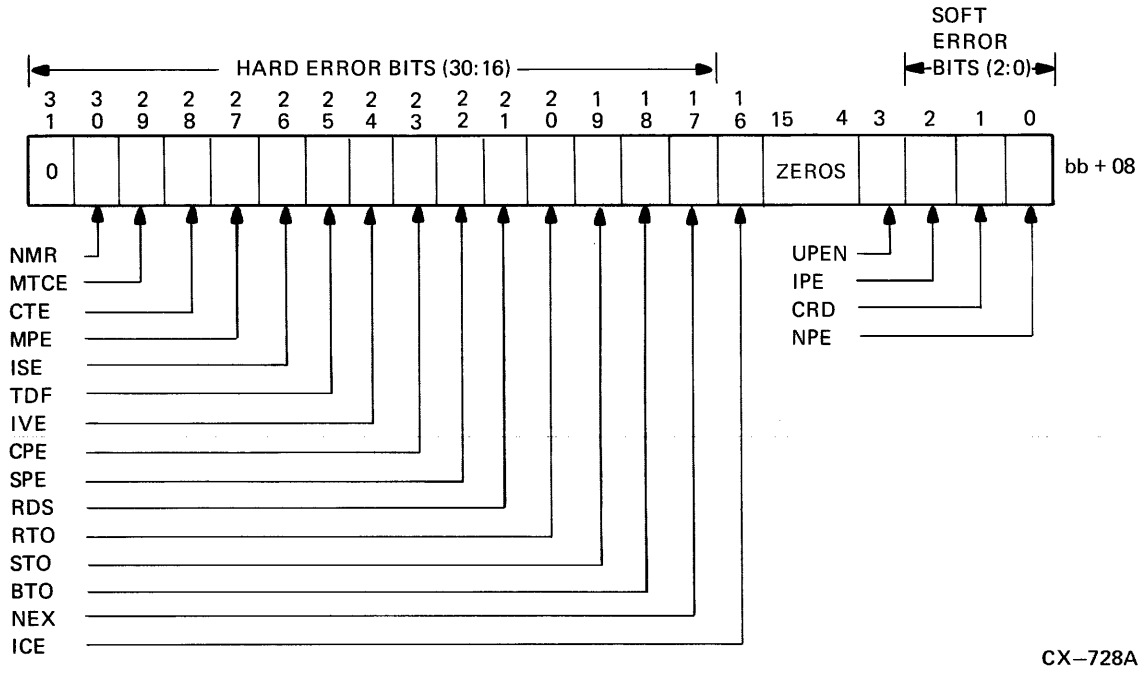


Figure 3-3 Bus Error Register Format

3.2.1.4 EINTRCSR - The error interrupt control register (EINTRCSR) (address bb + ØC) controls the operation of error interrupts--that is, error interrupts the BIIC generates when it detects a bus error. Figure 3-4 shows the format of the error interrupt control register.

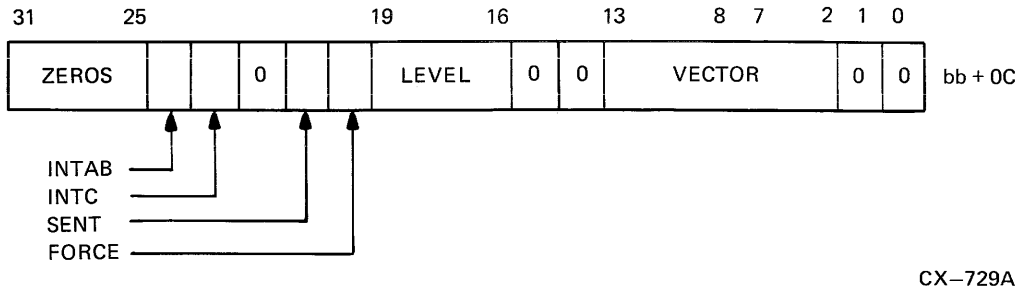


Figure 3-4 Error Interrupt Control Register Format

Two bits in the BICSR register enable BIIC-generated interrupts: HEIE for hard errors, and SEIE for soft errors. For the KDB5Ø, the host sets HEIE (SEIE is not set) just after the KDB5Ø finishes the self-test. Therefore, the KDB5Ø BIIC generates an error interrupt for every hard error it detects (HES bit set in the BICSR).

3.2.1.5 INTRDES - The KDB5Ø can generate two types of interrupts: interrupts the BIIC generates when it detects a bus error (which sets a bit in the BER register), and interrupts the BPROC generates by setting a bit in the UINTRCSR register. The interrupt destination register (INTRDES) (address bb + 1Ø) supplies the BI node address for both types of interrupts. Figure 3-5 shows the format of the interrupt destination register.

NOTE

After the KDB5Ø completes the self-test, the host initializes the INTRDES field to the host BI node address (NODE ID).

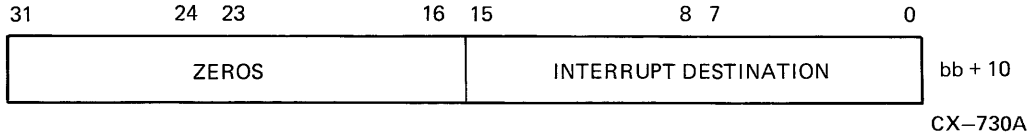


Figure 3-5 Interrupt Destination Register Format

3.2.2 BIIC Control Registers

The two control registers give the KDB5Ø microcode additional control over the BIIC and the ability to issue interrupts without using BCI INT <7:4>. They are:

1. BCI CONTROL REGISTER (BCICSR)

2. USER INTERFACE [KDB5Ø] INTERRUPT CONTROL REGISTER (UINTRCSR)

The following paragraphs describe the BIIC control registers.

3.2.2.1 **BCICSR** - In general, the BCI control and status register (BCICSR) (bb + 28) contains bits that enable or disable various BIIC operations. Figure 3-6 shows the format of the BCI control and status register.

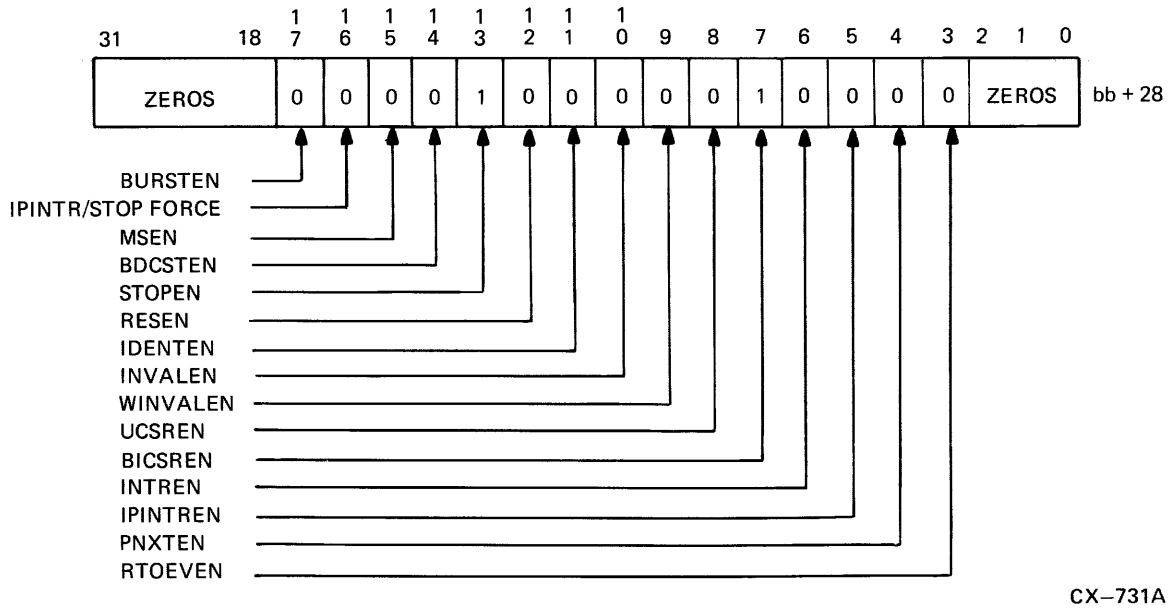


Figure 3-6 BCI Control And Status Register Format

3.2.2.2 **UINTRCSR** - The user interface interrupt control register (UINTRCSR) (bb + 4Ø) controls and monitors interrupts that the KDB5Ø generates. The KDB5Ø sets a force bit in this register to generate interrupts. Figure 3-7 shows the format of the user interface interrupt control register.

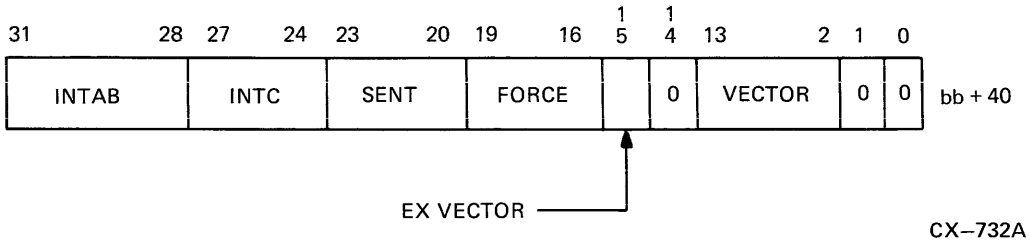


Figure 3-7 User Interface Interrupt Control Register

3.2.3 KDB5Ø Specific Registers

The KDB5Ø-specific registers are three BIIC general-purpose registers that implement the KDB5Ø IP and SA registers. The KDB5Ø specific registers are:

1. INITIALIZING AND POLLING (IP) REGISTER (GPRØ)
2. STATUS AND ADDRESS (SA) READ REGISTER (lower word of GPR1)
3. STATUS AND ADDRESS (SA) WRITE REGISTER (upper word of GPR1)

The following paragraphs describe the KDB5Ø-specific registers.

3.2.3.1 IP - The host informs the KDB5Ø that a command is in the command queue by reading the IP register (address bb + F2). When the host reads the IP register and a connection between the host and the KDB5Ø exists, the KDB5Ø examines the host-command queue for a command. The IP register exists only as a communications mechanism and has no specific bit definitions. Figure 3-8 shows the format of the KDB5Ø IP register.

NOTE

Writing the IP register has no effect.

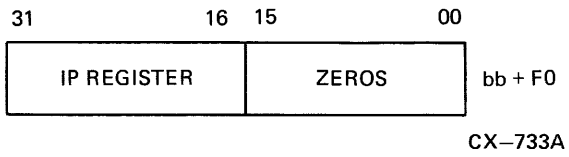


Figure 3-8 KDB5Ø IP Register Format

3.2.3.2 SA - The SA register is made up of a host read only register (SAr) and a host write only register (SAw). Together, these registers perform the following three functions:

1. When the host reads SAr during initialization, SAr contains data and error information relating to the initialization process.

2. When the host writes SAw during initialization, it communicates host-specific parameters to the KDB5Ø.

3. When the host reads SAr during normal operation, SAr contains status information, including port and controller-detected fatal errors.

For the KDB5Ø, the SA register is made up of two 16-bit registers which reside in one 32-bit general purpose BIIC register (GPR1). The SA register is also divided into read and write sections: SAr and SAw. SAr (address bb + F4) [viewed from the host] is the read portion of the SA register, while SAw (address bb + F6) [viewed from the host] is the write portion of the SA register. Certain bits in the SAr register are constant and do not depend on the initialization step. During normal operation, the SA register has the format shown in Figure 3-9.

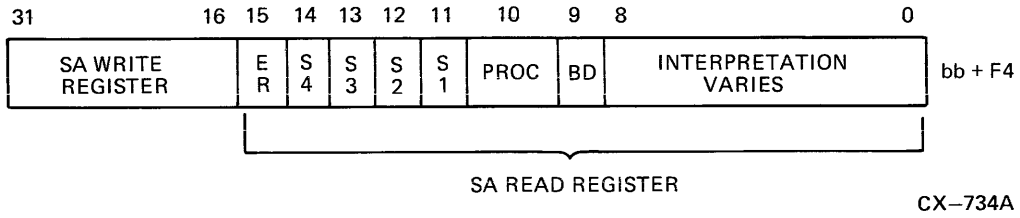


Figure 3-9 KDB5Ø General SA Register Format

NOTE

The data portion of the SAr register (bits Ø-8) is qualified either by the ER bit (bit 15) or by one of the step bits (bits 11-14). The content of the SAr register is unspecified if the ER bit is clear and a step bit is not set.

APPENDIX A

GLOSSARY

This glossary defines terms found in many publications that describe DSA products. It also contains terms used to describe devices that interface to the VAXBI bus.

ABORTED COMMAND

From the viewpoint of a class driver, any command for which the driver has issued an ABORT command. From the viewpoint of an MSCP server, a command for which the server has received an ABORT command, located the specified command, and taken explicit action to abort it. An aborted command will be either rejected or terminated.

ACK DATA CYCLE

A data cycle of a VAXBI read- or write-type transaction during which the slave asserts the ACK confirmation (CNF) code to acknowledge that no error has been detected and that the cycle will not be STALLed.

ADAPTER

A device that adapts information from the host bus format to a different bus format. Sometimes controllers are referred to as adapters because they adapt information from the host bus format to the tape or disk drive bus (STI, SDI) format.

ARBITRATION CYCLE

A cycle during which nodes arbitrate for control of the VAXBI bus.

ASSERT

To cause a signal to become "true" (asserted).

ASSERTED

Synonymous with true. Or, to be in the true state.

ASSERTION

The transition of a signal from false to true.

ATTENTION CONDITION

A status change in the drive significant enough to warrant controller interaction.

BAD BLOCK

A block containing a defect that:

1. Exceeds the correction capability of the subsystem error correction scheme.

2. Exceeds a drive-specified error threshold. Once a block exceeds this threshold, data integrity is not guaranteed.

3. Imposes too great a strain on system performance. In this case, the subsystem still assures data integrity, but the extensive error correction required for each block access causes too great a strain on system performance.

GLOSSARY

BAD BLOCK REPLACEMENT

Substituting a spare block (replacement block) for a bad block.

BAD BLOCK REVECTORING

Transferring a Read or a Write operation from a bad block to a replacement block. Once a bad block is replaced, revectoring occurs upon each access to the bad block. The two types of revectoring in ascending order of complexity and implementation are:

1. Primary -- The operation is revectored to the first replacement block on the current track.

2. Non-Primary -- The subsystem uses non-primary revectoring when primary revectoring is not possible. During non-primary revectoring, the operation revectors to a replacement block address supplied by the replacement and caching table (RCT).

BANDWIDTH

The data transfer rate measured in information units transferred per unit of time (for example, megabytes per second). All bandwidth figures quoted in this manual take into account command/address and imbedded ARB cycle overhead.

BCAI

Backplane Interconnect Chip Interface Adapter Interface; a chip that serves as an interface between the BCI and the KDB5Ø DBUS and control lines.

BCI

Backplane Interconnect Chip Interface; a synchronous interface bus that provides for all communication between the BIIC and the KDB5Ø control lines and internal data bus.

BIIC

Backplane interconnect interface chip; a chip that serves as a general purpose interface to the VAXBI bus.

BIIC CSR SPACE

The first 256 bytes of the 8-Kbyte nodespace which is allocated to BIIC internal registers. See also NODESPACE.

BIIC-GENERATED REQUEST

A transaction request generated by the BIIC rather than by the user interface (KDB5Ø data and control lines). The BIIC can request only error interrupts.

BIIC-GENERATED TRANSACTION

A transaction performed solely by the BIIC with no assistance from the master port interface. Only INTR and IPINTR transactions can be independently generated by the BIIC. The user interface initiates BIIC-generated transactions by using the IPINTR/STOP force bit, the user interface or error interrupt force bits, or by asserting one of the BCI INT<7:4> L lines. A BIIC-generated transaction can also result from a BIIC-generated request which results from a bus error that sets a bit in the Bus Error Register.

BIT

The term bit stands for Binary Digit. A binary digit assumes one of two possible states: Ø or 1. Sometimes, the terms false and true, or deasserted and asserted, or low and high are substituted for Ø and 1. However, care must be taken not to assume these terms are always synonymous. For example, a signal may be "true when low", and therefore asserted; however, the physical logic level would be the low logic level usually associated with Ø.

BLOCK

Block is synonymous with sector. A block is the smallest data unit addressable on a subunit. It occupies a specific physical position relative to the index, and it is available for reading or writing once per disk rotation. Five types of blocks follow:

1. Diagnostic Block -- The diagnostic block area is used for drive read/write diagnostics. This area is not visible to the host operating system. However, it is visible to the controller. Diagnostic block addresses are 28-bits wide and are called Diagnostic Block Numbers (DBNs).

2. External Block -- Contains the format control tables. The external block area is not visible to the host operating system. However, it is visible to the controller. External block addresses are 28-bits wide and are called External Block Numbers (XBNs).

3. Logical Block -- Contains the host applications area and the replacement and caching table. All logical blocks are visible to the host operating system. Logical block addresses are 28-bits wide and are called Logical Block Numbers (LBNs).

4. Physical Block -- This category contains all the blocks on a subunit. DBNs, LBNs, RBNs, and XBNs are subsets of the physical block area. Physical block addresses are 28-bits wide and are called Physical Block Numbers (PBNs).

5. Replacement Block -- A reserved block used as a replacement for a bad block on a subunit. Replacement block addresses are 28-bits wide and are called Replacement Block Numbers (RBNs).

BUS ACCESS LATENCY

The delay from the time a node desires to perform a transaction on the VAXBI bus until it becomes master.

BUS ADAPTER

See ADAPTER

BUSY EXTENSION CYCLE

A bus cycle during which a VAXBI node, not necessarily the master or the slave of a transaction, asserts the BI BSY L line to delay the start of the next transaction.

BYTE

A group of eight bits. For example, 10010101.

CLASS DRIVER

A class driver is a piece of software residing in the host that communicates with a class of devices, such as a disk class. The communication occurs via a port driver and MSCP messages. Hosts generally have a class driver for each class of device: disks, tapes, and the controller itself (for maintenance commands). For example, a disk class driver can communicate with a disk subsystem that has any combination of SDI disks. Contrast this with a device driver that can communicate with only one type of disk.

CLOCKS

Any signal that supplies timing or initiates an action on one or more devices can be considered a clock. A clock does not have to be continuous.

Pertaining to the SDI bus: The clock signal for all the SDI lines is imbedded within the SDI data pattern. Therefore, any data pattern (ones, zeros, or combination) generates a clock at the receiving device.

COMMAND/ADDRESS CYCLE

The first cycle of a VAXBI transaction. The information transmitted in this cycle is used to determine slave selection. In some cases the data on the BI D<31:0> L lines is not an actual address, but it serves the same purpose: to select the desired slave node(s). For example, during an INTR command a destination mask is used.

COMMAND AND RESPONSE RING BUFFER

For UNIBUS, QBUS, and some VAXBI bus controllers: An area in host memory that the controller and host use for communication. The ring buffer points to the location in host memory that contains command messages from the host and response messages from the controller.

COMMAND CATEGORIES

All MSCP commands fall into one of the following four command categories:

1. Immediate commands
2. Non-sequential commands
3. Sequential commands
4. Special commands

Each command category has certain constraints on when they may be executed, thus limiting the scope of controller optimizations.

COMMAND CONFIRMATION

The response sent by the slave(s) to the bus master to confirm participation in the transaction.

COMMAND CONFIRMATION CYCLE

The third cycle of a VAXBI transaction during which slave(s) confirm participation in the transaction.

COMMAND TIMER AND RESPONSE TIMER

Mechanisms in the drive and the controller that monitor controller/drive communication. They signal (time out) when controller/drive communication breaks down. Basically, if the drive doesn't receive a command from the controller, or if the controller doesn't receive a response from the drive within the appropriate time, the command or response timer

signals a breakdown in controller/drive communication.

COMMUNICATIONS PROTOCOL

A set of rules and conventions that devices use when communicating. For example, MSCP or SDI.

COMPLETED COMMAND

A command that the MSCP server executes entirely. The completed response is either "successful" or "error".

CONFIGURATION DATA

Data loaded into the BIIC on power up that includes the device type and revision code, the parity mode, and the node ID.

CONNECTION

A software term indicating useful MSCP communication exists between the intelligent controller microcode and the class driver. (Refer to CONTROLLER ON-LINE, CONTROLLER OFF-LINE, and CONTROLLER AVAILABLE.)

CONTROL MESSAGE

A set of sequentially transmitted frames that begins with a start frame and terminates with an end frame. The message contents convey information between the controller and drive. Control messages from the drive are called "Responses" and control messages from the controller are called "Commands".

CONTROLLER

An interface between the host computer and I/O peripherals. It communicates with the hosts via the Mass Storage Control Protocol, and to the peripherals through the STI or SDI. Sometimes called an adapter because it adapts information

from the host bus format to the SDI or STI format.

CONTROLLER AVAILABLE

State of the controller if its connection is not made, yet the controller recognizes a connection can be made.

CONTROLLER BYTE

One of the standard drive status bytes outside the generic status. It is used by the controller to record information necessary for the proper operation of the MSCP protocol.

CONTROLLER OFF-LINE

Controller state if its connection is not made and the controller recognizes a connection cannot be made.

CONTROLLER ON-LINE

Condition of the controller if it has a functioning connection to the class driver.

CONTROLLER TIMEOUT INTERVAL

A time interval measured in seconds that is supplied by the controller or MSCP server in the end message of the SET CONTROLLER CHARACTERISTICS command. Controllers or MSCP servers guarantee the completion of all immediate commands, plus some measurable amount of useful work on their oldest outstanding non-immediate command within the controller timeout interval.

CYCLE

The basic VAXBI bus cycle of 200 nanoseconds (nominal).

CYLINDER

See DISK GEOMETRY

DATA BIT TIME

At specific data rates, the time from data clock edge to data clock edge. For example, at 1 MHz the data bit time is 1/1MHz or 1 microsecond.

DATA CYCLE

A cycle in which the VAXBI data path is dedicated to transferring data (such as read or write data, as opposed to command/address or arbitration information) between the master and slave(s). During read STALL data cycles, the BI D<31:Ø> L and BI I<3:Ø> L lines contain undefined data. See also ACK DATA CYCLE, READ DATA CYCLE, STALL DATA CYCLE, VECTOR DATA CYCLE, and WRITE DATA CYCLE.

DATA TRANSFER COMMANDS

VAXBI commands that involve the transfer of data as well as command/address information: read-type, write-type, IDENT, and BDCST commands.

DATAGRAM COMMUNICATION SERVICE

A communication service used across the connection between a class driver and an MSCP server (controller). This service is used for MSCP error log messages. It must deliver messages sent on it with very high probability. Messages may (with very low probability) be delivered out of sequence, lost, or duplicated. The datagram communication service supports messages of at least 384 bytes.

DEASSERT

To cause a signal to become "false" (deasserted).

DEASSERTED

Synonymous with false. Or, to be in the false state.

DEASSERTION

The transition of a signal from true to false.

DECODED ID

The node ID expressed as a single bit in a 16-bit field.

DEVICE TYPE

A 16-bit code that identifies the node type. This code is contained in the BIIC's Device Register.

DIAGNOSTIC BLOCK NUMBER (DBN)

See Block

DISK GEOMETRY

Disk geometry, in this case, refers to how a physical disk geometry is organized into a logical disk geometry. All SDI disks are organized into logical tracks, logical groups, and logical cylinders. The following describes the logical track, group, and cylinder interrelationships.

A logical cylinder represents a collection of groups. A group represents a collection of tracks. A track represents sets of sectors which occupy contiguous physical disk locations. Cylinders, groups, and tracks are all related by access time; not by any physical properties. The following list relates cylinder, group, and track by their relative access times:

1. Cylinder -- It takes longer than one disk revolution (in time) to access a logical cylinder. A cylinder change requires a level 2 INITIATE SEEK command from the controller.

2. Group -- It takes less than one disk revolution (in time) to access a logical group. A group change requires a level 1 select group type command from the controller.

3. Track -- A logical track can be accessed within the intersector time. (Intersector time is from the end of the current sector to the beginning of the next read/write sector.) A track change requires a level 1 select track type command from the controller.

Since different drives have different access properties, they have a different logical geometries. Three examples follow:

1. The RA8Ø has 4 platters and seven physical oxide surfaces used for data. Each data surface has two data heads for a total of 14. Any of the 14 data heads may be selected within the intersector time. Therefore, the RA8Ø has 14 tracks. These 14 tracks make up a group. A one PHYSICAL cylinder seek requires less than one revolution to perform. Therefore, the RA8Ø has 2 groups; the one its on and the group that is one PHYSICAL cylinder away. Any further positioning takes longer than one disk revolution. So, anything else is a logical cylinder change.

   Thus, the RA8Ø has a logical geometry of 14 tracks/group, 2 groups/cylinder, and 279 cylinders.

   NOTE

   The RA8Ø has 558 PHYSICAL cylinders.

2. Like the RA8Ø, the RA81 has 4 platters and seven physical oxide surfaces used for data. Each data surface has two data heads for a total of 14. However, unlike the RA8Ø, the RA81 servo technology requires head settling time greater than the intersector time when a head switch is performed. Therefore, the RA81 has only one track per group. The RA81 has 14 groups per cylinder because switching heads takes greater than the intersector time, but less than the disk revolution time. Any further positioning requires greater than one revolution to complete, and therefore constitutes a logical cylinder change.

   Thus, the RA81 has a logical geometry of 1 track/group, 14 groups/cylinder, and 1258 cylinders.

NOTE

On the RA81, a logical group is the same as a physical head, and a logical cylinder is the same as a physical cylinder.

3. Consider this example: Semiconductor technology is beginning to provide thin-film heads capable of accommodating several read/write heads on a single chip. If an RA81 was fitted with such heads, each physical arm (where a head is mounted; there are 14 of them) would have multiple heads (assume 8 for discussion) that would be selectable in the intersector time. Therefore, this drive would have 8 tracks/group. Assuming the servo technology on this drive is the same as the RA81 servo technology, selecting a head on a different arm would require greater than the intersector time and less than the disk revolution time (group change). Therefore, this drive would have 14 groups/cylinder.

   Thus, the hypothetical drive would have a logical geometry of 8 tracks/group, 14 groups/cylinder, and 1258 cylinders.

DMA ADAPTER

An adapter that directly performs block transfers of data to and from memory.

DRIVE-AVAILABLE

A drive state relative to the controller. The drive is operational, but is not currently online to any controller.

DRIVE-OFFLINE

A drive state relative to the controller. The drive is not operational and may not communicate with the controller via the drive control protocol.

DRIVE-ONLINE

A drive state relative to the controller. The drive is dedicated to the exclusive use of a particular controller and is not available to any other controller.

DRIVE-UNAVAILABLE

A drive state relative to the controller. The drive is operational and on-line to another controller.

EMBEDDED-CONTROLLER

A controller that does not use a standard interconnect with its devices, but instead is usually integrated with them. Such a controller must contain implicit knowledge of the characteristics of its devices.

ENCODED ID

The node ID expressed as a 4-bit binary number. The encoded ID is used for the master ID transmitted during an imbedded ARB cycle.

END MESSAGES

The last frame of a sequentially transmitted message that began with a start frame. The end frame contains a checksum for all the data fields of the message start and continuation frames.

ERROR BYTE

One of the status bytes in the get status response the controller receives from the disk drive. All possible drive detected errors must fit into one of the following five classes available in the error byte:

1. DE -- Drive error.
2. RE -- SDI transmission error.
3. PE -- Level 2 protocol error.

4. DF -- Initialization diagnostic failure.
5. WE -- Write lock error (attempt to write while write locked).

ERROR LOG MESSAGES

Messages that pass error information from the controller to the host. They are grouped into four basic formats:

1. SDI Error Log Format.
2. Host Bus Error Log Format.
3. Disk Transfer Error Log Format.
4. Controller Error Error Log Format.

EVEN PARITY

Even parity refers to the number of ones in a data field. If the number of ones is even, parity is even. See also, ODD PARITY.

EXCHANGE

A pair of control messages. The first control message in an exchange is a command issued by the controller. The second control message in an exchange is a response sent by the drive.

EXTENDED STATUS

The additional set of status information maintained by the drive that is of interest to a host error log. Extended status is drive-type specific, and is not utilized by the controller except as input to the host error log and diagnostic processes.

EXTENSION CYCLE

A bus cycle during which a VAXBI transaction is "extended." See STALL DATA CYCLE, BUSY EXTENSION CYCLE, and LOOPBACK EXTENSION CYCLE.

EXTERNAL BLOCK NUMBER (XBN)

See BLOCK

FORCED ERROR

A forced error indicates data in the block is possibly invalid. It does not indicate the block is unreliable. For example, if an unrecoverable ECC error occurs when data is copied from one block to another block, the data is questionable. Because of the questionable data, the block that receives the data will contain a forced error. However, the receiving block itself is not unreliable.

FORCED ERROR INDICATOR

The logical flag, present in each disk block, used to record the presence of a Forced Error. Depending upon the detailed, low level format of the disk device, this may be implemented either as an actual bit flag or as a special pattern (such as the complement of the normal value) of error-correcting and/or error-detecting codes.

FRAME

Frame refers to serial data streams. For the SDI: a 16-bit quantity. It is the smallest unit of control information, command information, or response information passed between the controller and the disk drive by the interface hardware.

GENERIC STATUS

A subset of the status information maintained by the drive that is independent of drive type. It provides the basic information necessary for normal drive operation.

GROUP

See DISK GEOMETRY

H

Designates a high-voltage logic level (that is, the logic level closest to Vcc). Contrast with L. See also, BIT.

HOST

The central processing unit connected to the controller.

IDENT ARB CYCLE

The fourth cycle of an IDENT transaction during which nodes arbitrate to determine which is to send the vector.

ILLEGAL CONFIRMATION CODE

A confirmation (CNF) code that is not permitted in a particular VAXBI cycle (such as a RETRY command confirmation to a multi-responder command).

IMBEDDED ARBITRATION CYCLE

An arbitration cycle that occurs (is imbedded) in a VAXBI transaction.

IMMEDIATE COMMANDS

Commands that MSCP servers should execute immediately, without waiting for any other commands to complete. Immediate commands are typically status inquiries and must be completed within the controller timeout interval.

INTERLOCK COMMANDS

The two commands, IRCI (Interlock Read with Cache Intent) and UWMCI (Unlock Write Mask with Cache Intent), that are used to implement indivisible read-modify-write operations.

INTERNODE TRANSFER

A VAXBI transaction in which the master and slave(s) are in different VAXBI nodes. Contrast with INTRANODE TRANSFER.

INTERRUPT PORT

Those BCI signals that are used in generating INTR transactions.

INTERRUPT PORT INTERFACE

That portion of KDB5Ø logic used to interface to the interrupt port of the BIIC.

INTERRUPT SUBLEVEL PRIORITY

Interrupt priority information used during an IDENT transaction to determine which node with a pending interrupt is to provide the vector. The interrupt sublevel priority corresponds to the node ID.

INTERRUPT VECTOR

In VAX/VMS systems, an unsigned binary number used as an offset into the system control block. The system control block entry pointed to by the VAXBI interrupt vector contains the starting address of an interrupt handling routine. (The system control block is defined in the VAX-11 Architecture Reference Manual.)

INTRANODE TRANSFER

A transaction in which the master and slave are in the same node. Loopback transactions are intranode transfers. Contrast with INTERNODE TRANSFER.

L

Designates a low-voltage logic level (that is, the logic level closest to ground). Contrast with H. See also, BIT.

LATENCY

See BUS ACCESS LATENCY; READ ACCESS TIME.

LEVEL Ø, LEVEL 1, LEVEL 2

Refers to layers of the Standard Disk Interface (SDI).

- Level Ø -- Level Ø is the electrical interface.
- Level 1 -- Level 1 refers to the Real-Time SDI operations. These operations include the protocol associated with the Real-Time Controller State (RTCS) line and the Real-Time Drive State (RTDS) line. It also refers to level 1 real-time data commands, such as Select Group. These commands do not have the start-continue-end frame format associated with the level 2 commands. Level 1 commands also do not require a response from the drive.
- Level 2 -- Level 2 refers to the SDI command and response protocol. Level 2 commands require the start-continue-end frame format. Also, for each level 2 command the drive receives, the drive must respond with a level 2 response.

LOCAL MEMORY

VAXBI memory that can be accessed without using VAXBI transactions; for example, VAXBI-accessible memory on a single board computer.

LOGICAL BLOCK NUMBER (LBN)

See DISK GEOMETRY

LOOPBACK EXTENSION CYCLE

A cycle of a loopback transaction during which a node asserts both BI BSY L and BI NO ARB L to delay the start of the next transaction.

LOOPBACK REQUEST

A request from the master port interface asserted on the BCI RQ<1:Ø> L lines which permits intranode transfers to be performed without using the VAXBI bus.

LOOPBACK TRANSACTION

A transaction in which information is transferred within a given node without use of the VAXBI data path. Contrast with VAXBI TRANSACTION.

MASS STORAGE CONTROL PROTOCOL (MSCP)

A set of messages that allows the host to communicate with intelligent controllers.

MAPPED ADAPTER

A DMA adapter that performs data transfers between a system with a contiguous memory space and VAXBI address space (in which memory need not be contiguous). The mapping is done by using a set of map registers located in the adapter.

MASTER

The node that gains control of the VAXBI bus and initiates a VAXBI or loopback transaction. See also PENDING MASTER.

MASTER PORT

Those BCI signals used to generate VAXBI or loopback transactions.

MASTER PORT INTERFACE

That portion of user logic that interfaces to the master port of the BIIC.

MASTER PORT REQUEST

A request (either VAXBI or loopback) generated by the master port interface through the use of the BCI RQ<1:Ø> L lines.

MASTER PORT TRANSACTION

Any transaction initiated as a result of a master port request.

MESSAGE

A non-real time exchange of frames between a controller and disk drive. It is comprised of a command (which the controller sends to the disk drive) and its response (which the drive sends to the controller). See also, LEVEL Ø, LEVEL 1, LEVEL 2.

MODE BYTE

One of the status bytes in the generic status. It is used to store the current state of drive-operating modes that the controller can alter.

MSCP SERVER

Server that processes host MSCP commands and sends responses to host commands back to the issuing class driver.

MULTI-RESPONDER COMMANDS

VAXBI commands that allow for more than one responder. These include the INTR, IPINTR, STOP, INVAL, and BDCST commands.

MULTI-UNIT DRIVES

A single drive attached to the controller via a single SDI cable which has media divided into multiple independent subunits, each subunit representing a distinct logical unit to the host. The SDI limits the number of multiple subunits to four.

NODE

A VAXBI device that occupies one of sixteen logical locations on a VAXBI bus. A VAXBI node consists of one or more BI modules.

NODE ID

A number that identifies a VAXBI node. The source of the node ID is an ID plug attached to the backplane.

NODE RESET

A sequence that causes an individual node to be initialized; initiated by setting the start self test (SST) bit in the BI Control and Status Register.

NODESPACE

An 8-Kbyte block of I/O addresses that is allocated to each node. Each node has a unique nodespace based on its node ID.

NULL CYCLE

A cycle in which all VAXBI lines are deasserted (that is, no transaction or arbitration is taking place).

NON-PRIMARY REPLACEMENT BLOCK

Refer to BAD BLOCK REVECTORING

NON-SEQUENTIAL COMMANDS

Commands whose execution order MSCP servers may rearrange in order to optimize performance. The optimization may not move a non-sequential command past the barrier imposed by a sequential command.

ODD PARITY

Odd parity refers to the number of ones in a data field. If the number of ones is odd, parity is odd.

Pertaining to VAXBI parity: BI PØ L is either true or false based on the number of asserted bits in the data field. If the number of asserted bits is even (even parity), BI PØ L asserts to generate odd parity. If the number of asserted bits is odd (odd parity), BI PØ L deasserts so odd parity remains.

PARITY MODE

Specifies whether parity is generated by the BIIC or by the user interface.

PENDING MASTER

A node that has won an arbitration but which has not yet begun a transaction.

PENDING REQUEST

A request of any type, whether from the master port or a BIIC-generated request, that has not yet resulted in a transaction.

PHYSICAL BLOCK NUMBER (PBN)

See BLOCK

PIPELINE REQUEST

A request from the master port that is asserted prior to the deassertion of BCI RAK L for the present master port transaction; that is, a new request is posted prior to the completion of the previous transaction.

PORT DRIVER

A software message handler that passes MSCP messages between the class driver and a specific controller.

POWER-DOWN/POWER-UP SEQUENCE

The sequencing of the BI AC LO L and BI DC LO L lines upon the loss and restoration of power to a VAXBI system. See also SYSTEM RESET.

PRIMARY REPLACEMENT BLOCK

See BAD BLOCK REVECTORING

PRIVATE MEMORY

Memory that cannot be accessed from the VAXBI bus.

PROGRAMMED I/O (PIO) ADAPTER

An adapter that does not access memory on the VAXBI bus but interacts only with a host processor.

PROTOCOL

See COMMUNICATIONS PROTOCOL

READ ACCESS TIME

The delay from the time a node requests read data until it receives that data from the VAXBI bus.

RCLK (RECEIVE CLOCK)

The clock phase during which information is received from the VAXBI bus.

READ DATA

Relative to a particular device, read data is data it receives. The term read data may also indicate data read from the disk surface, relative to any device.

READ DATA CYCLE

A data cycle in which data is transmitted from a slave to a master.

READ-TYPE COMMANDS

Any of the various VAXBI read commands, including READ, RCI (Read with Cache Intent), and IRCI (Interlock Read with Cache Intent).

REAL-TIME COMMANDS

Level-one frames sent from the controller to the drive.

REINITIALIZATION

Resetting all devices in a system to a known state. Intelligent subsystems generally execute minimum integrity diagnostics before entering the proper idle state.

REJECTED COMMAND

A command that the MSCP server rejects, discards, aborts, or otherwise finishes before it begins the command execution.

REPLACEMENT BLOCK NUMBER (RBN)

See BLOCK

REQUEST BYTE

One of the status bytes in the generic status. It is used to signal requests from the drive for controller action.

RESERVED CODE

A code reserved for use by DIGITAL.

RESERVED FIELD

A field reserved for use by DIGITAL. The node driving the bus must ensure that all VAXBI lines in the RESERVED field are deasserted. Nodes receiving VAXBI data must ignore RESERVED field information. This requirement provides for adding functionality to future VAXBI node designs without affecting compatibility with present designs. Example: The BI D<31:Ø> L and BI I<3:Ø> L lines during the third cycle of an INTR transaction are RESERVED fields.

RESET MODULE

In a VAXBI system, the logic that monitors the BI RESET L line and any battery backup voltages and that initiates the system reset sequence.

RESETTING NODE

The node that asserts the BI RESET L line.

RESPONSE TIMER

Implemented by the controller as a way to detect a non-functioning disk drive.

RETRY STATE

A state that the BIIC enters upon receipt of a RETRY confirmation code from a slave. If the master reasserts the transaction request, the BIIC resends the transaction without having the user interface provide the transaction information again. The command/address information and the first data longword, if a write transaction, are stored in buffers in the BIIC.

SDI-CONTROLLER

A controller that attaches to its drives via the SDI interconnect.

SECTOR

See BLOCK

SEQUENTIAL COMMANDS

Commands that MSCP servers must execute in the exact order in which they are received from class drivers. Sequential commands typically change a unit's state or context.

SINGLE-RESPONDER COMMANDS

VAXBI commands that allow for only one responder. These include read- and write-type commands and the IDENT command. Although multiple nodes can be selected by an IDENT, only one returns a vector.

SLAVE

A node that responds to a transaction initiated by a node that has gained control of the VAXBI bus (the master).

SLAVE PORT

Those BCI signals used to respond to VAXBI and loopback transactions.

SLAVE PORT INTERFACE

That portion of user logic that interfaces to the slave port of the BIIC.

SPECIAL COMMANDS

Commands that have both the execution order constraints of non-sequential commands, plus certain special, command-dependent execution order constraints.

STALL DATA CYCLE

A data cycle of a read- or write-type transaction during which the slave asserts the STALL confirmation (CNF) code to delay the transmission of the next data word.

STANDARD DISK INTERFACE (SDI) BUS

The SDI bus is a four-signal radial bus that uses messages and protocol to communicate with the disk drives.

STATE BIT TIME

See DATA BIT TIME

SUMMARY STATUS

This is a partial set of the available status.

SYNC CHARACTER

Can be any recognizable character (bit) or series of characters in a serial data stream that synchronizes the receiver circuitry so that meaningful data is received.

For the SDI: The sync character is a 12-bit binary pattern that identifies the start of meaningful data on the SDI read/response or write/command data lines. The sync character itself is: 1111Ø1Ø11ØØ1 (with time=zero on the left). It is is preceded by an undefined number of leading zeroes, and followed immediately by two zeroes. Including the leading and trailing zeros, the sync pattern looks like this (time=zero on the left): ØØ1111Ø1Ø11ØØ1ØØ (3D64 Hex). Meaningful data immediately follows this pattern. Sometimes the sync character is referred to as 26BC Hex. 26BC is the same pattern as 3D64 except the time reference (time=zero) is on the right instead of on the left.

SYSTEM RESET

An emulation of the power-down/power-up sequence that causes all nodes to initialize themselves; initiated by the assertion of the BI RESET L line. See Also, REINITIALIZATION

TARGET BUS

The bus that a VAXBI node interfaces to the VAXBI bus.

TCLK (TRANSMIT CLOCK)

The clock phase during which information is transmitted on the VAXBI bus.

TERMINATED COMMAND

A command that the MSCP server terminates after it has been partially executed.

TRACK

See DISK GEOMETRY

TRANSACTION

The execution of a VAXBI command. The term "transaction" includes both VAXBI and loopback transactions.

UNDEFINED FIELD

A field that must be ignored by the receiving node(s). There are no restrictions on the data pattern for the node driving the VAXBI bus. Example: The BI D<31:Ø> L and BI I<3:Ø> L lines during read STALL data cycles and vector STALL data cycles are UNDEFINED fields.

USER INTERFACE

All node logic exclusive of the BIIC.

USER INTERFACE CSR SPACE

That portion of each nodespace allocated for user interface registers. The user interface CSR space is the 8-Kbyte nodespace minus the lowest 256 bytes which comprise the BIIC CSR space.

USER INTERFACE REQUEST

A transaction request from the user interface which can take the form of a master port request, an assertion of a BCI INT<7:4> L line, or the setting of a force bit.

VAX INTERRUPT PRIORITY LEVEL (IPL)

In VAX/VMS systems, a number between Ø and 31 that indicates the priority level of an interrupt with 31 being the highest priority. When a processor is executing at a particular level, it accepts only interrupts at a higher level, and on acceptance starts executing at that higher level.

VAX PORT ADAPTER

In a VAXBI system, an adapter that conforms to the VAX port architecture, uses interlock transactions to access command and response queues in VAXBI memory, and performs virtual-to-physical memory translation by using page tables located in memory on the VAXBI bus.

VAXBI PRIMARY INTERFACE

The portion of a node that provides the electrical connection to the VAXBI signal lines and implements the VAXBI protocol; for example, the BIIC.

VAXBI REQUEST

A request for a VAXBI transaction from the master port interface that is asserted on the BCI RQ<1:Ø> L lines.

VAXBI SYSTEM

All VAXBI cages, BI modules, reset modules, and power supplies that are required to operate a VAXBI bus. A VAXBI system can be a subsystem of a larger computer system.

VAXBI TRANSACTION

A transaction in which information is transmitted on the VAXBI signal lines. Contrast with loopback transaction.

VECTOR DATA CYCLE

A data cycle in which an interrupt vector is transmitted from a slave to a master.

WINDOW ADAPTER

A bus adapter that maps a contiguous portion of address space (window) from one bus to a similar, but not necessarily the same, portion of address space on another bus.

WINDOW SPACE

A 256-Kbyte block of I/O addresses allocated to each node based on node ID and used by bus adapters to map VAXBI transactions to other buses.

WRITE-TYPE COMMAND

Any of the various VAXBI write commands, including WRITE, WCI (Write with Cache Intent), WMCI (Write Mask with Cache Intent), and UWMCI (Unlock Write Mask with Cache Intent).

WRITE DATA

Relative to a particular device, write data is transmitted data.

INDEX

Digital Equipment Corporation • Colorado Springs, CO 80919