DECsystem 5500 Technical Manual

Order Number EK-D5500-TM-001

Digital Equipment Corporation

January 1991

This document was prepared and published by Educational Services Development and Publishing, Digital Equipment Corporation.

Contents

4 Firmware Specification

5 System Specification

A Q22-bus Specification

B Specifications

C System Block Diagram

D Prom Entry Points

Examples

Figures

Tables

About This Manual

The *DECsystem 5500 Technical Manual* contains the functional, physical and environmental characteristics of the KN220-AA CPU module set, and includes information on the MS220-Ax memory expansion modules.

Intended Audience

This manual is intended for a customer, a design engineer, or applications programmer who is familiar with Digital's extended LSI-11 bus (Q22-bus) and the VAX instruction set. This manual should be used along with the *VAX Architecture Reference Manual* as a programmer's reference to the module.

Organization

The manual is divided into five chapters and nine appendices.

Chapter	Description
1	Introduces the KN220-AA CPU module, the MS220-Ax memory module, and the KN220-AA I/O module, including module features and specifications.
2	Contains procedures for installing and configuring the CPU, the memory, and I/O module in the Q22-bus backplanes and system enclosures.
3	Describes the KN220-AA registers, instruction set, and memory.
4	Describes the entry/power up, boot diagnostics, device booting sequence, console program, and console commands.
5	Describes the functional, electrical, physical, environmental, and performance characteristics of the 220Qn family of the DECsystem 5500 upgrade kits, system building blocks, kernels, and supported options.
Appendix A	Describes the low-end member of Digital's bus family. All of Digital's microcomputers use the Q22-bus, such as the MicroVAX I, MicroVAX II, MicroVAX 3500, MicroVAX 3600, MicroPDP–11.
Appendix B	Describes the physical and environmental characteristics of the KN220-AA CPU module.
Appendix C	Contains a system block diagram.
Appendix D	Contains a list of entry points that are defined in the KN220-AA PROM.
Appendix E	Lists the devices from where the KN220-AA firmware supports Operating System bootstrapping.
Appendix F	Contains exit and maintenance command guidelines.

Chapter	Description
Appendix G	Contains reference information on the CVAX CPU.
Appendix H	Lists the KN220-AA R3000 address assignments and M7638-Ax R3000 physical I/O address space map.
Appendix I	Contains an overview, operation, and firmware commands on Prestoserve.

Related Documents

Below is a list of related documents.

- DECsystem 5500 Installation, EK-331AA-IN
- DECsystem 5500 System Operation, EK-332AA-OP
- DECsystem 5500 Cabinet System Operation, EK-334AA-OP
- DECsystem 5500 Troubleshooting and Diagnostics, EK-425AB-TS
- DECsystem 5500 Technical Information, EK-333AA-TI
- KN220 CPU System Manual, EK-375AA-SM

You can order these documents from Digital Equipment Corporation, at the address given below.

Digital Equipment Corporation
Accessories and Supplies Group
P.O Box CS2008
Nashua, NH 03061

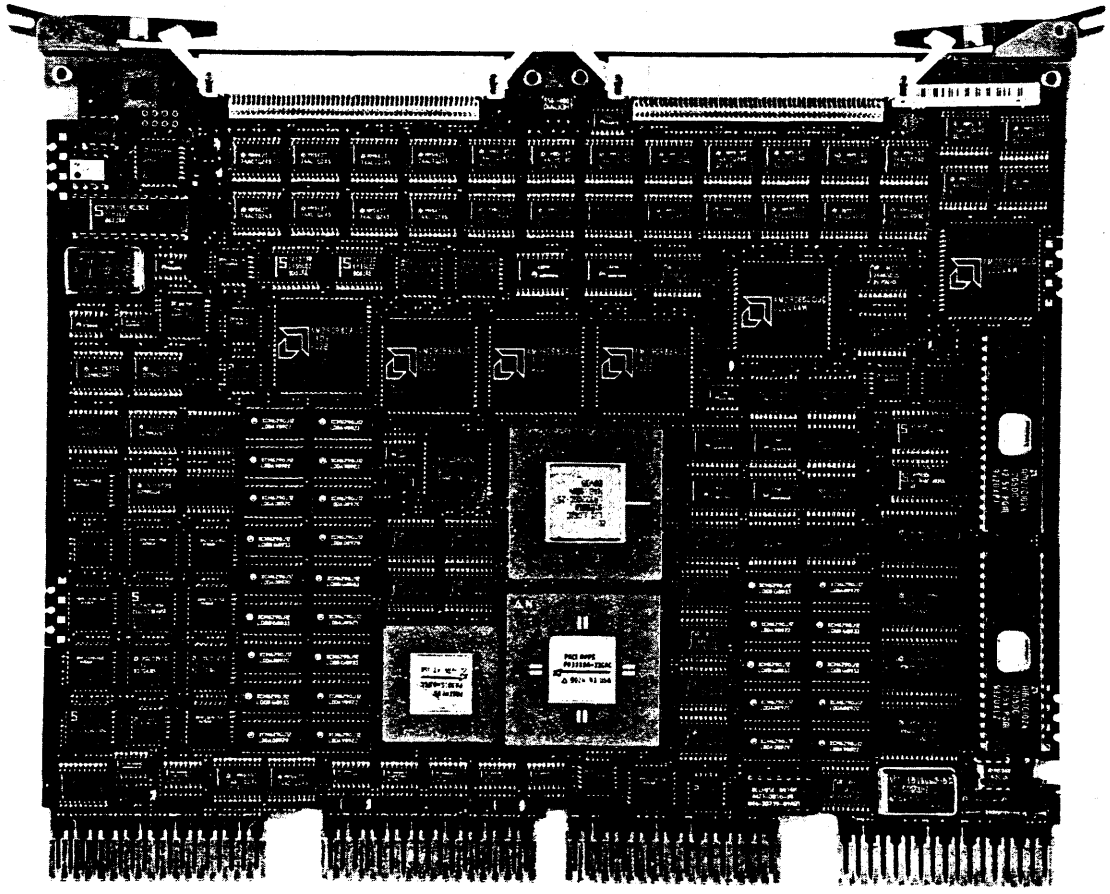
Attention: Documentation Products

1
KN220-AA Subsystem Overview

This chapter provides a brief overview of the KN220-AA module set and MS220-Ax memory modules.

1.1 Introduction

The KN220-AA CPU module set consists of the KN220-AA processor module and the KN220-AA I/O module. Both are quad-height modules for the Q22-bus (extended LSI-11 bus). The KN220-AA processor module features the R3000 RISC processor. The KN220-AA CPU module set is designed for high speed, multiuser, and multitasking functions. It incorporates a two-level cache to maximize system performance. Estimated compute performance for the KN220-AA CPU (Figure 1–1) is 20 times that of an 11/780 MIPS.



LJ-00426-TI0

Figure 1-1 KN220-AA CPU Module

The KN220-AA I/O module (Figure 1–2) features built-in DSSI, SCSI, and Ethernet controllers.

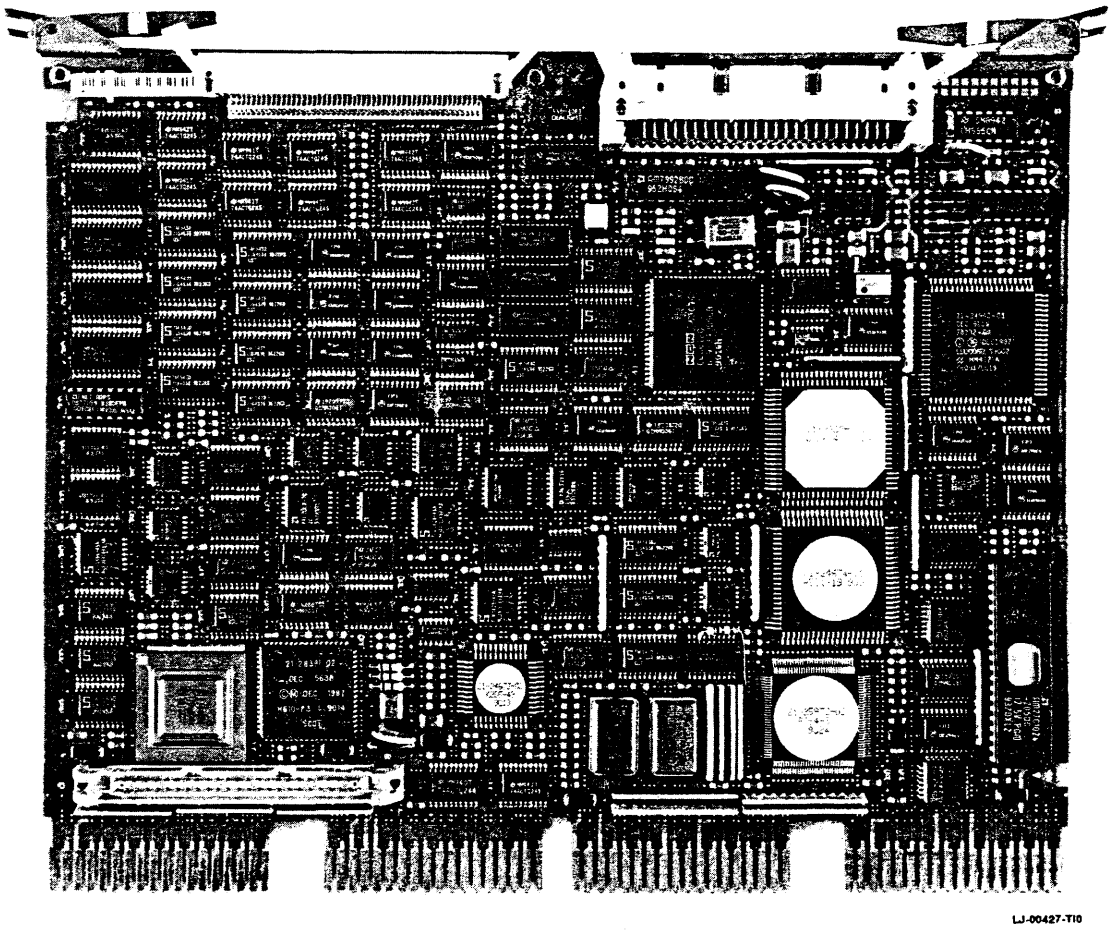


Figure 1–2 KN220-AA I/O Module

The KN220-AA CPU/I/O module subsystem is divided into four major areas:

- Central processing subsystem
- System support subsystem
- I/O subsystem
- Main memory controller

The KN220-AA CPU module set is used in the DECsystem 5500. The DECsystem 5500 utilizes a BA430 enclosure. There are no jumpers or switches to configure.

The KN220-AA uses a 50-pin ribbon cable to communicate with the console device through the H3602-AC CPU cover panel, which also contains configuration switches, Ethernet connector, and an LED display.

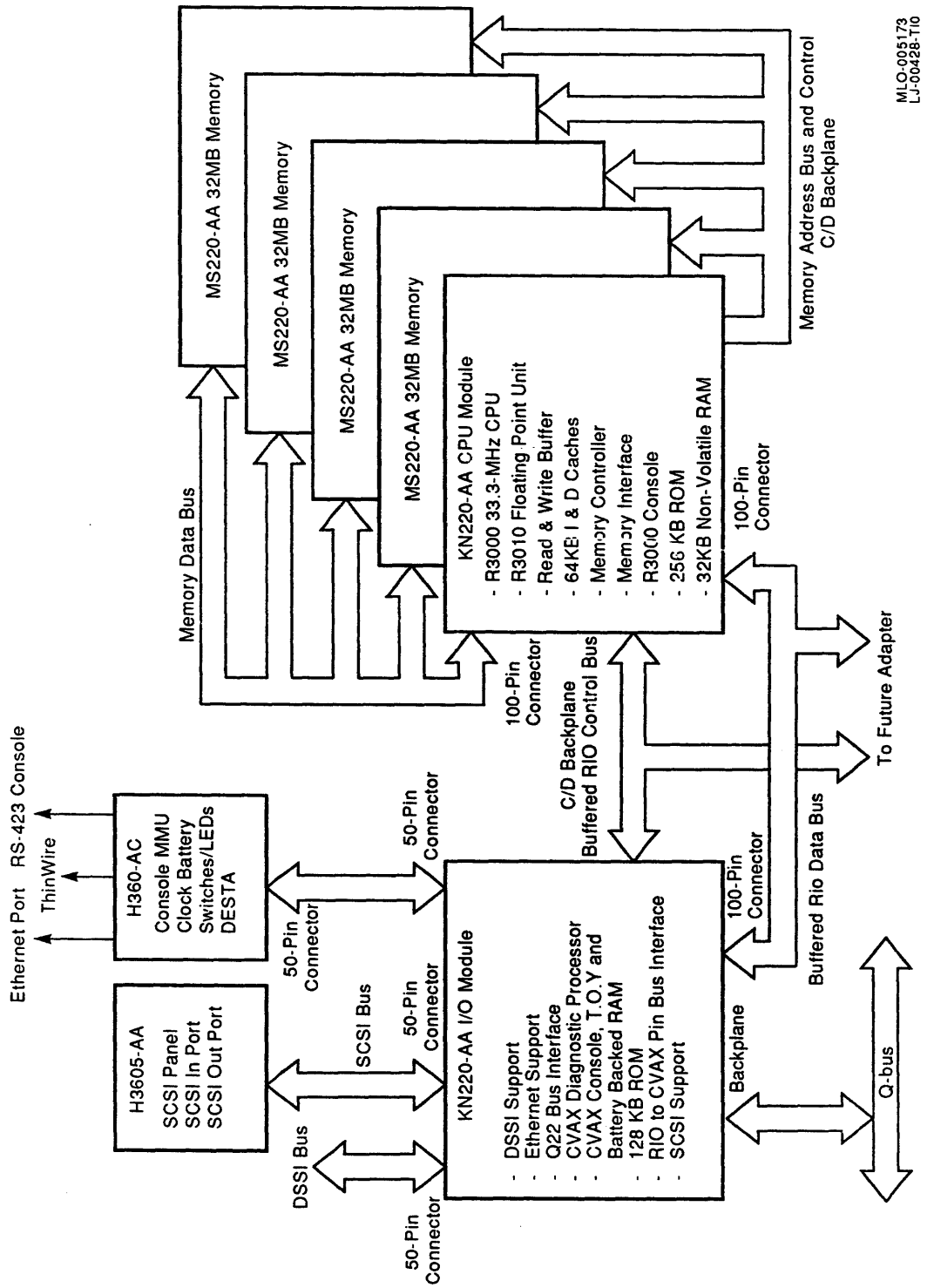


Figure 1–3 DECsystem 5500 Block Diagram

The KN220-AA CPU module set and MS220-Ax memory modules combined form a RISC CPU/memory subsystem. The RISC CPU/memory subsystem uses the Q22-bus, DSSI bus, SCSI bus, and Ethernet to communicate with mass storage and I/O devices. The KN220-AA CPU module set and MS220-Ax modules are mounted in standard Q22-bus backplane slots that implement the Q22-bus in the AB rows and the CD interconnect in the CD rows. The KN220-AA CPU module set can support up to four MS220-Ax modules, if enough Q22/CD slots are available. Figure 1–3 shows the system block diagram.

Figure 1–4 shows the CPU module layout. Figure 1–5 shows the I/O module layout.

The major hardware components of the KN220-AA CPU and I/O modules are as follows:

- R3000 Central processor (CPU) with a 30 MHz clock
- R3010 Floating point accelerator (FPU)
- DC521: Clock
- DC527 (CQBIC): Q22-bus interface
- DC541 (SGEC): Ethernet interface
- DC511 (SSC): System support chip
- DC509: Clock
- Firmware ROMs (2): 256 Kbytes; each 128 Kbytes by 8
- SII DSSI interface chips
- SCSI interface chip
- 50-pin connector to the H3602-AC CPU cover panel
- 100-pin connector to the RIO bus carrying signals for the RIO bus and the memory interconnect
- NVRAM and battery backup (can be enabled/disabled through the jumper) for Prestoserve.

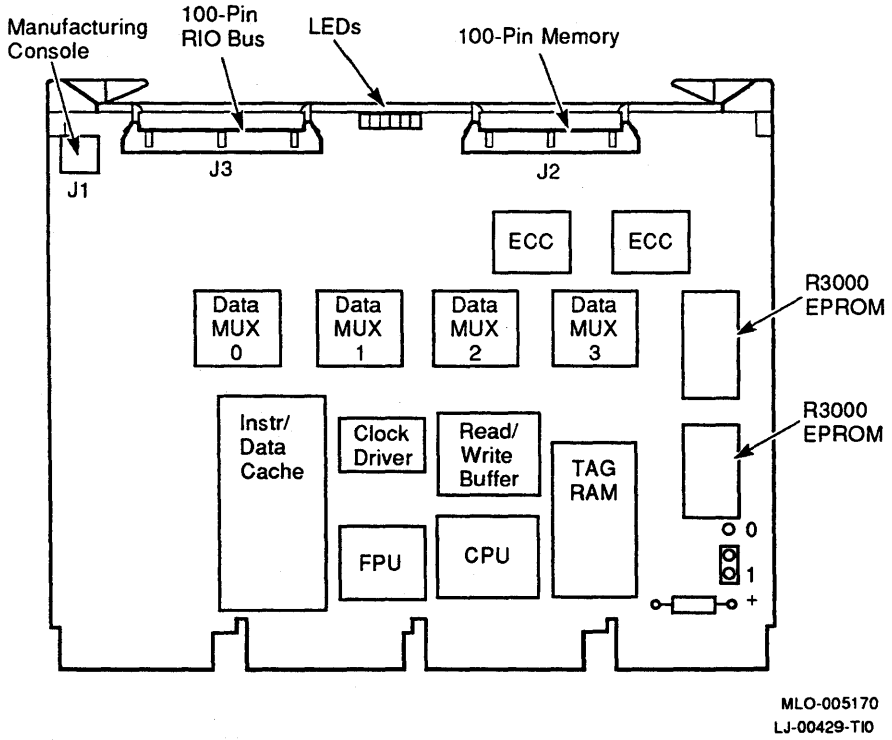


Figure 1–4 KN220-AA CPU Module Layout

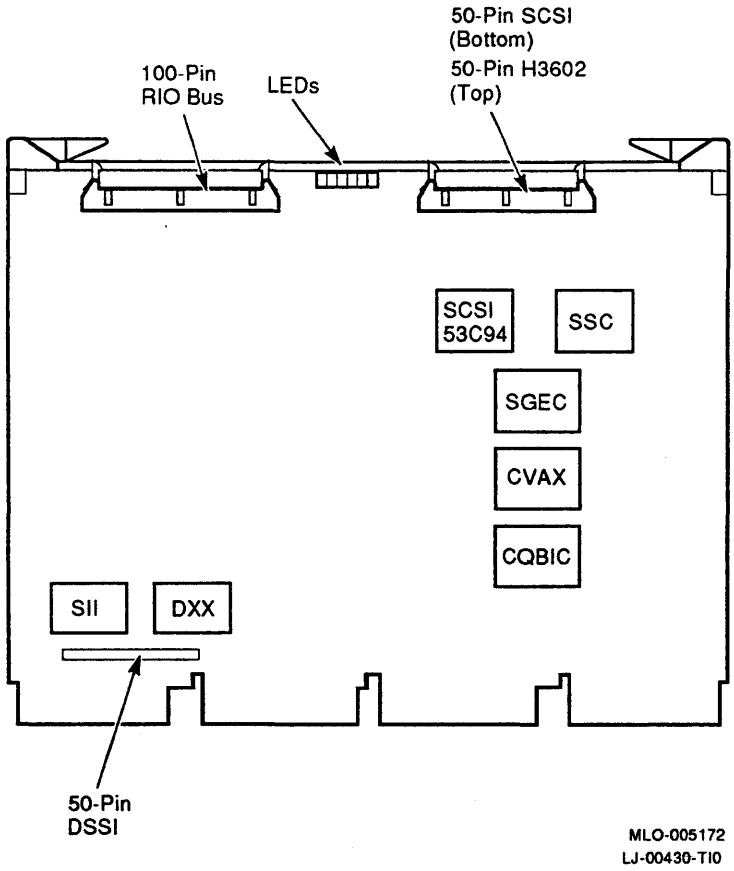


Figure 1–5 KN220-AA I/O Module Layout

1.2 Central Processing Subsystem

The central processing subsystem contains a CPU chip, a floating point accelerator chip (FPU), the cache RAMs, write buffer chip, and a memory controller.

1.3 R3000 RISC Processor

The R3000 RISC processor chip is the central processor of the KN220-AA module set. The R3000 chip implements two tightly coupled processors in a single VLSI chip. One processor is a 32-bit CPU, and the second is a system control processor (CP0). The R3000 RISC processor contains the following:

- 32-bit operation
- A 5-stage pipeline
- cache control
- memory management
- Coprocessor interface

1.4 Floating-Point Accelerator

The floating-point accelerator is implemented by the R3010 floating-point accelerator (FPA) chip. The R3010 FPA operates as a coprocessor for the R3000 processor and extends the R3000's instruction set to perform arithmetic operations on values in floating-point representations. The R3010 FPA interfaces with the R3000 processor to form a tightly-coupled unit with seamless integration of floating-point and fixed-point instruction sets.

1.5 Cache Memory

To maximize CPU performance, the KN220-AA module utilizes cache memory. The cache memory is organized as two separate 64 Kbyte caches; one for instructions, and the other for data.

1.6 MS220-Ax Memory Modules

The MS220-Ax memory module (Figure 1–6) supports two 39-bit paths (32 data bits and 7 ECC bits) resulting in a 78-bit data flow between the MS220-Ax and the memory controller. The two 32-bit data paths are protected by 7 ECC bits each with single bit correction/double error detection capability.

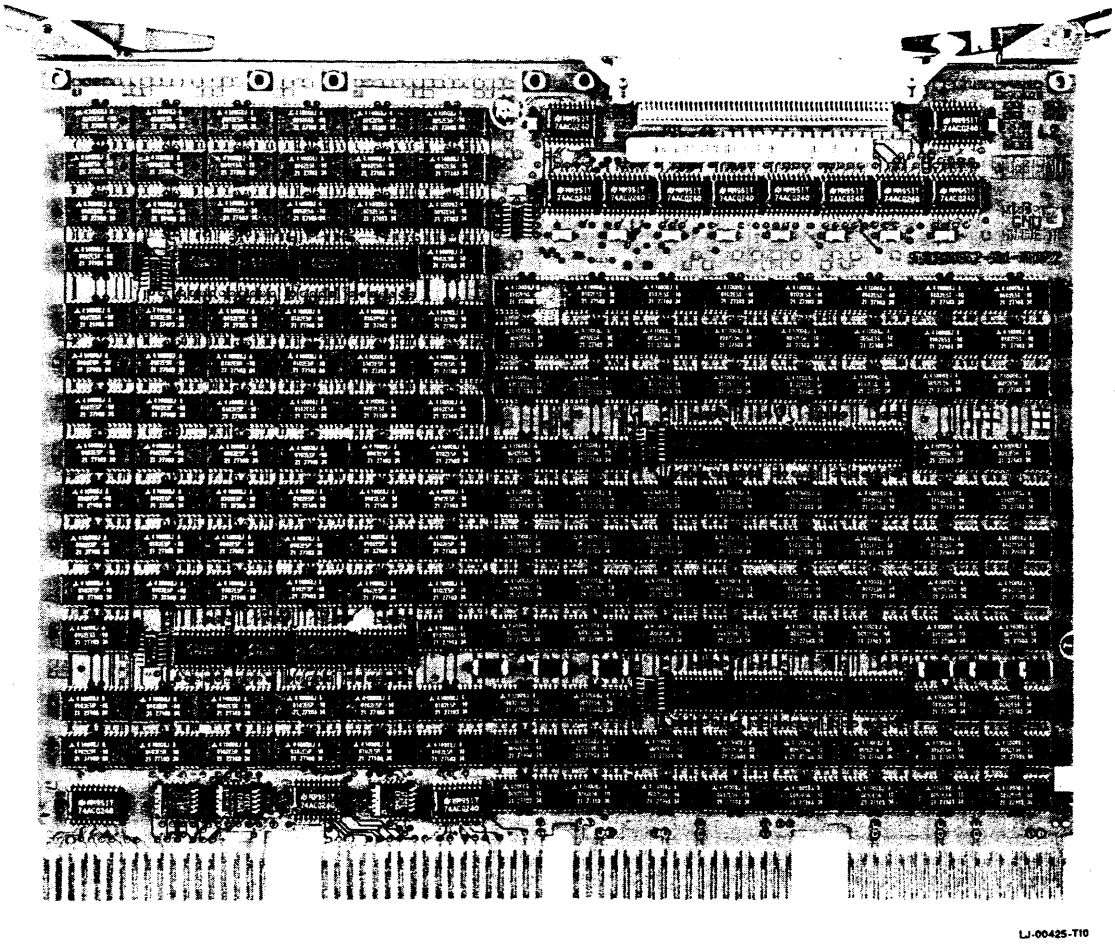
The MS220-Ax is a "non-intelligent" memory array that is completely controlled by a memory controller on the CPU module. All DRAM timing and control signals (Ras_time, RAS, CAS, WE) address signals, address multiplexing, data, and data transceiver control signals, and so on, originate from the memory controller logic on the CPU module. The MS220-Ax merely provides the DRAM array, control/address line buffers/drivers.

1.7 Mass Storage Interfaces DSSI/SCSI

Both SCSI and DSSI mass storage interfaces reside on the KN220-AA I/O module. Up to seven SCSI ISEs and seven DSSI ISEs can be connected to these interfaces (seven devices on each bus).

The DSSI bus interface is implemented through the SII chip and four 32Kb x 8 static RAMs (128 Kbytes of DSSI buffer RAM). Data is transferred between the DSSI bus and main memory through this DSSI buffer RAM. Sixteen registers are used in the DSSI interface. The data path is eight bits wide.

The SCSI bus interface is implemented through a SCSI controller, four 32Kb x 8 static RAMs (128 Kbytes of SCSI buffer RAM) and the RIO interface/DMA controller. The buffer is accessed using the 16-bit DMA path. The processor uses the 32-bit RIO bus to load the buffer before starting write transfers (or unload the buffer after completion of SCSI reads). The SCSI data path is eight bits wide.



LJ-00425-TI0

Figure 1–6 MS220-Ax Memory Module

1.8 Ethernet Interface

The KN220-AA I/O module features an on-board network interface. When used in conjunction with the H3602-AC CPU cover panel, this interface allows the KN220-AA CPU module set to be connected to either a ThinWire or standard Ethernet network.

The Ethernet interface includes registers for control and status reporting as well as a DMA controller.

1.9 Q22-bus Interface

The Q22-bus interface is implemented by the CQBIC chip, located on the KN220-AA processor module. The CQBIC chip supports up to 16-word block mode transfers between a Q22-bus DMA device and main memory, and up to 2-word block mode transfers between the CPU and Q22-bus devices. The Q22-bus interface contains the following:

- A 16-entry map cache for the eight 192-entry main memory-resident "scatter-gather" map, used for translating 22-bit Q22-bus addresses into 26-bit main memory addresses.
- Interrupt arbitration logic that recognizes Q22-bus interrupt requests BR7-BR4
- Q22-bus termination

1.10 Firmware

The firmware consists of 384 Kbytes of EPROM. There are three 128 Kbyte ROMs. Two ROMs are located on the KN220-AA CPU module, and one on the KN220-AA I/O module. The ROMs provide the following functions:

- Board initialization
- Power up self-testing of the KN220-AA processor and I/O modules, and the MS220-Ax modules
- Emulation of a subset of the VAX standard console (automatic/manual bootstrap, automatic/manual restart, and a simple command language for examining/altering the state of the processor)
- Booting from supported Q22-bus devices, DSSI, or Ethernet
- Multilingual capability
- MOP support
- Capability to operate in both secure and insecure fashion to control system access

2
Installation and Configuration

The chapter discusses the following topics:

- Installing the KN220-AA and MS220-Ax modules
- Configuring the KN220-AA
- KN220-AA connectors

2.1 Installing the KN220-AA and MS220-Ax Memory Modules

The KN220-AA CPU module and the MS220-Ax memory modules may be installed in the five rightmost backplane slots. Note that the KN220-AA modules (I/O and CPU) are installed in backplane slots 1 and 2, and the memory modules are installed in slots 3 through 6.

To install the KN220-AA and MS220-Ax modules:

1. Install the KN220-AA I/O in slot 1 of the Q22-bus/CD backplane.
2. Install the KN220-AA CPU module in slot 2.
3. Install MS220-Ax memory modules in slots 3 through 6.
   - If you only use one memory module , you can install it in any one of the slots 3 through 6.
   - If you use more than one memory module, you must install the first memory module in slot 3, the second in slot 4, and so on. Do not leave a gap between memory modules.
4. Install a 100-pin ribbon cable between the KN220-AA CPU and the console module.

Figure 2–1 shows the BA430 backplane and the position of the KN220-AA CPU, KN220-AA I/O, and MS220-Ax memory modules.

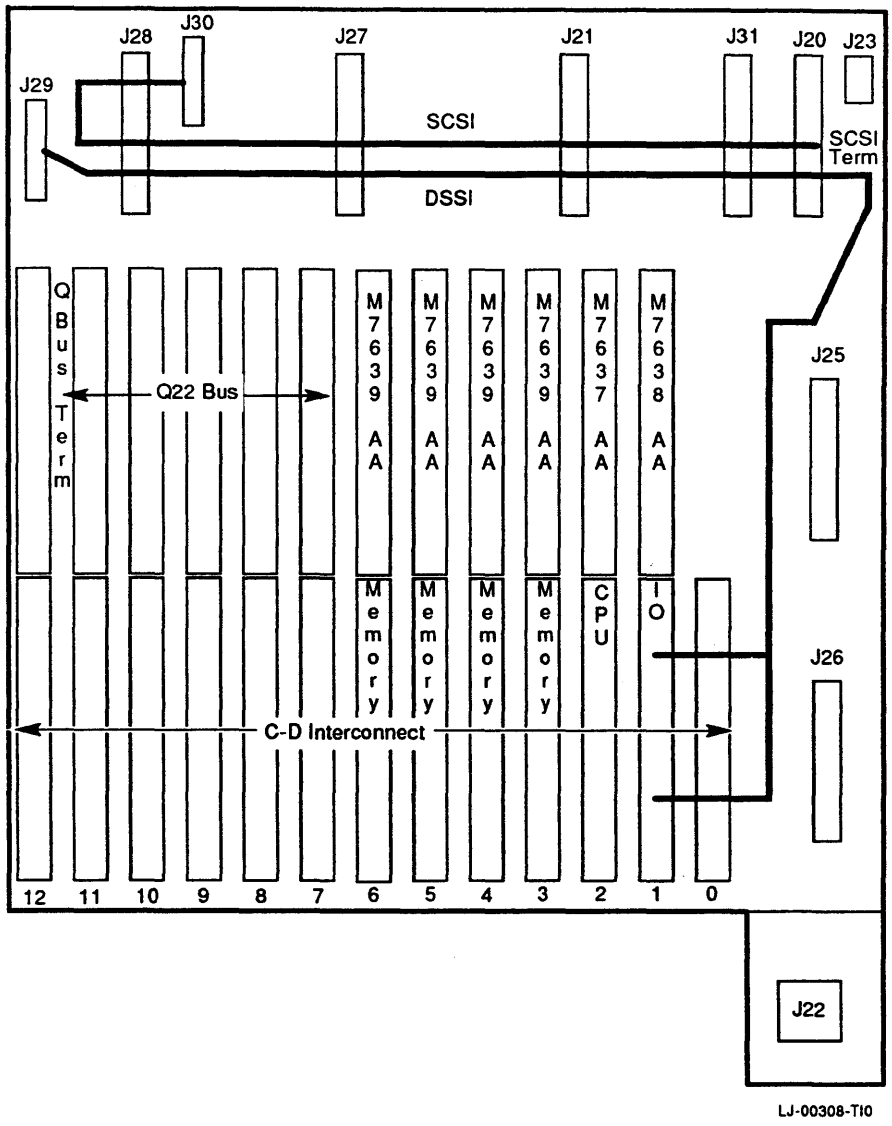


Figure 2–1 BA430 Backplane Layout

Figure 2–2 shows the position of the CPU, I/O, and memory modules in the BA213 backplane and H9644 enclosure used for system upgrades.

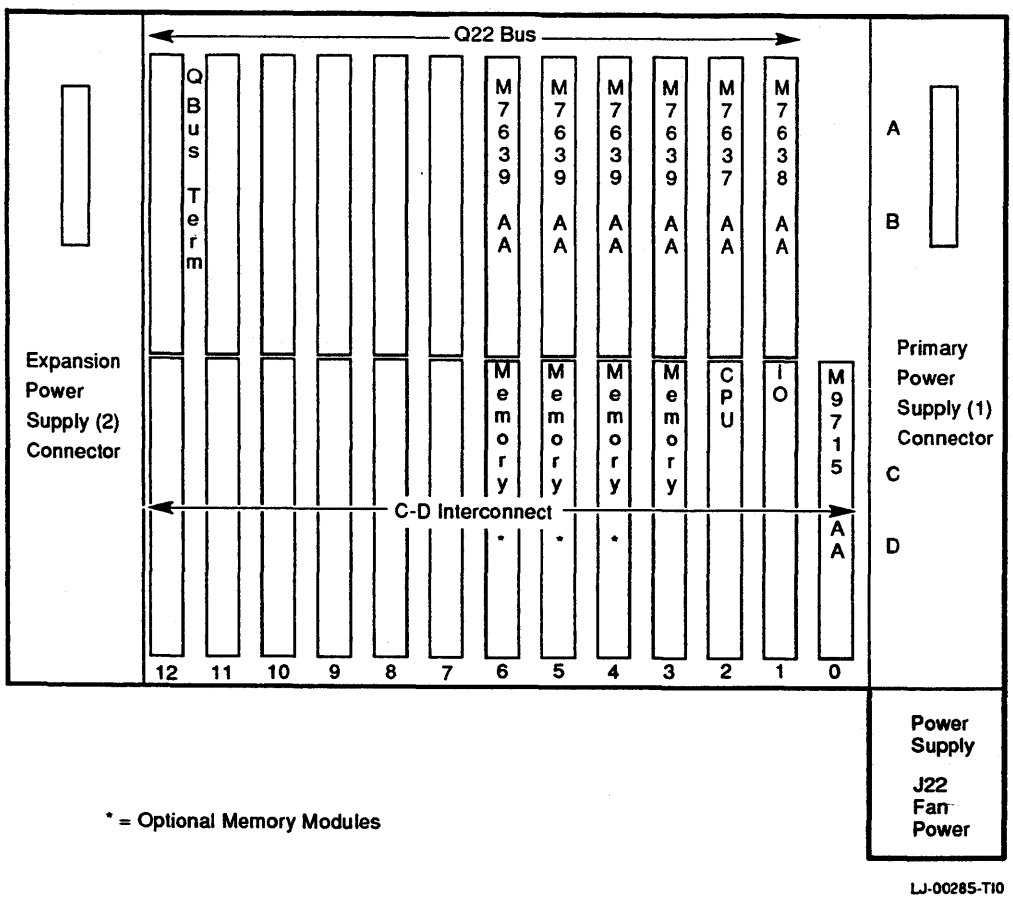


Figure 2–2 BA213 Backplane Layout

2.2 Module Configuration and Naming

Each module in a system must use a unique device address and interrupt vector. The device address is also known as the control and status register (CSR) address. Most modules have switches or jumpers for setting the CSR address and interrupt vector values. The value of a floating address depends on what other modules are housed in the system.

To set the CSR addresses and interrupt vectors for a module:

1. Determine the correct values for the module with the CONFIGURE maintenance command. The Config utility eliminates the need to boot the VMS operating system to determine CSRs and interrupt vectors. Type config at the console prompt (>>>), then type help for the list of supported devices:

```
>>> config
Enter device configuration, HELP, or EXIT
Device, Number? help
Devices:

LPV11       KXJ11       DLV11J   DZQ11    DZV11    DFA01
RLV21       TSV05       RXV21    DRV11W   DRV11B   DPV11
DMV11       DELQA       DEQNA    RQDX3    KDA50    RRD50
RQC25       KXXXX-DISK  TQK50    TQK70    TU81E    RV20
KXXXX-TAPE  KMV11       IEQ11    DHQ11    DHV11    CXA16
CXB16       CXY08       VCB02    QDSS     DRV11J   DRQ3B
VSV21       IBQ01       IDV11A   IDV11B   IDV11C   IDV11D
IAV11A      IAV11B      MIRA     ADQ32    DTC04    DESQA
IGQ11
```

NOTE
If the console is in the normal operating mode (>>), type maint to switch to the maintenance mode as follows:
>>maint
>>>

The LPV11–SA has two sets of CSR addresses and interrupt vectors. To determine the correct values for an LPV11–SA, enter LPV11,2 at the DEVICE prompt for one LPV11–SA, or enter LPV11,4 for two LPV11–SA modules.

2. See the *KN220-AA CPU System Maintenance Manual* (EK-375AA-SM) for switch settings and CSR and interrupt vector jumper settings for supported options.

2.3 Mass Storage Configuration

There is space for up to four mass storage devices—either three integrated storage elements (ISEs) and one TK70 or TLZ04 tape drive, or four ISEs. The ISEs are part of the Digital Small Storage Interconnect (DSSI) or Small Computer System Interface (SCSI).

With the BA430 enclosure, the DSSI bus is part of the backplane. The ISEs are of the RF- or RZ- series, and they plug directly into the backplane to connect to the DSSI or SCSI bus. This eliminates cabling external to the ISE. Each ISE must have its own unique DSSI or SCSI node ID number. The ISE has its node ID address set through a plug (located on the front of the ISE) on the control panel. For SCSI ISE devices, a cable is connected from the H3605-AA bulkhead to a transition connector. This connector brings the SCSI signals to the backplane and then to the SCSI based ISE.

NOTE

With the BA213 enclosure, internal SCSI ISEs may not be used. External SCSI ISEs may be connected through the H3505-AA bulkhead. The bulkhead will be located on the BA213 enclosure after a system upgrade.

You can access local programs in the RF-series ISE through the MicroVAX diagnostic monitor (MDM), or through the VMS operating system (version 5.0) and console I/O mode SET HOST/DUP command. This command creates a virtual terminal connection to the storage device and the designated local program using the diagnostic and utilities protocol (DUP) standard dialog.

2.3.1 Changing the Node Name

Each ISE has a node name that is maintained in EPROM onboard the I/O module. This node name is determined in manufacturing from an algorithm based on the drive serial number. You can change the node name of the DSSI device to something more meaningful by performing the following procedure. In the example, the node name for the ISE at DSSI node address 1 is changed from R3YBNE to DATADISK.

NOTE

To change the DSSI node name, the system must be in the maintenance console mode. The prompt >>> indicates that the system is in the maintenance console mode.

The following display shows how to change the DSSI node name.

```
>>> sho dssi
DSSI Node 0 (MDC)
-DIA0 (RF71)

DSSI Node 1 (R3YBNE)        !The node name for this drive will be
-DIA1 (RF71)                !changed from R3YBNE to DATADISK.

DSSI Node 7 (*)
>>>
>>> set host/dup/dssi 1
Starting DUP server...
Copyright  1988  Digital Equipment Corporation
DRVEXR V1.0  D  5-NOV-1988 15:33:06
DRVTST V1.0  D  5-NOV-1988 15:33:06
HISTRY V1.0  D  5-NOV-1988 15:33:06
ERASE  V1.0  D  5-NOV-1988 15:33:06
PARAMS V1.0  D  5-NOV-1988 15:33:06
DIRECT V1.0  D  5-NOV-1988 15:33:06
End of directory
Task Name? params
Copyright  1988  Digital Equipment Corporation

PARAMS> sho nodename

Parameter     Current           Default         Type     Radix
--------- ---------------  ---------------  --------  -----
NODENAME          R3YBNE            RF71      String  Ascii  B

PARAMS> set nodename datadisk

PARAMS> write                   !This command writes the change
                                !to EEPROM.
Changes require controller initialization, ok? [Y/(N)] y
```

```
Stopping DUP server...
>>> sho dssi
DSSI Node 0 (MDC)
-DIA0 (RF71)

DSSI Node 1 (DATADISK)      !The node name has changed from
-DIA1 (RF71)                !R3YBNE to DATADISK.

DSSI Node 7 (*)
```

2.3.2 Changing the DSSI Unit Number

By default, the ISE drive assigns the disk's unit number to the same value as the DSSI node address for that drive.

The following display shows how to change the unit number of a DSSI device. This example changes the unit number for the RF71 drive at DSSI node address 2 from 1 to 50 (decimal). You must change two parameters: UNITNUM and FORCEUNI. Changing these parameters overrides the default, which assigns the unit number the same value as the node address.

```
>>> sho dssi
DSSI Node 0 (MDC)
-DIA0 (RF71)

DSSI Node 1 (R3QJNE)      !The unit number for this drive will be
-DIA1 (RF71)              !changed from 1 to 50 (DIA1 to DIA50).

DSSI Node 7 (*)
>>>
>>> set host/dup/dssi 1
Starting DUP server...
Copyright  1988  Digital Equipment Corporation
DRVEXR V1.0  D  5-NOV-1988 15:33:06
DRVTST V1.0  D  5-NOV-1988 15:33:06
HISTRY V1.0  D  5-NOV-1988 15:33:06
ERASE  V1.0  D  5-NOV-1988 15:33:06
PARAMS V1.0  D  5-NOV-1988 15:33:06
DIRECT V1.0  D  5-NOV-1988 15:33:06
End of directory

Task Name? params
Copyright 1988  Digital Equipment Corporation

PARAMS> sho unitnum

Parameter     Current           Default         Type      Radix
--------- ---------------  ---------------  --------  -----
UNITNUM               0                0      Word    Dec   U

PARAMS> sho forceuni

Parameter     Current           Default         Type      Radix
--------- ---------------  ---------------  --------  -----
FORCEUNI              1                1   Boolean    0/1   U

PARAMS> set unitnum 50

PARAMS> set forceuni 0

PARAMS> write       !This command writes the changes to EEPROM.

PARAMS> ex
Exiting...

Task Name?
```

```
Stopping DUP server...
>>>
>>>sho dssi
DSSI Node 0 (MDC)
-DIA0 (RF71)

DSSI Node 1 (R3QJNE)          !The unit number has changed
-DIA50 (RF71)                 !and the node ID remains at 1.

DSSI Node 7 (*)
```

3
KN220-AA Architecture

3.1 KN220-AA CPU Module Summary

This chapter documents the functional, physical, and environmental characteristics of the KN220-AA processor module (module number M7637-Ax).

3.2 General Description

The KN220-AA CPU module, KN220-AA I/O module, and MS22(Ax memory modules combine to form a RISC CPU/Memory/I/O subsystem with a private I/O bus (called the RIO bus) and the Q22-bus, both used to communicate with mass storage and I/O devices. The KN220-AA processor, I/O, and MS220-Ax modules are mounted in standard Q22-bus backplane slots that implement the Q22-bus in the AB rows. The CD interconnect is used for auxiliary communication signals among these three modules and other modules. In the BA430 cabinet, side 1 of slot 1 of the CD interconnect will be used for the DSSI interconnect for internal DSSI disk drives. A separate connector mounted on the I/O module is used for the DSSI interconnect in the BA213 cabinet. The KN220-AA I/O module communicates with the KN220-AA CPU module over a separate 100-pin cable and through the Q22-bus CD interconnect.

In normal operation (when both the CPU and I/O boards are installed in a system), the KN220-AA CPU communicates with the console device through the KN220-AA I/O distribution insert or cover panel (H3602-AC), which also contains configuration switches and an LED display.

The KN220-AA CPU consists of two, quad height Q22-bus modules (M7637-Ax and M7638-Ax) that use external and in-house VLSI chips. The KN220-AA CPU provides the following functions:

- R3000 RISC processor with a 33ns cycle time
- R3010 floating point unit
- LSI logic R322x Read/Write buffer for enhanced memory throughput
- 64 Kbytes, 12ns, instruction cache
- 64 Kbytes, 12ns, data cache
- DSSI mass storage interface
- SCSI mass storage interface
- Ethernet interface
- Main memory controller that resides on the KN220-Ax CPU module.

- One to four MS220-Ax memory modules, depending on the system configuration.
- VAX compatible console port whose baud rate can be set through an external switch located on the H3602-AC cover panel
- RIO bus interface that supports DMA transfers from devices on the I/O board and can also support an additional module, which may also contain DMA devices. The RIO interface contains:
  - — DMA arbitration logic for all potential DMA devices
  - — Separate 32-bit address and data buses with associated control signals
- Q22-bus interface that supports up to 16-word, block mode transfers between a Q22-bus DMA device and main memory, and up to 2-word, block mode transfers between the CPU and Q22-bus devices. This interface contains:
  - — A 16-entry map cache for the 8,192 entry, main memory resident "scatter-gather" map, which is used for translating 22-bit Q22-bus addresses into 26-bit main memory addresses.
  - — Interrupt arbitration logic that recognizes Q22-bus interrupt requests BR7-BR4.
  - — 240 Ohm termination for the Q22-bus.
- CVAX diagnostic processor with an 80ns cycle time.
- ROM diagnostics, boot and console code

3.3 KN220-AA I/O Module Summary

The KN220-AA I/O module contains the CVAX diagnostic processor used to diagnose and initialize some of the I/O devices in the system. It holds the VAX compatible console port and the hardware that is needed for the CVAX ROM and MDM diagnostic functions.

The CVAX diagnostic processor is also used for other maintenance functions and utilities. The following are examples of these functions and utilities:

- Configure
- Show device
- Show memory
- Set host/dup

The KN220-AA I/O module provides higher performance mass storage and network interfaces than are available using compatible Q22-bus devices. The mass storage interface consists of a DSSI port capable of a 4MB per second transfer rate that can control up to seven devices on the DSSI bus and a SCSI port capable of a 4MB per second transfer rate. Both ports utilize a 32 bit data path to the R3000 or CVAX processors and each has 128 Kbytes of static RAM for buffer space. The network interface is a Second Generation Ethernet Controller (SGEC).

NOTE
Because the R3000 architecture does not provide for interlocked instructions, Auxiliary mode is not supported for the KN220-AA.

All R3000 and CVAX addresses in this document are physical addresses.

All addresses in this document are hex unless explicitly stated otherwise.

3.4 KN220-AA R3000 RISC Processor

The R3000 RISC processor plus its associated R3010 floating point unit is the central processor of the KN220-AA.

3.4.1 Processor Features

The R3000 chip implements two tightly coupled processors in a single VLSI chip. One processor is a 32-bit CPU, and the other one is a system control processor (CP0).

The combined CPU/CP0 processors provide the following features:

- **32-bit operation** - The R3000 is a 32-bit machine with thirty-two 32-bit registers, all addressing is 32-bits.
- **Pipeline** - The CPU contains a five-stage pipeline capable of executing one instruction per 33ns cycle (best case).
- **On chip cache control** - The R3000 supports separate instruction and data caches of up to 256 Kbytes (DECsystem 5500 supports 64 Kbytes each). Both caches can be accessed in a single CPU cycle.
- **On chip memory management** - A 4-Gbyte virtual address space is mapped with a 64 entry fully associative translation lookaside buffer.
- **Coprocessor interface** - The R3000 provides a tightly coupled coprocessor interface for up to four coprocessors. The first CP0 is contained on the CPU chip and the second CP1 is the floating point unit. CP2 and CP3 are unused in the DECsystem 5500 design.
- **Write buffer** - The DECsystem 5500 design uses an R322x 6-word deep write buffer. All writes pass through this write buffer. A single read latch also resides in this chip and all reads that miss the cache pass through this chip. The device implements a number of features that enhance memory throughput. It is capable of notifying the memory subsystem of potential page mode writes and can identify block reads that request data already present in the write buffer and notify the memory controller accordingly.

3.4.1.1 General Purpose Registers

The R3000 provides thirty-two 32-bit general purpose registers, a 32-bit program counter, and two 32-bit registers used in integer multiply and divide operations.

3.4.1.2 Instruction Set

Figure 3–1 shows the three types of R3000 instructions.

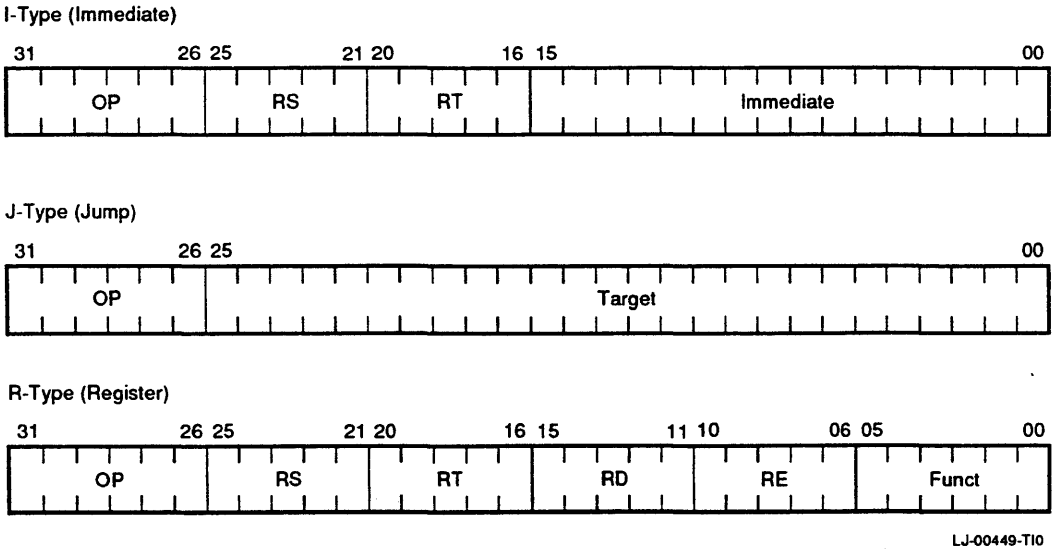


Figure 3–1 R3000 Instructions

3.4.1.2.1 Instruction Summary

OP	Description
Load/Store Instructions	
LB	Load Byte
LBU	Load Byte Unsigned
LH	Load Halfword
LHU	Load Halfword Unsigned
LW	Load Word
LWL	Load Word Left
LWR	Load Word Right
SB	Store Byte
SH	Store Halfword
SW	Store Word
SWL	Store Left
SWR	Store Right
Arithmetic Instructions (ALU Immediate)	
ADDI	Add Immediate

OP	Description
ADDIU	Add Immediate Unsigned
SLTI	Set on Less Than Immediate
SLTIU	Set on Less Than Immediate Unsigned
ANDI	AND Immediate
ORI	OR Immediate
XORI	Exclusive OR Immediate
LUI	Load Upper Immediate
ADD	Add
Arithmetic Instructions (3-operand, register-type)	
ADDU	Add Unsigned
SUB	Subtract
SUBU	Subtract Unsigned
SLT	Set on Less Than
SLTU	Set on Less Than Unsigned
AND	AND
OR	OR
XOR	Exclusive OR
NOR	NOR
Shift Instructions	
SLL	Shift Left Logical
SRL	Shift Right Logical
SRA	Shift Right Arithmetic
SLLV	Shift Left Logical Variable
SRLV	Shift Right Logical Variable
SRAV	Shift Right Arithmetic Variable
Multiply/Divide Instructions	
MULT	Multiply
MULTU	Multiply Unsigned
DIV	Divide
DIVU	Divide Unsigned

OP	Description
MFHI	Move From HI
MTHI	Move To Hi
MFLO	Move From LO
MTLO	Move To LO
Jump and Branch Instructions	
J	Jump
JAL	Jump and Link
JR	Jump to Register
JALR	Jump and Link Register
BEQ	Branch on Equal
BNE	Branch Not Equal
BLEZ	Branch on Less than or Equal to Zero
BGTZ	Branch on Greater Than Zero
BLTZ	Branch on Less than Zero
BGEZ	Branch on Greater than or Equal to Zero
BLTZAL	Branch on Less than Zero and Link
BGEZAL	Branch on Greater than or Equal to Zero and Link
Special Instructions	
SYSCALL	System Call
BREAK	Break
Coprocessor Instructions	
LWCz	Load Word from Coprocessor
SWCz	Store Word to Coprocessor
MTCz	Move to Coprocessor
MFCz	Move from Coprocessor
CTCZ	Move Control to Coprocessor
CFCz	Move Control from Coprocessor

OP	Description
COPz	Coprocessor Operation
BCzT	Branch on Coprocessor z True
BCzF	Branch on Coprocessor z False
System Control Coprocessor (CP0) Instructions	
MTCO	Move to CP0
MFCO	Move from CP0
TLBR	Read indexed TLB entry
TLBWI	Write indexed TLB entry
TLBWR	Write Random TLB entry
TLBP	Probe TLB for matching entry
RFE	Restore from Exception

3.4.2 Coprocessors

The R3000 can operate with up to four tightly coupled coprocessors, two of which are implemented in the DECsystem 5500 design.

3.4.2.1 Coprocessor (0)

The System Control Coprocessor (CP0) is part of the R3000 chip itself. Its function is to support the virtual memory system and exception handling functions. It contains the Translation Lookaside Buffer (TLB) plus the following registers:

- **EntryHi** - High half of a TLB entry.
- **EntryLo** - Low half of a TLB entry.
- **Index** - Programmable pointer into the TLB.
- **Random** - Pseudo-random pointer into the TLB.
- **Status** - Mode, interrupt enables, and diagnostic status information.
- **Cause** - Cause of last exception.
- **EPC** - Exception program counter.
- **Context** - Pointer into kernel's virtual page table entry array.
- **BadVA** - Most recent bad virtual address.
- **PRId** - Processor revision identification.

3.4.2.2 Coprocessor (1)

Coprocessor one (CP1) is the R3010 floating point unit. The R3010 floating point processor implements the IEEE arithmetic functions.

3.4.2.3 Coprocessor (2)

Coprocessor two (CP2) is not implemented but its branch condition input is used to signal whether the write buffer is full or not. BC2F branches if the write buffer is full, and BC2T branches if the write buffer is not full.

3.4.2.4 Coprocessor (3)

Coprocessor three (CP3) is not implemented but its branch condition input is used to signal whether the write buffer is busy or not. BC3F branches if the write buffer is busy, and BC3T branches if the write buffer is empty.

NOTE

Be careful when writing to I/O devices because all writes pass through the write buffer. The write buffers will pack adjacent byte writes into 2-byte, 3-byte, or word transactions. If it is important that I/O devices see individual byte writes, then software must ensure the write buffer flushes after each write. This can be accomplished by checking coprocessor three's conditions. Unlike the DECsystem 5400, reads are given priority over writes and doing an uncached read will not guarantee a write buffer flush.

3.4.3 Memory Management

The R3000 provides for logical expansion of memory space by translating virtual address into physical addresses, two Gbytes for the kernel and two Gbytes for the users. Virtual addresses are mapped through the use of a Translation Lookaside Buffer (TLB). This buffer contains 64 entries, each mapping a 4-Kbytes block. Controls are included for R/W accesses, cache or non-cache, and process identification. Figure 3–2 shows the memory space.

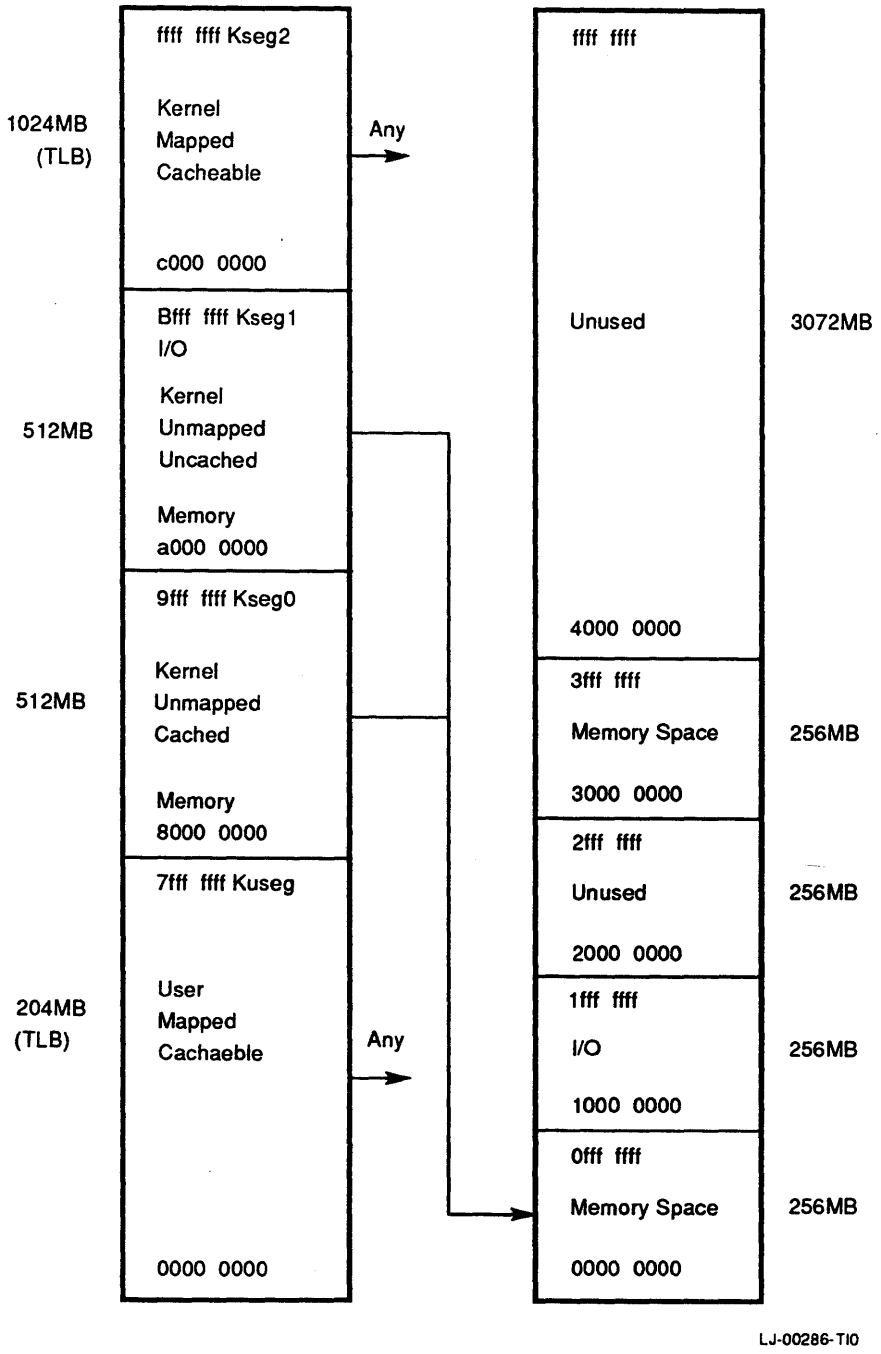


Figure 3–2 R3000 Virtual to Physical Memory Map

3.4.3.1 Operating Modes

The R3000 has two operating modes, kernel and user. All exceptions are handled in kernel mode. User mode is entered through the Restore From Exception (RFE) instruction.

3.4.3.1.1 Kernel Mode

Kernel mode contains four separate segments:

- **KUSEG** - references treated the same as user mode.
- **KSEG0** - references cached but not mapped.
- **KSEG1** - references not cached and not mapped.
- **KSEG2** - references mapped and cacheability is determined through the TLB entries.

3.4.3.1.2 User Mode

Two Gbytes of virtual address are available in this mode. Virtual addresses are extended with a 6-bit process identifier field allowing up to 64 user processes. All references in this mode are mapped through the TLB. Pages can be cached or uncached through the use of bits within the TLB entries.

3.4.3.2 EntryHi (EH) and EntryLow (EL) Registers

These registers are used to read, write, or probe the TLB. During exceptions, these registers are loaded with information about the address that caused the exception. The register format (Figure 3–3 and Figure 3–4) is the same as the format of a TLB entry.

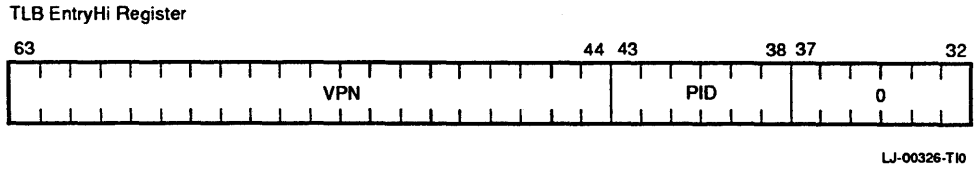


Figure 3–3 TLB EntryHi Register

Table 3–1 lists the bit descriptions.

Table 3–1 TLB EntryHi Register Bit Descriptions

Data Bit	Name	Description
VPN	Virtual Page Number	Contains bits <31:12> of the virtual address.
PID	Process ID Field	Allows multiple processes to share the TLB.
EntryHi<5:0>	Unused	Read as 0.

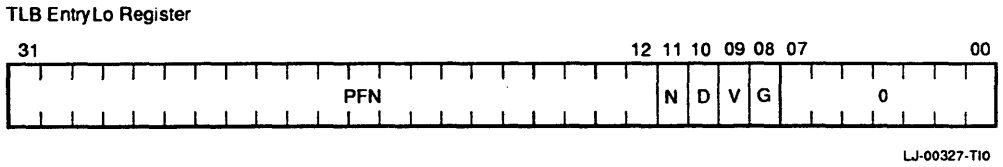


Figure 3–4 TLB EntryLo Register

Table 3–2 lists the bit descriptions.

Table 3–2 TLB EntryLo Register Bit Descriptions

Data Bit	Name	Description
PFN	Page Frame Number	Contains bits <31:12> of the physical address.
N	Non-Cacheable	When set the page mapped by this entry will not be cached.
D	Dirty	When set, the page mapped by this entry is writeable.
V	Valid	This bit validates the page thus allowing translations to take place, otherwise a TLBL or TLBS miss will occur.
G	Global	When set, the R3000 will ignore the PID match requirement for translations through this entry.
EntryLo<8:0>	Unused	Read as 0.

3.4.3.3 The Index Register (IR)

The Index register is used to index into the TLB when doing either TLB reads or TLB writes (Figure 3–5). The high-order bit returns status information for the Probe instruction.

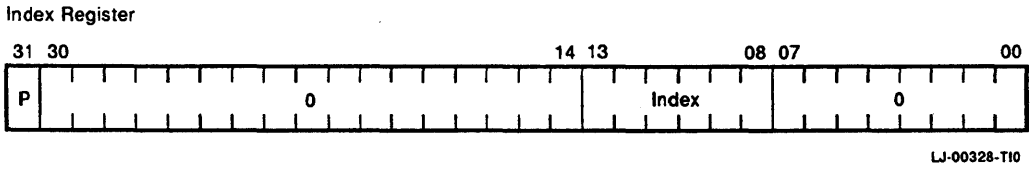


Figure 3–5 Index Register

Table 3–3 lists the bit descriptions.

Table 3–3 Index Register Bit Descriptions

Data Bit	Name	Descriptions
P	Probe Failure	Set to (1) if the last probe instruction failed.
<30:14>	Unused	Read as 0.
<13:8>		Indexes the TLB entry for TLB reads or writes.
<7:0>	Unused	Read as 0.

3.4.3.4 The Random Register (RR)

The Random register is used when replacing TLB entries (Figure 3–6). The TLBWR instruction is used to write the TLB entry pointed to by this register. The first eight TLB entries will never be written using the TLBWR instruction.

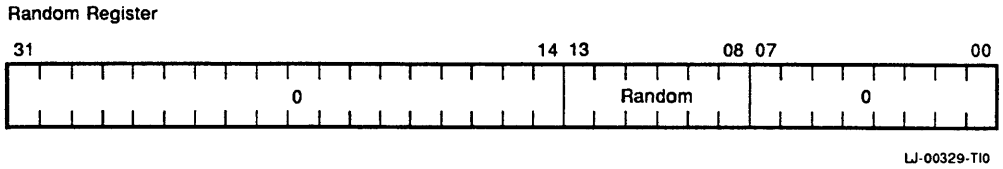


Figure 3–6 Random Register

Table 3–4 lists the bit descriptions.

Table 3–4 Random Register Bit Descriptions

Data Bit	Name	Description
<31:14>	Unused	Read as 0.
<13:8>	–	Random index ranging from 8 to 63 into the TLB.
<7:0>	Unused	Read as 0.

3.4.4 Exception Handling Registers

Exceptions are handled through the use of six Coprocessor Zero (CP0) registers and one module specific interrupt status register. The software uses these registers during exception handling to determine the cause of the exception. These registers are described in the following sections.

3.4.4.1 The Cause Register (CR)

The Cause Register is a 32-bit register used when servicing exceptions (Figure 3–7). It describes why the last exception was taken. All bits in this register are read only except for the SW bit.

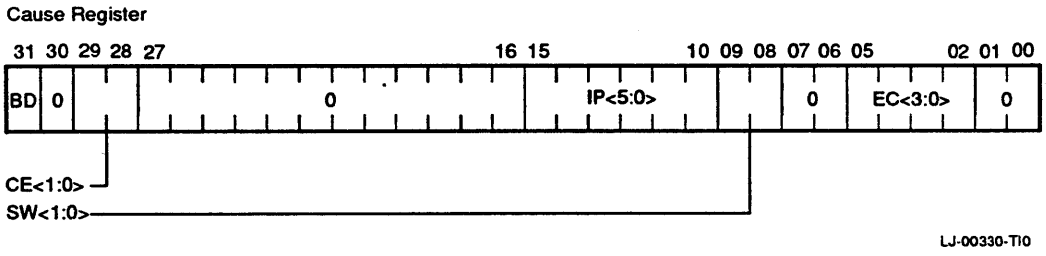


Figure 3–7 Cause Register

Table 3–5 lists the bit descriptions.

Table 3–5 Cause Register Bit Descriptions

Data Bit	Name	Description
<31>	BD	Branch Delay. Set to (1) if last exception was taken while executing in a branch delay slot.
<30>	Unused	Read as 0.

Table 3–5 (Cont.) Cause Register Bit Descriptions

Data Bit	Name	Description
<29:28>	CE<1:0>	Coprocessor Error. Indicates the unit number referenced when a coprocessor unusable exception is taken.
<27:16>	Unused	Read as 0.
<15:10>	IP<5:0>	Interrupt Pending. Indicates which of the external interrupts are pending. Table 3–6 shows the mapping of physical interrupt requests to the Cause register IP<5:0> status bits.
<9:8>	SW<1:0>	Software Interrupts. Indicates which software interrupts are pending. These bits are read/write.
<7:6>	Unused	Read as 0.
<5:2>	EC Exception Code	Table 3–7 shows the exception code field.
<1:0>	Unused	Read as 0.

Table 3–6 Physical Interrupt Requests to Cause Register

IP<5>	FPU
IP<4>	HALT (Described in the ISR section), PRFL
IP<3>	MEMINT
IP<2>	IRQ3 (Q22-bus BIRQ7), 100HZ clock
IP<1>	IRQ2 (Q22-bus BIRQ6), DSSI, SCSI, NI
IP<0>	IRQ0 or IRQ1 (Q22-bus BIRQ4 or BIRQ5), Console

Table 3–7 Exception Code Field

Number	Name	Description
0 (10)	INT	External Interrupt
1	MOD	TLB modification exception
2	TLBL	TLB miss exception (load or fetch)
3	TLBS	TLB miss exception (store)
4	ADEL	Address error exception (load or fetch)
5	ADES	Address error exception (store)
6	IBE	Bus error exception (fetch)
7	DBE	Bus error exception (load or fetch)
8	SYS	Syscall exception
9	BP	Breakpoint exception
10	RI	Reserved Instruction exception
11	CPU	Coprocessor unusable exception
12	OVF	Arithmetic overflow exception

Table 3–7 (Cont.) Exception Code Field

Number	Name	Description
13-15	reserved	

3.4.4.2 Exception Program Counter (EPC)

The Exception Program Counter contains the virtual address of the instruction that caused the exception to be taken. If the instruction is in a branch delay slot, the EPC contains the virtual address of the preceding branch or jump instruction.

3.4.4.3 The Status Register (SR)

The Status Register is a 32-bit register containing various processor statuses (Figure 3–8).

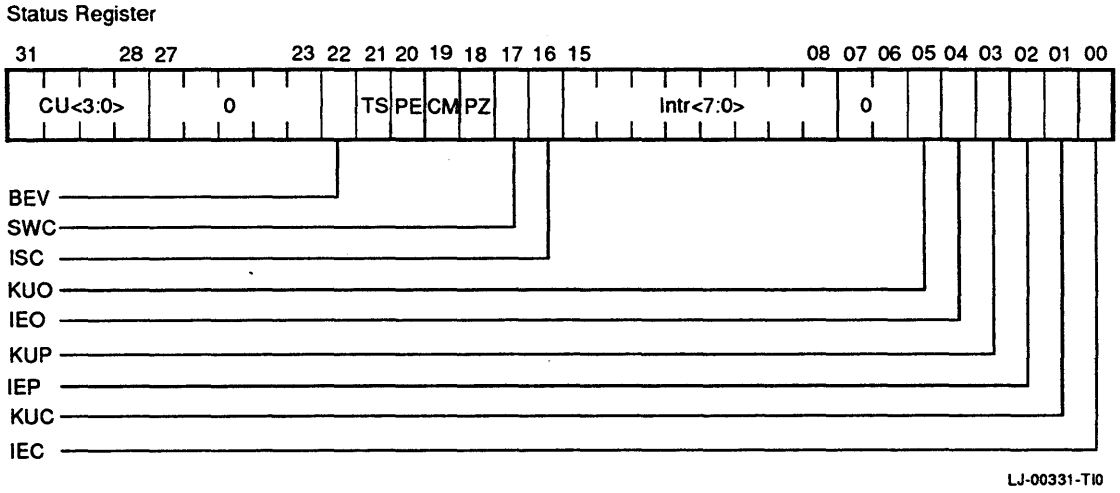


Figure 3–8 Status Register

Table 3–8 lists the bit descriptions.

Table 3–8 Status Register Bit Descriptions

Data Bit	Name	Description
<31:28>	CU<3:0>	Coprocessor Usability. CU<3:0> controls the availability of the four possible coprocessors. Setting a CU bit to (1) enables the coprocessor.
<27:23>	Unused	Read as 0.
<22>	BEV	Bootstrap Exception Vector. When set to (1) causes the R3000 to use the alternate bootstrap vectors for UTLB miss and general exceptions.
<21>	TS	Translation Buffer Shutdown. Set to (1) if the R3000 has disabled the translation buffer because of error.
<20>	PE	Parity Error. Set to (1) if a cache parity error occurs. Cleared by writing a (1).

Table 3–8 (Cont.) Status Register Bit Descriptions

Data Bit	Name	Description
<19>	CM	Cache Miss. Set to (1) if most recent D-cache load resulted in a miss when the D-cache is isolated.
<18>	PZ	Parity Zero. Setting to (1) forces parity bits to (0).
<17>	SWC	Swap Caches. Swaps I-cache and D-cache.
<16>	ISC	Isolate Cache. Isolates the D-cache from the memory system.
<15:10>	INTR<7:2>	Interrupt Mask. Setting these INTR bits to (1) enables their corresponding hardware interrupt.
<9:8>	INTR<1:0>	Interrupt Mask. Setting these INTR bits to (1) enables their corresponding software interrupt.
<5>	KUO Kernel /User mode, Old	Set to (0) if kernel, (1) if user.
<4>	IEO Interrupt Enable, Old	Set to (1) to enable, (0) to disable.
<3>	KUP Kernel /User mode, Previous	Set to (0) if Kernel, (1) if User.
<2>	IEP Interrupt Enable Previous	Set to (1) to enable, to (0) to disable.
<1>	KUC Kernel /User mode, Current	Set to (0) if Kernel, to (1) if User.
<0>	IEC Interrupt Enable, Current	Set to (1) to enable, to (0) to disable.

3.4.4.4 BadVaddr Register (BVA)

Saves the virtual address for any addressing exception.

3.4.4.5 Context Register (CR)

Used by the UTLB miss handler. Saves some of the same information as the BadVAddr register but in a more usable format. Figure 3–9 shows the context register.

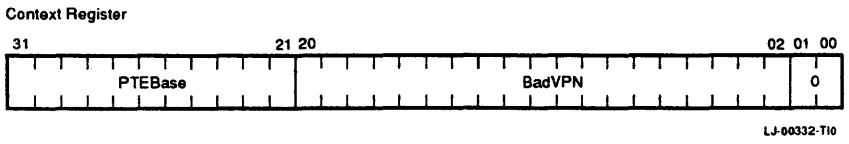


Figure 3–9 Context Register

Table 3–9 lists the bit descriptions.

Table 3–9 Context Register

Data Bit	Name	Description
<31:21>	PTEBase	Page Table Entry Base. Holds the base for the page table entry.
<20:2>	BadVPN	Bad Virtual Page Number. Holds the failing virtual page number. Read only. Set by hardware.
<1:0>	Unused	Read as 0.

3.4.4.6 Processor Revision Identifier Register (PRR)

This register contains implementation and revision numbers for the R3000 chip (Figure 3–10).

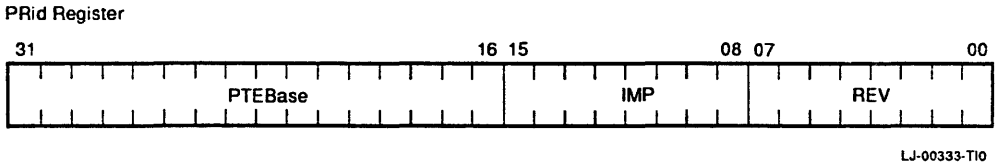


Figure 3–10 Processor Revision Identifier Register

Table 3–10 lists the bit descriptions.

Table 3–10 Processor Revision Identifier Register

Data Bit	Name	Description
<31:16>	Unused	Read as 0.
<15:8>	IMP<7:0>	Implementation Identifier.
<7:0>	REV<7:0>	Revision Identifier.

3.4.4.7 Interrupt Service and Interrupt Vectors

The R3000 uses the Vector Read Registers when servicing IRQ<2:0> interrupt requests. When servicing the IRQ<3> interrupt request, it uses the Vector Write Address. Reading any of the vector read registers generates an Interrupt Acknowledge Cycle on the peripheral bus (CVAX bus).

Bits <15:0> of the data returned will be a unique vector from the device responding to the interrupt acknowledge. If no device responds a **BUS ERROR EXCEPTION** will be taken by the R3000. When the vector write address is written to, the CPU module interpretes this as a memory interrupt acknowledge cycle. A pending memory interrupt IRQ<3> gets cleared by such a cycle.

System software must guarantee that interrupt requests from higher priority devices are serviced before lower priority devices. This can be accomplished by reading the VRR register corresponding to the highest IRQ<3:0> pending in the Cause/Status Register. The CVAX processor has no access to these registers.

Register	IPL Generated
VRR0	14 (lowest priority)
VRR1	15
VRR2	16
VRR3	17
VWR	none (highest priority)

NOTE
Because of the way interrupt requests are posted on the Q22-bus, interrupts at high level requests will also assert interrupt lower level requests. It is important that software read the VRR register corresponding to the highest IRQ pending in the CAUSE register. Lower level requests pending because of higher level interrupts will deassert at the same time the high level request deasserts.

3.4.4.7.1 Vector Read Register 0 (VRR0)

Figure 3–11 shows the vector read register 0.

VRR0 - Vector Read Register 0

- CVAX - no access
- R3000 - 16000050

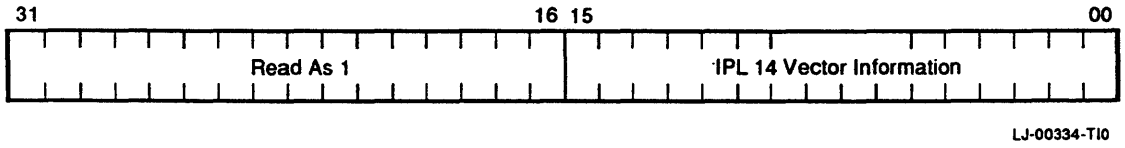


Figure 3–11 Vector Read Register 0

3.4.4.7.2 Vector Read Register 1 (VRR1)

Figure 3–12 shows the vector read register 1.

VRR1 - Vector Read Register 1

- CVAX - no access
- R3000 - 16000054

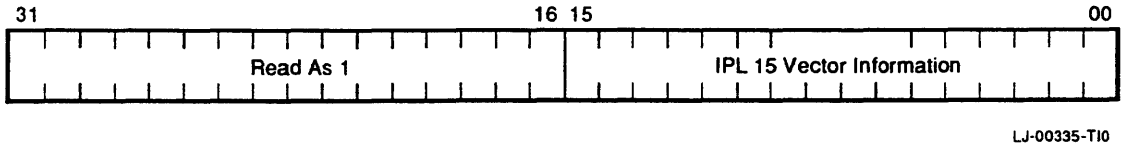


Figure 3–12 Vector Read Register 1

3.4.4.7.3 Vector Read Register 2 (VRR2)

Figure 3–13 shows the vector read register 2.

VRR2 - Vector Read Register 2

- CVAX - no access
- R3000 - 16000058

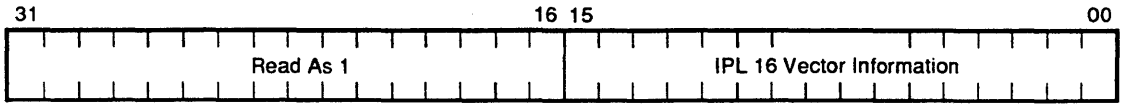


Figure 3–13 Vector Read Register 2

3.4.4.7.4 Vector Read Register 3 (VRR3)

Figure 3–14 shows the vector read register 3.

VRR3 - Vector Read Register 3

- CVAX - no access
- R3000 - 1600005C

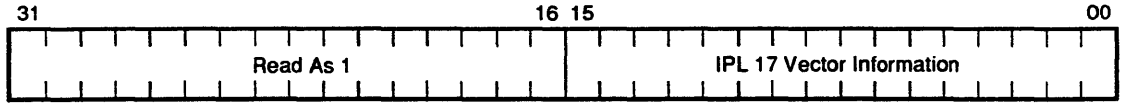


Figure 3–14 Vector Read Register 3

3.4.4.7.5 Vector Write Address

VWR - Vector Write Address

- CVAX - no access
- R3000 - 1610005C

3.4.5 Exceptions

The R3000 handles exceptions through the use of three different exception vectors. These exception vectors are:

- **BFC00000 virtual** - RESET exception vector.
- **80000000 virtual** - UTLB Miss exception vector.
- **80000080 virtual** - General exception vector.

If the Boostrap Exception Vector (BEV) bit is set in the Status Register then the UTLB and General Exception vector addresses will be changed, putting them in the ROM address space.

- **BFC00100 virtual** - UTLB Miss exception vector.

- **BFC00180 virtual** - General exception vector.

3.4.5.1 General Exception Vector

The General Exception Vector handles the following exceptions:

- Address error exception
- Breakpoint exception
- Bus Error exception
- Coprocessor unusable exception
- Interrupt exception
- Overflow exception
- Reserved instruction exception
- TLB miss exception
- TLB modified exception

3.4.5.2 Reset Exception Vector

The Reset exception is activated during power up when the R3000 RESET signal is deasserted. The Reset vector (BFC00000 virtual) resides in Kseg1 which is unmapped and uncached. It is also the first location in the ROM (physical 1FC00000).

When the Reset Exception occurs, the contents of all R3000 registers are undefined except for the following:

- TS, SWc, KUc, and IEc bits of the Status register are cleared.
- BEV bit of the Status register is set.
- Random register is cleared.

3.5 KN220-AA Cache Memory

To maximize CPU performance, the KN220-AA incorporates a 64KB instruction cache and a 64KB data cache.

3.5.1 Cache Organization

The cache is organized as two separate caches; one for instructions, one for data. Each cache contains 16K entries (or 64 Kbytes). Both caches have the same organization, they are direct mapped with a block size of one word (4-bytes). Fill size is either one word (4-bytes) or four words (16-bytes).

NOTE
Caching of the I/O space is permitted but not recommended.

3.5.2 Cache Isolation

The Data cache can be isolated from the memory system by setting the ISC bit in the status register. This allows the software to flush a cache location in a single cycle without affecting the memory system.

3.5.3 Cache Swapping

The caches can be swapped to help flush the instruction cache . This is done by setting the SWC bit in the status register. Precautions must be taken before setting this bit. The processor must be executing from an uncached region and must not be executing loads or stores near to the time of doing the swap.

3.5.4 Cache Line Format

Each cache line is 60 bits long, and each cache can be accessed once per cycle. This is accomplished by alternately putting out instruction cache addresses and data cache addresses on each phase of the clock. These cache addresses are latched externally. Figure 3–15 shows the cache line format.

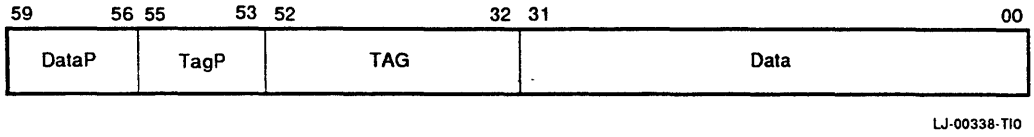


Figure 3–15 Cache Line Format

Table 3–11 lists the bit descriptions.

Table 3–11 Cache Line Format Bit Descriptions

Name	Description
DataP	Holds the parity bits for the data field.
TagP	Holds the parity bits for the TAG field.
TAG	Holds the Valid bit and the Page Frame Number.
Data	Cached data.

Figure 3–16 shows the tag.

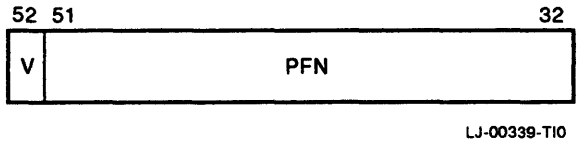


Figure 3–16 Tag

Table 3–12 lists the bit descriptions.

Table 3–12 Tag Bit Descriptions

Name	Description
V	Holds the Valid bit. The R3000 sets V whenever it writes a full word to the cache and clears V for any other writes. This allows the software to flush a cache location by doing byte store operations.

Table 3–12 (Cont.) Tag Bit Descriptions

Name	Description
PFN	Page Frame Number.

3.6 KN220-AA Main Memory System

The KN220-AA includes a main memory controller implemented through a number of programmable parts which contain the memory controller state machine and memory control signals. The KN220-AA Main Memory Controller communicates with the MS220-Ax memory boards over the MS220-Ax Memory Interconnect, which uses the CD interconnect for the address and control lines and a 100-pin ribbon cable for the data lines.

The controller supports the following:

- Up to four MS220-Ax memory boards.
- Synchronous 32-bit word read references from the R3000, and synchronous 4-word read references generated by cacheable R3000 references that miss the cache.
- Longword, quadword, hexword, or octaword asynchronous reads generated by the CVAX or any other DMA device that sits on the RIO bus.

  The expected ordering of data for transfers greater than a longword on the RIO bus is the same as the ordering for the CDAL bus.
- Masked or unmasked synchronous or asynchronous write references generated by the write buffer.
- Synchronous pagemode unmasked writes which can reduce write time by performing single-RAS, multiple-CAS cycles for consecutive words being written to the same page of memory.
- Asynchronous writes generated by DMA devices.

  The device performing the writes must notify the memory controller of pending writes that are on the same page. The write buffer always assumes a page size of 8Mbytes. Any DMA devices performing asynchronous quadword, hexword, or octaword writes also take advantage of the page mode capabilities of the memory controller.
- Asynchronous writes generated by the CVAX or any other DMA device that sits on the RIO bus, are also supported and may be longword, quadword, hexword, or octaword transfers. Only the first or last or both 32-bit access of quadword, hexword, or octaword transfers can have masked bytes in keeping with the CVAX architecture.
- The controller also supports synchronous or asynchronous masked write references.

  These references are executed by the memory controller as a read of a full word followed by a write of a full word, all taking place in one memory cycle. In addition, page mode writes can follow a masked write as long as all the ensuing writes are full word writes.

The memory controller performs CAS before RAS refreshing of all of the memory boards. All banks of all memory boards are refreshed simultaneously. Refreshes are requested every 15.6 microseconds, so 512 refresh cycles take place every 7.987 milliseconds. Table 3–13 lists the read reference timing.

Table 3–13 Read Reference Timing (all values are estimates)

	R3000	CVAX
single word (VAX longword)	333ns	400-500ns
VAX quadword	no access	600-700ns
VAX first longword	no access	400-500ns
VAX second longword	no access	200ns
VAX aborted reference	no access	400-500ns
VAX longword (locked)	no access	900-1000ns min
VAX aborted reference	no access	400-500ns
VAX retry (locked)	no access	500ns
4 word block read (VAX octalword)	500ns	no access

Table 3–14 lists the write reference timing of the memory.

Table 3–14 Write Reference Timing (all values are estimates)

	R3000	CVAX
Single word (VAX longword)	333ns	400ns
Single word (VAX longword) masked	600ns	700ns
Additional words in pagemode	133ns (each)	133ns (each)

3.6.1 Main Memory Error Detection and Correction

The KN220-AA main memory controller does not generate bus parity on CPU read references, nor does it check bus parity on CPU write references.

The memory controller protects main memory by using a 32-bit modified hamming code to encode the 32-bit data longword with seven check bits. This is implemented in a single 29C660D chip. One of these chips sits on each half of the 64-bit wide memory data path and generates and checks independent 7-bit codes for each half of the memory bank pairs. This allows the controller to detect and correct single-bit errors in the data field and detect single bit errors in the check bit field and double-bit errors in the data field. The most likely causes of these errors are failures in either the memory array or the 100-pin cable.

3.6.1.1 Main Memory Single Bit Errors

Upon detecting a correctable error on a read reference or during the read portion of a masked write reference, the data is corrected (if it is in the data field) before placing it on the RIO Bus, R3000 bus, or back in main memory. An interrupt is generated at IRQ3 for the R3000 (IPL17 for the CVAX). The Memory Error Address Register (MEAR) is loaded with:

- The address of the page containing the location that caused the error
- The NXM bit to indicate if the access was to non-existent memory

- The R3KBUSMASTER bit to indicate if the error occurred during an R3000 cycle or a DMA cycle.

The Memory Error Syndrome Register (MESR) is loaded with the error syndrome that indicates which bit was in error and which side of the bank pair it occurred in.

NOTE
The corrected data is not rewritten to main memory, so the single bit error will remain there until rewritten by the software.

3.6.1.2 Main Memory Multi-Bit Errors

Upon detecting an uncorrectable error on a read reference or during the read portion of a masked write reference, an interrupt is generated at IRQ3 if the reference was made by the R3000. A BUSERROR exception cycle will also be generated if a read was taking place. An interrupt is generated at IPL17 and an ERROR cycle occurs on the RIO bus if it happened during a DMA access. The MEAR is loaded with:

- The address of the memory containing the location that caused the error
- The NXM bit to indicate if the access was to non-existent memory
- The R3KBUSMASTER bit to indicate if the error occurred during an R3000 cycle or a DMA cycle.

The MESR is loaded with the error syndrome that indicates which word was in error and which side of the bank pair the error occurred in. These registers get updated even if a memory interrupt is pending so an uncorrectable error has the power to overwrite any other type of memory error. In such instances, earlier errors are lost. An uncorrectable error is rather serious and it is unwise to attempt to continue machine operation anyway.

3.6.1.3 Non-Existent Memory Errors

Upon detecting a memory reference to a non-existent memory location, indicated by the NXM signal from the memory board, an interrupt is generated at IRQ3 if the reference was made by the R3000. A BUSERROR exception cycle will also be generated if a read was taking place. An interrupt is generated at IPL17 and an ERROR cycle occurs on the RIO bus if it happened during a DMA access. The MEAR is loaded with the address of the non-existent memory reference and bit 0 of the register is set to 1 to indicate that this address does not correspond to a populated memory address. This register is also loaded with the R3KBUSMASTER bit to indicate if the NXM occurred during an R3000 cycle or a DMA cycle. The MESR is loaded with the error syndrome generated during the attempted reference; however, the data in this register is not particularly meaningful since the data bus was not driven by any memory devices during the cycle. These registers get updated even if a memory interrupt is pending so a non-existent memory error has the power to overwrite any other type of memory error. In such instances, earlier errors are lost. An uncorrectable error is rather serious and it is unwise to attempt to continue machine operation anyway.

3.6.1.4 Memory Interrupts during I/O cycles

Memory interrupts will also be generated when a bus owning device attempts to perform an I/O access that causes an ERROR cycle on the RIO bus. In this case, when the R3000 is bus master IRQ3 will be generated, and if it was a read cycle, a BUSERROR exception cycle will occur at the R3000. If it happened during a DMA cycle, an interrupt is generated at IPL17.

NOTE
Typically an I/O cycle by a DMA device does not make sense. However, the R3000 considers the CVAX diagnostic processor as a DMA device. The CVAX is the only DMA device which can perform I/O cycles.

The MEAR is loaded with the address of the error producing address. Bits 29 and 28 will either be 0 and 1 or 1 and 0, identifying the error as having occurred during an I/O cycle. The NXM bit will be set since the cycle did not go to memory and the R3KBUSMASTER bit will also get saved. The MESR will once again be unmeaningful since the NXM bit is on.

3.6.1.5 MEAR and MESR Updates

Upon detecting a correctable error on a read reference or during the read portion of a masked write reference, an interrupt is generated only if one is not already posted that has not been serviced yet. Also, if an interrupt from the memory controller is not already pending, the MEAR is loaded with the address of the memory containing the location that caused the error, and the MESR is loaded with the error syndrome which indicates which word was in error and which side of the bank pair the error occurred in. Since these errors are correctable, the machine is able to continue operation. However, since the data is only corrected for the CPU and cache during this access and not permanently in the memory array, it is useful for the CPU to know about them anyway through the interrupt. However, since an uncorrectable multi-bit ECC error and NXM error are far more critical, we do not want to lose one of those by overwriting it with information about a correctable error. Therefore, only update these registers during correctable errors when a memory error is not already pending.

3.7 KN220-AA Console Serial Line

The KN220-AA contains two console lines; one on the CPU board, and one on the I/O board. The console on the I/O board is the standard VAX console implemented using the SSC chip. It is available to the user through the H3602-AC patch panel.

The console serial line on the CPU module can be programmed to provide the KN220-AA processor with a full duplex, RS-423 EIA, serial line interface. It is implemented using the 2681 DUART chip. Only one channel of this chip is available and it feeds to an MMJ connector mounted on the CPU module itself. Hence, this port will only be available during debug and manufacturing and will not be available to the general user.

3.7.1 Console Registers

The operation of the DUART is programmed by reading and writing the appropriate registers. Operational feedback is provided by status registers that can be read by the CPU. The addressing of the registers is described in Table 3–15. All registers are 8-bits wide. Bits 31-8 of all registers are unused and return ones on reads.

Table 3–15 Console Register

Register Name	Mnemonic	R3000 Address	VAX Address	Reg. Type
Mode Register A	MRA	18100000	28100000	read/write
Status Register A	SRA	18100004	28100004	read only
Clock Select Register A	CSRA	18100004	28100004	write only

Table 3–15 (Cont.) Console Register

Register Name	Mnemonic	R3000 Address	VAX Address	Reg. Type
Command Register A	CRA	18100008	28100008	write only
Rx Holding Register A	RXRA	1810000C	2810000C	read only
Tx Holding Register A	TXRA	1810000C	2810000C	write only
Input Port Change Register	IPCR	18100010	28100010	read only
Aux. Control Register	ACR	18100010	28100010	write only
Interrupt Status Register	ISR	18100014	28100014	read only
Interrupt Mask Register	IMR	18100014	28100014	write only
Counter/Timer Upper Register	CTUR	18100018	28100018	read/write
Counter/Timer Lower Register	CTLR	1810001C	2810001C	read/write
Mode Register B	MRB	18100020	28100020	read/write
Status Register B	SRB	18100024	28100024	read only
Clock Select Register B	CSRB	18100024	28100024	write only
Command Register B	CRB	18100028	28100028	write only
Rx Holding Register B	RXRB	1810002C	2810002C	read only
Tx Holding Register B	TXRB	1810002C	2810002C	write only
Input Port		18100034	28100034	read only
Output Port Configuration Register	OPCR	18100034	28100034	write only
Start Counter Command		18100038	28100038	read only
Set Output Port Bits Command		18100038	28100038	write only
Stop Counter Command		1810003C	2810003C	read only
Reset Output Port Bits Command		1810003C	2810003C	write only

3.8 KN220-AA Boot and Diagnostic Facility

The KN220-AA Boot and Diagnostic facility is split across the I/O and CPU modules. Both boards have the hardware necessary to at least minimally boot and test the module as a stand-alone device. The features on the CPU module consist of two 40-pin ROM sockets capable of holding up to 256 Kbytes of read-only memory (configured as 64K, 32-bit words). There is also 32 Kbytes of battery backed up RAM (configured as 8K, 32-bit words). The ROM and battery backed up RAM may be accessed through longword, word, or byte references.

If this ROM is replaced for special applications, the new ROM must initialize and configure the board, provide HALT and console emulation, as well as provide boot and diagnostic functionality.

3.8.1 R3000 LED Register

The R3000 LED Register (RLEDR) is a write only register used to toggle the LEDs on the CPU module (Figure 3–17). LED<3:0> is used for this purpose. Bits 31-4 of this register are undefined and writing to them will have no effect. All four bits are cleared on power-up.

- R3000 - 17200000
- CVAX - 27200000

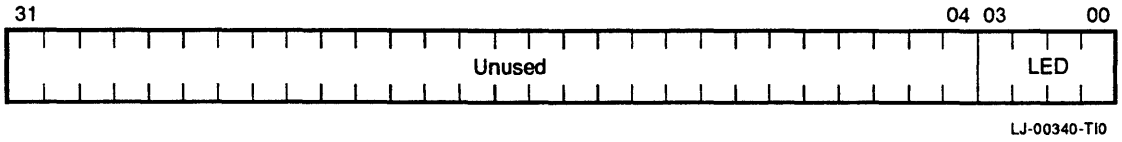


Figure 3–17 R3000 LED Register

Table 3–16 lists the bit descriptions.

Table 3–16 R3000 LED Register Bit Descriptions

Data Bit	Name	Description
<31:4>	Unused	–
<3:0>	LED Display	Write Only. These four bits update an external LED display. Writing a "0" to a bit lights the corresponding LED. Writing a "1" to a bit turns its LED off. The display bits are cleared (all LEDs are lit) on power-up.

3.8.2 ROM Memory

The KN220-AA CPU board supports 256 Kbytes of ROM memory for storing code for board initialization, board self-tests, and boot code. There is an additional 128 Kbytes on the I/O board for the CVAX processor. ROM memory may be accessed through byte, word, and longword references. Bus parity is neither checked nor generated on ROM references.

3.8.2.1 ROM Socket

The CPU module provides two 40-pin 600 mil wide ROM socket to accept two 64K x 16-bit EPROMs or ROMs. For code and driver development, CYM1623 RAM modules have been modified to be pin compatible with these sockets as well.

3.8.2.2 ROM Address Space

Two 64 Kbytes by 16-bit ROMs reside on the KN220-AA CPU module. One similar ROM resides on the KN220-AA I/O module. Both processors have access to all three ROMs.

The CPU module ROM address space goes from 1FC00000 to 1FC3FFFF in the R3000 memory map (2FC00000 to 2FC3FFFF in CVAX space).

3.8.2.2.1 Power Up Modes

The Boot and Diagnostic ROM programs use Boot and Diagnostic Code <1:0> to determine the power up modes as follows:

Code	Mode
00	Run
01	Language Inquiry
10	Test
11	Manufacturing

3.8.3 Battery Backed-Up RAM

The KN220-AA CPU module contains 512 Kbytes of battery backed-up static RAM for use as a console "scratchpad." It can also be used for disk caching to enhance system performance. The battery power for the RAM is provided through an on board battery. There is a jumper on board that can be used to prevent battery drain when the boards are in inventory. Setting 0 of this jumper prevents drain. Setting 1 permits battery backup of the RAMs. The jumper should be moved from setting 0 to 1 before a system is shipped to a customer.

The RAM is organized as a 128K X 32-bit array. The array appears in a 512 Kbytes block of R3000 address space at 18000000-1807FFFF and in the CVAX I/O page at addresses 28000000-2807FFFF.

This array is not protected by parity, and bus parity is neither checked nor generated on reads or writes to this RAM.

3.8.4 KN220-AA CPU Module Initialization

There is only one kind of hardware initialization for the CPU module. The initialization can occur three different ways: power-up initialization, processor initialization, and I/O bus initialization.

3.8.4.1 Power Up Initialization

Power up initialization is the result of the restoration of power and includes a hardware reset, a processor initialization, and an I/O bus initialization.

3.8.4.2 Hardware Reset

A KN220-AA hardware reset occurs on power up, the negation of DCOK when SCR<7> is clear, on the assertion of BINIT on the Q22-bus.

3.8.4.3 I/O Bus Initialization

An I/O bus initialization occurs on any one of the following:

- power-up, the negation of DCOK
- As the result of an R3000 write to the IORESET register (101400DC)
- A CVAX MTPR to IPR 55 (IORESET)
- A CVAX write to the IORESET register (201400DC)

An I/O bus initialization clears the IPCR and DSER and causes the Q22-bus interface to acquire both the CDAL and Q22-buses, then assert the Q22-bus BINIT signal. It also causes the assertion of the BRIORESET L signal to the CPU module.

3.8.4.4 Processor Initialization

A processor initialization occurs on power-up, the negation of DCOK, the assertion of BINIT as the result of a console INITIALIZE command, and after a halt caused by an CVAX error condition.

KN220-AA firmware must configure main memory, the local I/O page, and the Q22-bus map during a processor initialization.

3.8.5 I/O Presence Register (IOPRE)

The IOPRE register indicates to the CPU module whether any of the other modules on the RIO bus are present (Figure 3–18). Currently, the I/O module is the only other module that can be plugged into the RIO bus. There is a bit available for a future I/O option. This is a READ ONLY register.

- CVAX - 2700 0000
- R3000 - 1700 0000

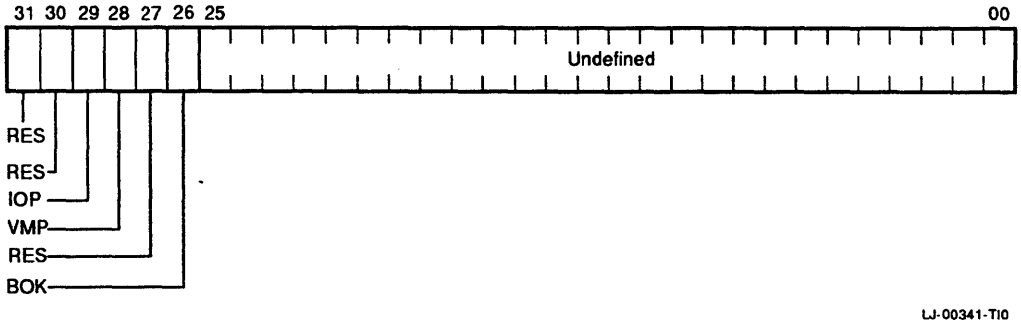


Figure 3–18 I/O Presence Register

Table 3–17 lists the bit descriptions.

Table 3–17 I/O Presence Register Bit Descriptions

Data Bit	Name	Description
<31:30>	Reserved	Reserved. These bits will always be read as 1.
<29>	IOP	When this bit is 1, this bit indicates that the I/O board is present. When 0, this bit indicates that the CPU board is running without the I/O board and the software should neither attempt to pass control to the CVAX nor try to access any of the devices on the I/O board.
<28>	VMP	When this bit is 1, this bit will indicate an I/O option module (future) is present. When 0, this bit indicates that the option module is not installed and the CPU should not try to access any of the devices on the option module.
<27>	Reserved	Reserved. This bit will always be read as 0.
<26>	BOK	When this bit is 1, this bit indicates that the battery backup voltage level is acceptable. When this bit is 0, the bit indicates that the battery voltage is too low and needs to be serviced.

Table 3–17 (Cont.) I/O Presence Register Bit Descriptions

Data Bit	Name	Description
<25:0>	Undefined	Undefined. Reading will produce unpredictable results.

3.8.6 Memory Error Syndrome Register (MESR)

The MESR register latches the ECC syndrome bits and ECC error type bits for any memory cycle producing an error (Figure 3–19). Any subsequent memory cycle which causes an error will change the MESR value. The MESR register is undefined on powerup. It is also used to monitor and drive the check bits during memory diagnostic mode (for example, when the DIAG bit, bit 3, of the MIDR is active).

- CVAX - 2704 0000
- R3000 - 1704 0000

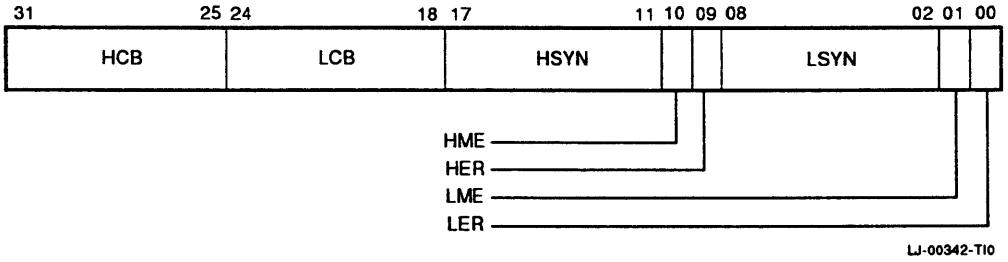


Figure 3–19 Memory Error Syndrome Register

Table 3–18 lists the bit descriptions.

Table 3–18 Memory Error Syndrome Register Bit Descriptions

Data Bit	Name	Description
<31:25>	HCB<6:0>	The seven check bits of the ECC chip that correspond to the high word of data. These bits can only be read or written when bit 3 of the MIDR is high. The read and write data values are actually two separate registers whose values do not update each other. The read value is updated whenever a read or partial word write happens in memory diagnostic mode.
<24:18>	LCB<6:0>	The seven check bits of the ECC chip that correspond to the low word of data. These bits can only be read or written when bit 3 of the MIDR is high. The read and write data values are actually two separate registers whose values do not update each other. The read value is updated whenever a read or partial word write happens in memory diagnostic mode.
<17:11>	HSYN<6:0>	The seven bit syndrome of the ECC chip checking the high word of data. These bits are read only.
<10>	HME	When this bit is 0, this bit indicates that multiple errors were detected by the ECC check of the high word of data. This bit is read only.

Table 3–18 (Cont.) Memory Error Syndrome Register Bit Descriptions

Data Bit	Name	Description
<9>	HER	When this bit is 0 and HME is 1, a single error was detected by the ECC check of the high word of data. This bit is read only.
<8:2>	LSYN<6:0>	The seven bit syndrome of the ECC chip checking the low word of data. These bits are read only.
<1>	LME	When this bit is 0, this bit indicates that multiple errors were detected by the ECC check of the low word of data. This bit is read only.
<0>	LER	When this bit is 0 and LME is 1, a single error was detected by the ECC check of the low word of data. This bit is read only.

3.8.7 Memory Error Address Register (MEAR)

The MEAR register latches bits 29-2 of the bus address, the R3KBUSMASTER bit, and the NXM signal for any cycle producing a memory error (Figure 3–20). Any subsequent cycle which causes an error may change the MEAR value depending on the priority level of the previous and current errors. The MEAR register is undefined on powerup. This is a read only register. Bits 31 and 30 will always be 0.

- CVAX - 2708 0000
- R3000 - 1708 0000

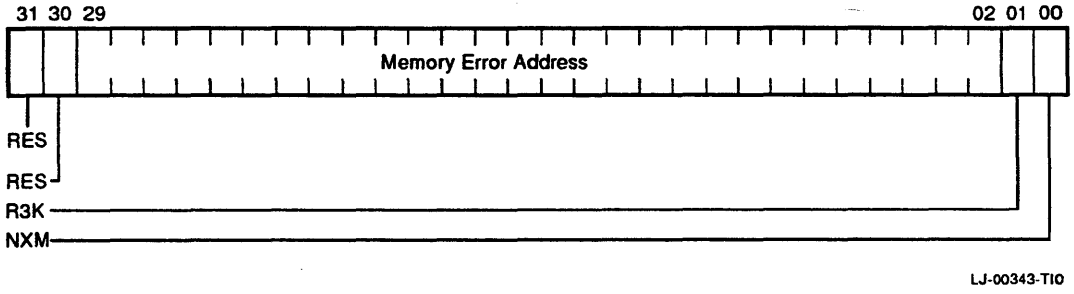


Figure 3–20 Memory Error Address Register

Table 3–19 lists the bit descriptions.

Table 3–19 Memory Error Address Register Bit Descriptions

Data Bit	Name	Description
<31:30>	Reserved	Read as 0.
<29:2>	Memory Error Address	Bits 29 through 2 of the address caused the highest priority memory error since the last time that the memory error interrupt was cleared.

Table 3–19 (Cont.) Memory Error Address Register Bit Descriptions

Data Bit	Name	Description
<1>	R3K	When this bit is high, it indicates that the currently logged error occurred while the R3000 was bus master. When this bit is low, it indicates that the error was during a DMA cycle.
<0>	NXM	When this bit is active high, it indicates that the memory address in the MEAR is a non-existent memory location.

3.9 I/O Module Specification

3.9.1 KN220-AA I/O Module Summary

The KN220-AA I/O module provides higher performance mass storage and network interfaces than are available over the Q22-bus. The mass storage interface consists of a DSSI port capable of a 4MB per second transfer rate that can control up to seven devices on the DSSI bus, and a SCSI port capable of a 4MB per second transfer rate. Both ports utilize a 32-bit data path to the R3000 or CVAX processors and each has 128 KB of static ram for buffer space. The network interface is a Second Generation Ethernet Controller (SGEC).

3.10 KN220-AA CPU Interface

3.10.1 I/O Devices

The KN220-AA I/O module connects to the CPU module through the 100-pin RIO bus cable and the C/D fingers in the backplane. It appears as nine asynchronous devices on the processor's buffered RIO bus. These devices are:

- Master/slave devices
  - CVAX diagnostic processor
  - CVAX Q22-bus interface chip (CQBIC)
  - Ethernet controller chip (SGEC)
- Slave only devices
  - Ethernet station address ROM
  - VAX compatible console port (through the SSC chip)
  - DSSI controller chip (SII)
  - DSSI buffer memory
  - SCSI controller chip (53C94)
  - SCSI buffer memory

Of these devices, the CVAX, SGEC, CQBIC and SSC are connected to each other through the I/O module's CP-bus. A separate interface controller translates between this CP-bus and the RIO bus, which contains the DSSI and SCSI subsystems.

The address decode logic selects the appropriate device to begin an I/O cycle. The device will assert BRIORDY L to signal the CPU that the cycle may terminate. CPU accesses to either buffer memory are 32 bits wide. CPU accesses to the SII or 53C94 chips are 16 bits wide on 32-bit boundaries.

CPU accesses to the station address ROM are 8 bits wide on 32-bit boundaries. All internal registers of the CVAX, SGEC, and CQBIC are 32 bits wide.

NOTE
All Addresses are shown as seen from the R3000 CPU. From the CVAX diagnostic processor, the addresses have an MSB of two instead of one.

3.10.2 KN220-AA I/O Initialization

There are three kinds of hardware initializations: power up initialization, processor initialization, and RIO bus initialization.

3.10.2.1 Power up Initialization

Power up initialization is the result of the restoration of power and includes a hardware reset, a processor initialization, and an RIO bus initialization.

3.10.2.2 Hardware Reset

A KN220-AA hardware reset occurs on power up, the negation of DCOK when SCR<7> is clear, on the assertion of BINIT on the Q22-bus. A hardware reset causes the hardware halt procedure to be initiated with a CVAX halt code of 03. It also initializes some IPRs and most I/O page registers to a known state. The effect a hardware reset has on I/O space registers is documented in the description of the register.

3.10.2.3 I/O Bus Initialization

An RIO bus initialization occurs on any one of the following:

- power up, the negation of DCOK
- As the result of an R3000 write to the IORESET register (101400DC)
- CVAX MTPR to IPR 55 (IORESET)
- CVAX write to the IORESET register (201400DC)

An RIO bus initialization clears the IPCR and DSER and causes the Q22-bus interface to acquire both the CDAL and Q22-buses, then assert the Q22-bus BINIT signal. It also causes the assertion of the BRIORESET L signal to the CPU module.

3.10.2.3.1 I/O Bus Reset Register

- R3000 - 101400DC
- CVAX - 201400DC (IPR 55)

The I/O bus reset register (IORESET) is implemented in the SSC chip.

3.10.2.4 Processor Initialization

A processor initialization occurs on power up, the negation of DCOK, the assertion of BINIT as the result of a console INITIALIZE command and after a halt caused by a CVAX error condition.

KN220-AA firmware must configure main memory, the local I/O page, and the Q22-bus map during a processor initialization.

3.10.2.4.1 Configuring the Local I/O Page

There are several registers that control the configuration of the KN220-AA local I/O Page. These registers must be configured by the KN220-AA firmware during a processor initialization and includes the:

- SSC base address register
- SSC configuration register
- SSC match and mask registers (ADMTR0,ADMTR1,ADMKR0,ADMKR1)
- CDAL bus timeout register

3.10.2.5 SSC Base Address Register (SSCBR)

The SSC Base Address Register controls the base addresses of a two Kbyte block of the local I/O space which includes (Figure 3–21):

- Battery backed-up RAM
- Registers for the programmable timers
- CACR and BDR address decode match and mask registers
- Diagnostic LED register
- CDAL bus timeout register
- A set of diagnostic registers that allow several IPRs to be accessed through I/O page addresses.

This read/write register is set to 2014 0000 (hex) on power up and the negation of DCOK when SCR<7> is clear. SSCBR<31:30,10:0> are unused. They read as 0, and must be written as 0. SSCBR<29> is read as 1 and must be written as 1. This register should also be set to 2014 0000 (hex) by firmware during processor initialization. The SSCBR has the following format:

- R3000 - 10140000
- CVAX - 20140000

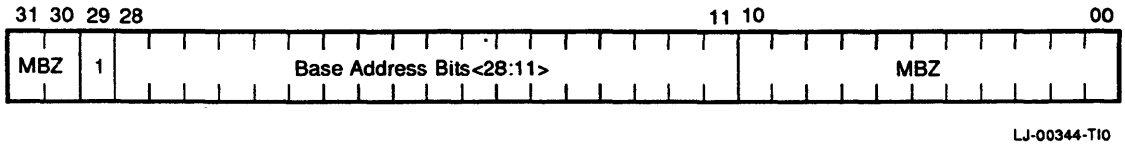


Figure 3–21 SSC Base Address Register

3.10.2.6 Address Decode Match Register (ADMTR0)

The Local I/O Device Address Decode Match Register 0 controls the base address of the board diagnostic register, select processor register, and interrupt status register (Figure 3–22). This read/write register is cleared on power up and the negation of DCOK when SCR<7> is clear. ADMTR0<31:30,1:0> are unused. They read as 0, and must be written as 0. This register should be set to 2008 4000 (hex) by firmware during processor initialization. The ADMTR0 has the following format:

- R3000 - 10140130

- CVAX - 20140130

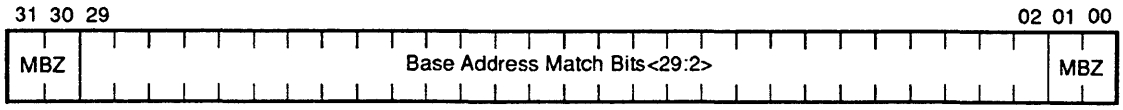


Figure 3–22 Address Decode Match Register 0

3.10.2.7 Address Decode Mask Register (ADMKR0)

The Local I/O Device Address Decode Mask Register 0 controls the range of addresses the BDR, IS, and SPR registers respond to (Figure 3–23). This read/write register is cleared on power up and the negation of DCOK when SCR<7> is clear. ADMKR0<31:30,1:0> are unused. They read as 0, and must be written as 0. This register should be set to 0000 000C (hex) by firmware during processor initialization. The ADMKR0 has the following format:

- R3000 - 10140134
- CVAX - 20140134

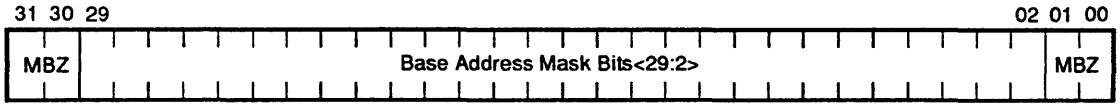


Figure 3–23 Address Decode Mask Register 0

3.10.2.8 Address Decode Match Register (ADMTR1)

The Local I/O Device Address Decode Match Register 1 controls the address of the CLK100 status register (Figure 3–24). This read/write register is cleared on power up and the negation of DCOK when SCR<7> is clear. ADMTR1<31:30,1:0> are unused. They read as 0, and must be written as 0. This register should be set to 2008 4010 (hex) by firmware during processor initialization. The ADMTR1 has the following format:

- R3000 - 10140140
- CVAX - 20140140

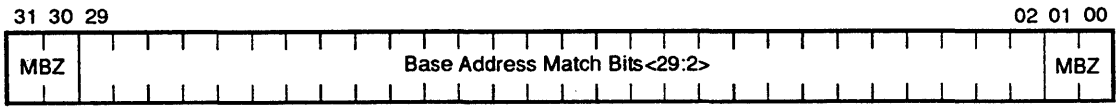


Figure 3–24 Address Decode Match Register 1

3.10.2.9 Address Decode Mask Register (ADMKR1)

The Local I/O Device Address Decode Mask Register controls the range of addresses the CLK100 register responds to (Figure 3–25). This read/write register is cleared on power up and the negation of DCOK when SCR<7> is clear. ADMKR1<31:30,1:0> are unused. This register should always be written with 0. The ADMKR1 has the following format:

- R3000 - 10140144
- CVAX - 20140144

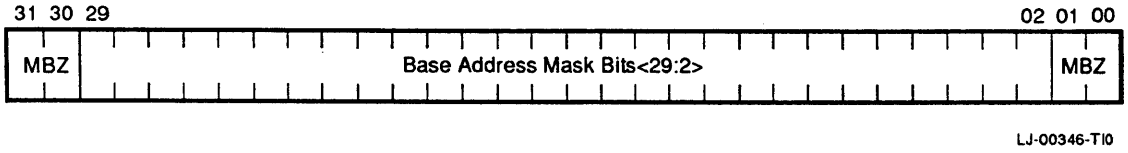


Figure 3–25 Address Decode Mask Register 1

3.10.2.10 SSC Configuration Register (SSCCR)

The SSC Configuration Register controls the set-up parameters for the console serial line, programmable timers, ROM, TOY clock, plus programmable address decode strobes (Figure 3–26).

- R3000 - 10140010
- CVAX - 20140010

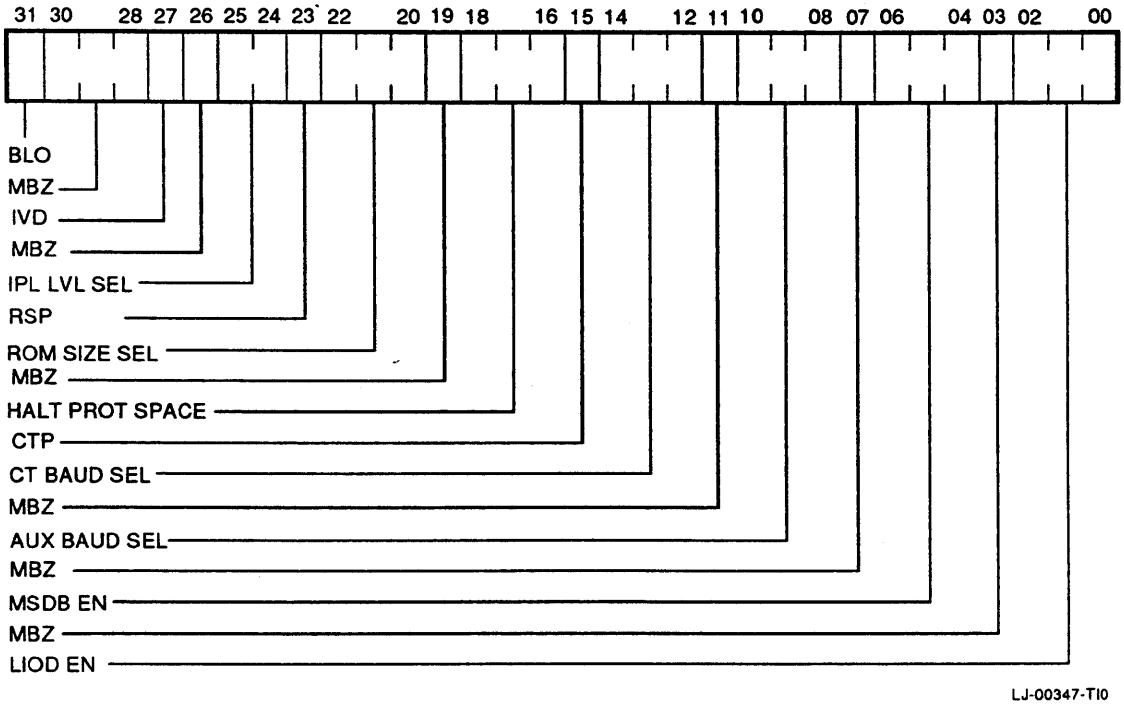


Figure 3–26 SSC Configuration Register

Table 3–20 lists the bit descriptions.

Table 3–20 SSC Configuration Register Bit Descriptions

Data Bit	Name	Description
<31>	BLO	Battery Low. Not used.
<30:28>	Unused	Read as 0, must be written as 0.
<27>	IVD	Interrupt Vector Disable. Read/Write. When set, the console serial line and programmable timers do not respond to interrupt acknowledge cycles. Cleared on power up by the negation of DCOK when SCR<7> is clear and during a processor initialization.
<26>	Unused	Read as 0, must be written as 0.
<25:24>	IPL LVL SEL	IPL Level Select. Read/Write. These bits are used to specify the IPL level of interrupt acknowledge cycle that the console serial line and programmable timers respond to. These bits must be cleared for the console serial line and programmable timers to respond to interrupt acknowledge cycles that they generated (IPL 14). These bits are cleared on power up by the negation of DCOK when SCR<7> is clear and during a processor initialization.
<23>	RSP	ROM Speed. Read/Write. This bit is used to select the ROM access time. This bit must be set for the KN220-AA ROMs to run at maximum speed. This bit is cleared on power up and by the negation of DCOK when SCR<7> is clear. It must be set to 1 during a processor initialization.
<22:20>	ROM SIZE SEL	ROM Address Space Size Select. Read/Write. These bits control the size of the range of addresses to which the ROM responds. These bits must be set at 101 (binary), yielding an R3000 address range of (1FC00000-1FC1FFFF) and a CVAX address range of (2004 0000 - 2007 FFFF hex). These bits are cleared on power up and by the negation of DCOK when SCR<7> is clear, yielding an R3000 address range of (1FC00000-1FC01FFF) and a CVAX address range of (2004 0000 - 2004 1FFF).
<18:16>	HALT PROT SPACE	ROM Halt Protect Address Space Size Select. Read/Write. These bits control the size of the Halt Mode address range. Setting these bits to 100 (binary) will halt protect the entire ROM address space. Setting these bits to 111 (binary) eliminates the entire halt protected address space. These bits are cleared on power up and by the negation of DCOK when SCR<7> is clear, yielding a Halt Mode address range of 8KB (2004 0000 - 2004 1FFF hex).
<15>	CTP	Control P Enable. Read/Write. Typing Ctrl/P at the console when this bit is set, causes the CPU to be halted, if halts are enabled (BDR<7> set). Typing Break at the console when this bit is cleared, causes the CPU to be halted, if halts are enabled (BDR<7> set). This bit is cleared on power up and by the negation of DCOK when SCR<7> is clear.

Table 3–20 (Cont.) SSC Configuration Register Bit Descriptions

Data Bit	Name	Description
<14:12>	CT BAUD SELECT	Console Terminal Baud Rate Select. Read/Write. These bits are used to select the baud rate of the console terminal serial line. They are cleared on power up and by the negation of DCOK when SCR<7> is clear. They should be loaded from BDR<6:4> by a processor initialization. The bit encodings correspond to selected baud rates as shown below:

SSCCR<14:12>	Baud Rate
000	300
001	600
010	1200
011	2400
100	4800
101	9600
110	19.2K
111	38.4K

<11>	Unused	Read as 0, must be written as 0.
<10:8>	Unused	Read/Write. Read as written. Cleared on power up and by the negation of DCOK when SCR<7> is clear.
<7>	Unused	Read as 0, must be written as 0.
<6:4>	CLK100 EN	Read/Write. These bits are used to enable the CLK100. They are cleared on power up and by the negation of DCOK when SCR<7> is clear. These bits must be set to 111 (binary) during initialization to enable the CLK100.
<3>	Unused	Read as 0, must be written as 0.
<2:0>	LIOD EN	Read/Write. These bits are used to enable the ISR, BDR, SPR registers. They are cleared on power up and by the negation of DCOK when SCR<7> is clear. These bits must be set to 111 (binary) during initialization to enable the ISR, BDR, and SPR registers.

3.10.3 CDAL Bus Timeout Control Register (CBTCR)

The CDAL Bus Timeout Register controls the amount of time allowed to elapse before a CDAL bus cycle is aborted (Figure 3–27). The effect of this timer is blocked by the KN220-AA logic on all Q22-bus references, since the Q22-bus interface has its own timers for Q22-bus references. This timer prevents unanswered CDAL bus cycles (other than those that go to the Q22-bus interface) from hanging the system any longer than the timeout interval. Note that even though the effect of the timer is blocked on all Q22-bus references that bits<31:30> will still be set on Q22-bus references and take longer than the programmmed value (4us). In this case these bits are not useful as error indicators.

- R3000 - 10140020
- CVAX - 20140020

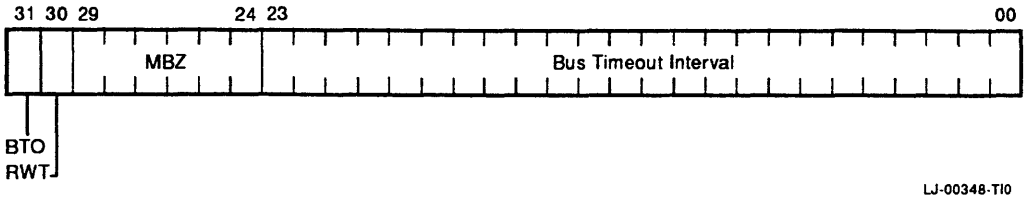


Figure 3–27 CDAL Bus Timeout Control Register

Table 3–21 lists the bit descriptions.

Table 3–21 CDAL Bus Timeout Control Register Bit Descriptions

Data Name	Bit	Description
<31>	BTO	CDAL Bus Timeout. Read/Write to clear. This bit is set when the bus timeout interval set in bits <23:0> has expired during a CDAL bus cycle. This bit is cleared by writing a 1 on power up and the negation of DCOK when SCR<7> is clear.
<30>	RWT	CDAL Bus Read/Write Timeout. Read/Write to clear. This bit is set when the bus timeout interval set in bits <23:0> has expired during a CPU or DMA read or write cycle on the CDAL bus. This bit is cleared by writing a 1 on power up and the negation of DCOK when SCR<7> is clear.
<29:22>	Unused	Read as 0, must be written as 0.
<23:0>	Bus Timeout Interval	Read/Write. These bits are used to program the desired timeout period. The available range of 1 to FFFFFF (hex) corresponds to a selectable timeout range of 1us to 16.77 seconds in 1us increments. Writing a 0 to this field disables the bus timeout function. The BTO bit is used to signify that a bus timeout has occurred. This field is cleared on power up and the negation of DCOK when SCR<7> is clear. This register should be loaded with 0000 0004 (hex) on a processor initialization for a timeout value of 4us.

3.10.4 Select Processor Register (SPR)

The Select Processor Register determines which processor controls the memory and I/O subsystems (Figure 3–28). This register is write only, reading produces unpredictable results.

- CVAX - 2008 4008
- R3000 - 1008 4008

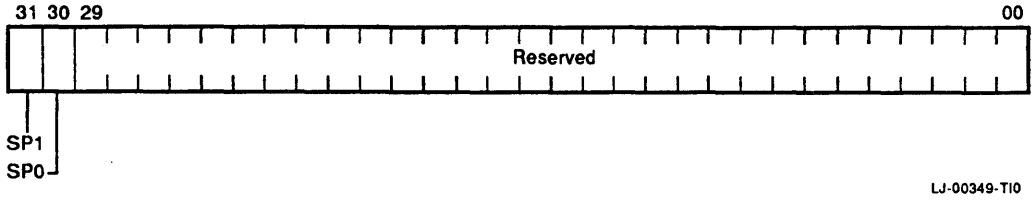


Figure 3–28 Select Processor Register

Table 3–22 lists the bit descriptions.

Table 3–22 Select Processor Register Bit Descriptions

Data Bit	Name	Description
<31:30>	SP	Select Processor. Write Only. When set to 01, it enables R3000 operation. When set to 10, it enables CVAX operation. On power up, the enabled CPU is determined by the signals CPUCOND<1:0>, IOPRESENT, and CPUPRESENT. The initial state of these bits does not matter. The R3000 is disabled by stalling memory operations while the CVAX is enabled. The CVAX is disabled by asserting DMA request and holding the bus while the R3000 is enabled. Writing a 00 or 11 to these bits does not affect the state of either processor. These bits will always be 1 on reads.
<29:0>	Undefined	Undefined. Reading produces unpredictable results.

3.10.5 Boot and Diagnostic Register (BDR)

The Boot and Diagnostic Register is a byte-wide register located in the VAX I/O page (Figure 3–29). It can be accessed by either processor, but not by the external Q22-bus devices. The BDR allows either processor to read various KN220-AA configuration bits. Only the low byte of the BDR should be accessed, bits <31:8> are undefined.

- R3000 - 10084004
- CVAX - 20084004

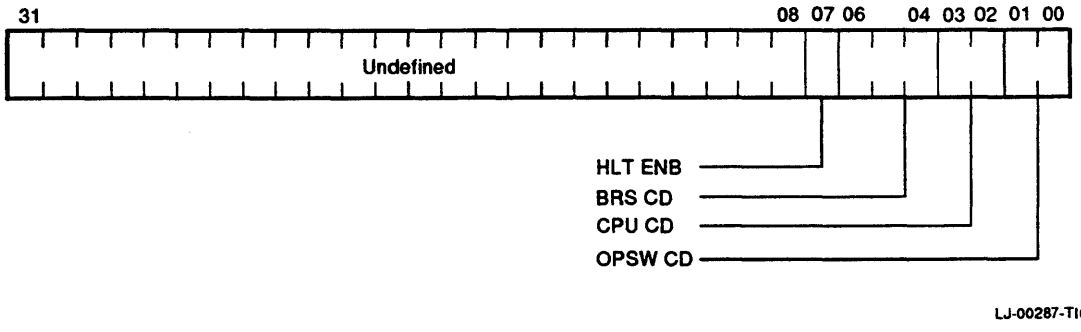


Figure 3–29 Boot and Diagnostic Register

Table 3–23 lists the bit descriptions.

Table 3–23 Boot and Diagnostic Register Bit Descriptions

Data Bit	Name	Description
<31:08>	Undefined	Should not be read or written.
<7>	HLT ENB	Halt Enable. Read Only. Writes have no effect. This bit reflects the state of pin 35 (ENBHALT L) of the 40-pin connector. The assertion of this signal enables the halting of the CPU upon detection of a console BREAK condition. On a power up, the KN220-AA resident firmware reads the HLT ENB bit to decide whether to enter the console emulation program (HLT ENB set) or to boot the operating system (HLT ENB clear).
<4:6>	BRS CD	Baud Rate Select. Read Only. Writes have no effect. These three bits originate from pins <20:28> (BRS<2:0>) of the 40-pin connector. They reflect the setting of the baud rate select switch on the CPU distribution insert.

BDR<6:4>	Baud Rate
000	300
001	600
010	1200
011	2400
100	4800
101	9600
110	19200
111	38400

<3:2>	CPU CD	CPU Code. Read Only. Writes have no effect. These two bits originate from pins <40:39> (CPU CD<1:0>) of the 40-pin connector.

CPU CD <1:0>	Configuration
00	CVAX on power up
01	R3000 on power up
10	undefined
11	Aux Mode (Unsupported)

Table 3–23 (Cont.) Boot and Diagnostic Register Bit Descriptions

Data Bit	Name	Description
<1:0>	OPSW CD	Operation Switch Code. Read Only. Writes have no effect. This 2-bit code reflects the status of configuration and display connector pins <37:36> (BDG CD<1:0>). The KN220-AA ROM programs use BDG CD <1:0> to determine power up modes.

OPSW CD <1:0>	Power Up Mode
00	Run
01	Language Inq.
10	Test
11	Manufacturing

3.10.6 Diagnostic LED Register (DLEDR)

The Diagnostic LED Register is implemented in the SSC chip and contains four Read /Write bits that control the external LED display (Figure 3–30). A zero in a bit lights the corresponding LED; all four bits are cleared on power up and the negation of DCOK when SCR<7> is clear to provide a power up lamp test.

- R3000 - 10140030
- CVAX - 20140030

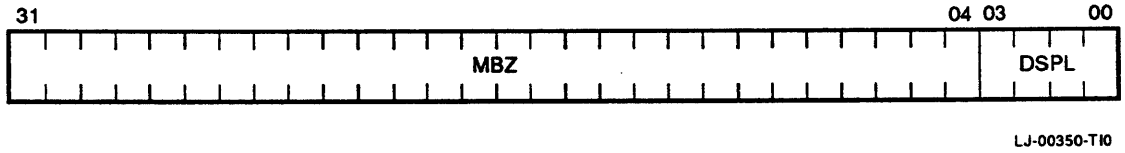


Figure 3–30 Diagnostic LED Register

Table 3–24 lists the bit descriptions.

Table 3–24 Diagnostic LED Register

Data Bit	Name	Description
<31:4>	Unused	Read as 0, must be written as 0.
<3:0>	DSPL	Display. Read/Write. These four bits update an external LED display. Writing a "0 to a bit lights the corresponding LED. Writing a "1" to a bit turns its LED off. The display bits are cleared (all LEDs are lit) on power up and the negation of DCOK when SCR<7> is clear.

3.11 I/O Device Interrupts

Devices on the KN220-AA I/O module can post interrupt requests to the CPU on BINT<3:0>, with BINT3 having the highest priority and BINT0 the lowest priority.

The priority of the Interrupt Acknowledge Daisy Chain is as follows:

- SSC console port
- 100HZ interval timer
- SGEC Ethernet controller
- SII
- 53C94
- Memory controller
- CQBIC

3.12 KN220-AA Console Serial Line

The console serial line provides the KN220-AA processor with a full duplex RS-423 EIA serial line interface, which is also RS-232C compatible. The only data format supported is 8-bit data with no parity and 1 stop bit. The four CVAX Internal Processor Registers (IPRs) that control the operation of the console serial line are a super-set of the VAX console serial line registers.

3.12.1 Console Registers

There are four registers associated with the console serial line unit. They are implemented in the SSC chip. They are memory mapped for the R3000 and are either memory mapped or accessible as internal processor register by the CVAX. Table 3–25 lists the registers.

Table 3–25 Console Registers

Register Name	Mnemonic	R3000 Address	VAX Address	VAX IPR#
Console receiver Control/status	RXCS	10140080	20140080	32
Console receiver Data buffer	RXDB	10140084	20140084	33
Console transmit Control/status	TXCS	10140088	20140088	34
Console transmit Data buffer	TXDB	1014008C	2014008C	35

3.12.1.1 Console Receiver Control/Status Register (RXCS)

The Console Receiver Control/Status Register is used to control and report the status of incoming data on the console serial line (Figure 3–31).

- R3000 - 10140080

- VAX - 20140080 (IPR 32)

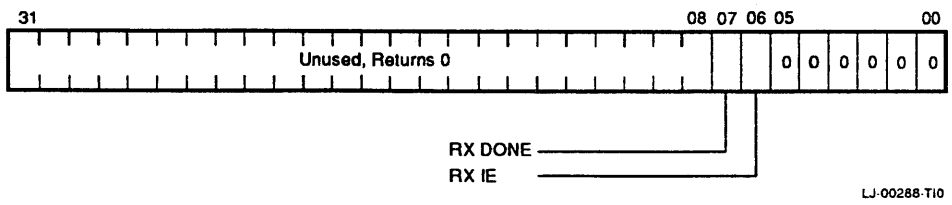


Figure 3–31 Console Receive Control/Status Register

Table 3–26 lists the bit descriptions.

Table 3–26 Console Receive Control/Status Register Bit Descriptions

Data Bit	Name	Description
<31:8>	Unused	Read as 0. Writes have no effect.
<7>	RX DONE	Receiver Done. Read Only. Writes have no effect. This bit is set when an entire character has been received and is ready to be read from the RXDB register. This bit is automatically cleared when the RXDB register is read. It is also cleared on power up and the negation of DCOK when SCR<7> is clear.
<6>	RX IE	Receiver Interrupt Enable. Read/Write. When set, this bit causes an interrupt to be requested at IPL14 with an SCB offset of F8 If RX DONE is set. When cleared, interrupts from the console receiver are disabled. This bit is cleared on power up and the negation of DCOK when SCR<7> is clear.
<5:0>	Unused	Read as 0. Writes have no effect.

3.12.1.2 Console Receiver Data Buffer (RXDB)

The Console Receiver Data Buffer is used to buffer incoming data on the serial line and capture error information (Figure 3–32).

- R3000 - 10140084
- VAX - 20140084 (IPR 33)

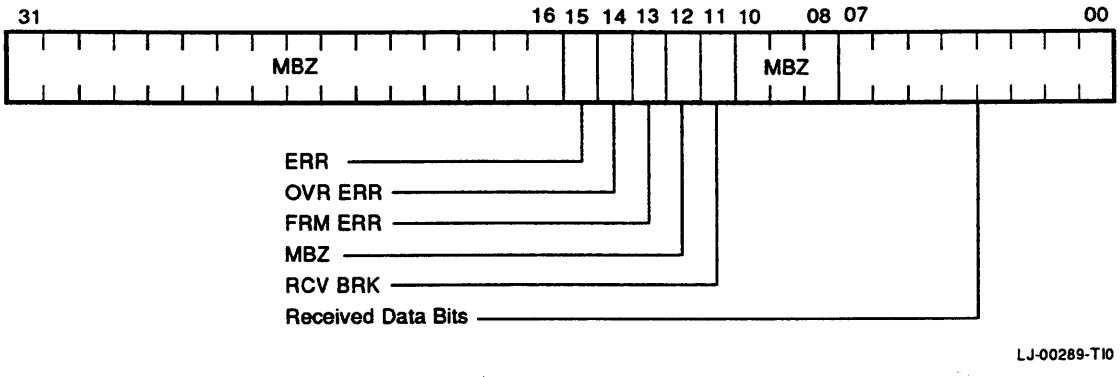


Figure 3–32 Console Receiver Data Buffer

Table 3–27 lists the bit descriptions.

Table 3–27 Console Receiver Data Buffer

Data Bit	Name	Description
<31:16>	Unused	Always read as 0. Writes have no effect.
<15>	ERR	Error. Read only. Writes have no effect. This bit is set if RBUF <14> or <13> is set. It is clear if these two bits are clear. This bit cannot generate a program interrupt. Cleared on power up and the negation of DCOK when SCR<7> is clear.
<14>	OVR ERR	Overrun Error. Read only. Writes have no effect. This bit is set if a previously received character was not read before being overwritten by the present character. Cleared by reading the RXDB on power up and the negation of DCOK when SCR<7> is clear.
<13>	FRM ERR	Framing Error. Read only. Writes have no effect. This bit is set if the present character did not have a valid stop bit. Cleared by reading the RXDB on power up and the negation of DCOK when SCR<7> is clear. **NOTE** **Error conditions remain present until the next character is received at which point the error bits are updated.**
<12>	Unused	This bit always reads as 0. Writes have no effect.
<11>	RCV BRK	Received Break. Read only. Writes have no effect. This bit is set at the end of a received character for which the serial data input remained in the SPACE condition for 20 bit times. Cleared by reading the RXDB on power up and the negation of DCOK when SCR<7> is clear.
<10:8>	Unused	These bits always read as 0. Writes have no effect.
<7:0>	Received Data Bits	Read only. Writes have no effect. These bits contain the last received character.

3.12.1.3 Console Transmitter Control/Status Register (TXCS)

The Console Transmitter Control/Status Register is used to control and report the status of outgoing data on the console serial line (Figure 3–33).

- R3000 - 10140088
- CVAX - 20140088 (IPR 34)

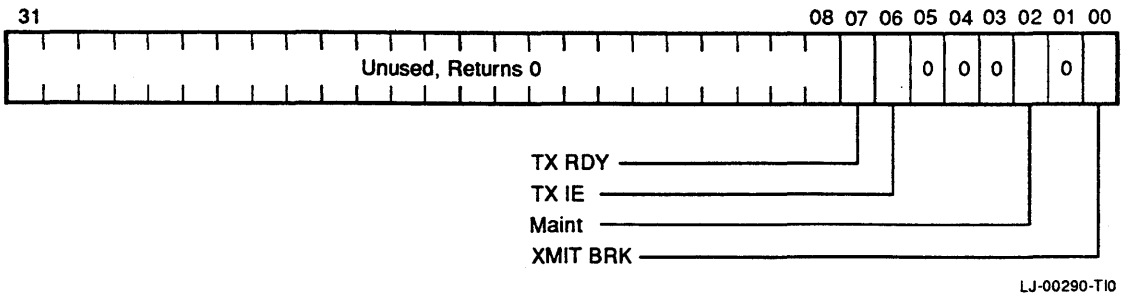


Figure 3–33 Console Transmitter Control/Status Register

Table 3–28 lists the bit descriptions.

Table 3–28 Console Transmitter Control/Status Register Bit Description

Data Bit	Name	Description
<31:8>	Unused	Read as 0. Writes have no effect.
<7>	TX RDY	Transmitter Ready. Read only. Writes have no effect. This bit is cleared when TXDB is loaded and set when TXDB can receive another character. This bit is set on power up and the negation of DCOK when SCR<7> is clear.
<6>	TX IE	Receiver Interrupt Enable. Read/Write. When set, this bit causes an interrupt to be requested at IPL14 with an SCB offset of FC if TX RDY is set. When cleared, interrupts from the console receiver are disabled. This bit is cleared on power up and the negation of DCOK when SCR<7> is clear.
<5:3>	Unused	Read as 0. Writes have no effect.
<2>	MAINT	Maintenance. Read/Write. This bit is used to facilitate a maintenance self-test. When MAINT is set, the external serial output is set to MARK and the serial output is used as the serial input. This bit is cleared on power up and the negation of DCOK when SCR<7> is clear.
<1>	Unused	Read as 0. Writes have no effect.
<0>	XMIT BRK	Transmit Break. Read/Write. When this bit is set, the serial output is forced to the SPACE condition after the character in TXB<7:0> is set. While XMIT BRK is set, the transmitter operates normally, but the output line remains low. Thus, the software can transmit dummy characters to time the break. This bit is cleared on power up and the negation of DCOK when SCR<7> is clear.

3.12.1.4 Console Transmitter Data Buffer (TXDB)

The Console Transmitter Data Buffer, is used to buffer outgoing data on the serial line.

- R3000 - 1014008C
- CVAX - 2014008C (IPR 35)

Table 3–29 lists the bit descriptions.

Table 3–29 Console Transmitter Data Buffer (TXDB) Bit Description

Data Bit	Name	Description
<31:8>	Unused	Read as 0. Writes have no effect.
<7:0>	Transmitted Data Bits	Write only. These bits are used to load the character to be transmitted on the console serial line.

3.12.2 Break Response

The console serial line unit recognizes a BREAK condition, which consists of 20 consecutively received SPACE bits. If the console detects a valid break condition, the RCV BRK bit is set in the RXDB register. If the break is the result of 20 consecutively received SPACE bits, the FRM ERR bit is also set. RCV BRK is cleared by reading RXDB. Another MARK followed by 20 consecutive SPACE bits must be received to set RCV BRK again.

3.12.3 Baud Rate

The receive and transmit baud rates are always identical and are controlled by the SSC Configuration Register bits <14:12>.

The user selects the desired baud rate through the baud rate select signals (BRS <2:0> L) which are received from an external 8-position switch through the 40-pin connector mounted at the top of the module. The KN220-AA firmware reads this code from boot and diagnostic register bits <6:4> and loads it into SSC configuration register bits <14:12>.

Table 3–30 shows the baud rate select signal voltage levels (H or L), the corresponding INVERTED code as read in the boot and diagnostic register bits <6:4>, and the code that should be loaded into SSC configuration register bits <14:12>:

Table 3–30 Baud Rate Select Signal Voltage Level

Baud Rate	BRS <2:0>	BDR <6:4>	SSC <14:12>
300	HHH	000	000
600	HHL	001	001
1200	HLH	010	010
2400	HLL	011	011
4800	LHH	100	100
9600	LHL	101	101
19200	LLH	110	110
38400	LLL	111	111

3.12.4 Console Interrupt Specifications

The console serial line receiver and transmitter both generate interrupts at IPL 14. The receiver interrupts with a vector of F8 (hex), while the transmitter interrupts with a vector of FC (hex).

3.13 KN220-AA TODR Clock and Timers

The KN220-AA clocks include Time Of Year Clock (TODR), a subset Interval Clock (subset ICCS), and two additional programmable timers modeled after the VAX standard interval clock, plus a 100Hz Interval Timer used as the R3000 interval clock.

3.13.1 R3000 Interval Timer Register (ITR)

The ITR register provides a 100Hz interval timer for the R3000 (Figure 3–34).

- R3000 - 10084010
- CVAX - 20084010

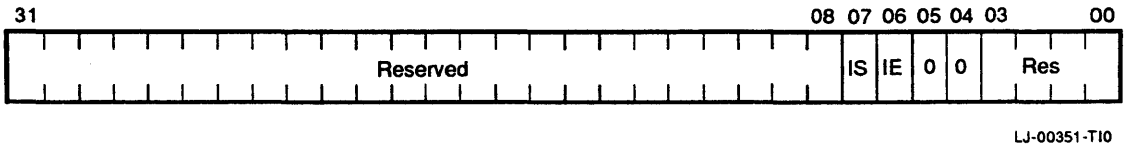


Figure 3–34 R3000 Interval Timer Register

Table 3–31 lists the bit descriptions.

Table 3–31 R3000 Interval Timer Register Bit Descriptions

Data Bit	Name	Description
<31:8>	Unused	Read as 1. Writes have no effect.
<7>	IS	Interrupt Status. Read only. Status bit shows when a 100Hz interrupt is pending. Cleared by reading VRR3 for the R3000 and by an IPL17 interrupt acknowledge for the CVAX.
<6>	IE	Interrupt Enable. Read/Write. This bit enables and disables the ITR interval timer interrupts. When this bit is set, an interval timer interrupt is requested every 10 msec with an error of less than .01%. When this bit is clear, interval timer interrupts are disabled. This bit is cleared on power up and during an I/O reset.
<5:4>	Reserved	Read as 0. Writes have no effect.
<3:0>	Reserved	Read as 1. Writes have no effect.

Interval timer requests are posted at IPL 17 with a vector of C0. The interval timer is the highest priority device at this IPL for the R3000.

3.13.2 Time-of-Year Clock

The KN220-AA Time-of-Year clock forms an unsigned 32-bit binary counter that is driven from a 100Hz oscillator, so that the least significant bit of the clock represents a resolution of 10 milliseconds with less than .0025% error. The register counts only when it contains a non-zero value. This register is implemented in the SSC chip (Figure 3–35).

- R3000 - 1014006C
- CVAX - 2014006C (IPR 27)

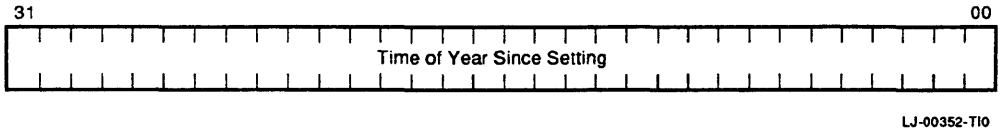


Figure 3–35 Time-of-Year Clock

The time-of-year (TOY) clock is maintained during power failure by battery backup circuitry, which interfaces through the external connector to a set of batteries that are mounted on the CPU distribution insert. The TOY remains valid for greater than 162 hours when using the NiCad battery pack (three batteries in series) mounted on the I/O distribution insert panel.

3.13.3 Interval Timer (ICCS)

The KN220-AA Interval Timer is implemented according to standards set for subset processors. The Interval Clock Control/Status Register is implemented as the standard subset of the Standard VAX ICCS in the CVAX CPU chip. NICR and ICR are not implemented (Figure 3–36).

- R3000 - 1001400F8
- CVAX - 2001400F8 (IPR 24)

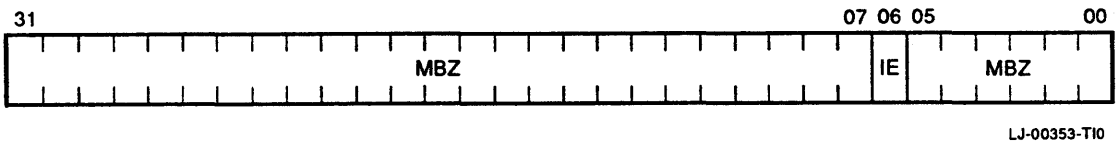


Figure 3–36 Interval Clock Control/Status Register

Table 3–32 lists the bit descriptions.

Table 3–32 Interval Clock Control/Status Register Bit Description

Data Bit	Name	Description
<31:7>	Unused	Read as 0. Must be written as 0.
<6>	IE	Interrupt Enable. Read/Write. This bit enables and disables the interval timer interrupts. When this bit is set, an interval timer interrupt is requested every 10 msec with an error of less than .01%. When this bit is clear, interval timer interrupts are disabled. This bit is cleared on power up and the negation of DCOK when SCR<7> is clear.
<5:0>	Unused	Read as 0. Must be written as 0.

Interval timer requests are posted at IPL 16 with a vector of C0. The interval timer is the highest priority device at this IPL.

3.13.4 Programmable Timers

The KN220-AA features two programmable timers. Although they are modeled after the VAX Standard Interval Clock, they are accessed as I/O space registers (rather than as internal processor registers). A control bit has been added, which stops the timer upon overflow. If it has been enabled, the timers interrupt at IPL 14 upon overflow. The interrupt vectors are programmable and are set to 78 and 7C by the firmware.

Each timer is composed of four registers:

- Timer n control register
- Timer n interval register
- Timer n next interval register
- Timer n interrupt vector register

  The letter n represents the timer number (0 or 1).

3.13.4.1 Timer Control Registers (TCR0-TCR1)

The KN220-AA has two Timer Control Registers, one for controlling timer 0 (TCR0), and one for controlling timer 1 (TCR1) (Figure 3–37). These registers are implemented in the SSC chip.

- R3000 - 10140100 and 10140110
- CVAX - 20140100 and 20140110

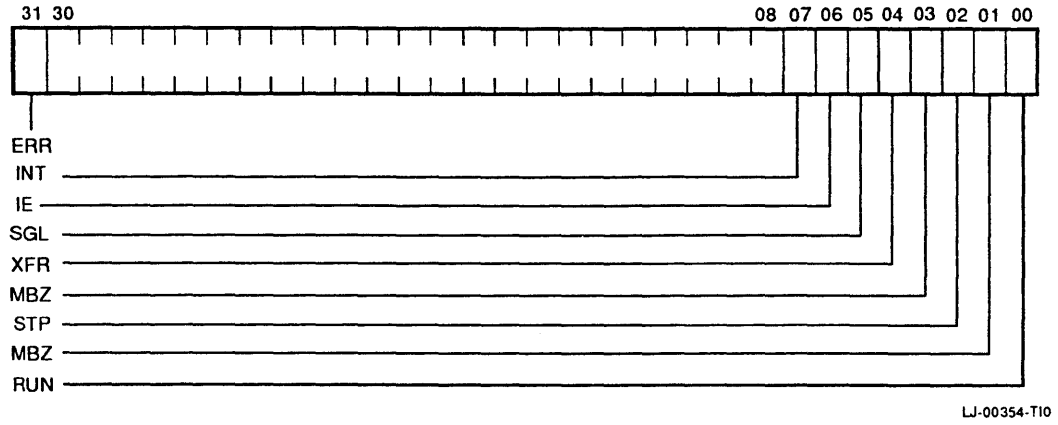


Figure 3–37 Timer Control Registers

Table 3–33 lists the bit descriptions.

Table 3–33 Timer Control Registers Bit Descriptions

Data Bit	Name	Description
<31>	ERR	Error. Read/Write to clear. This bit is set whenever the timer interval register overflows and the INT bit is already set. Thus, the ERR bit indicates a missed overflow. Writing a 1 clears the bit. This bit is cleared on power up and the negation of DCOK when SCR<7> is clear.
<30:8>	Unused	Read as 0. Must be written as 0.
<7>	INT	Read/Write to clear. This bit is set whenever the timer interval register overflows. If IE is set when INT is set, an interrupt is posted at IPL 14. Writing a 1 clears this bit. This bit is cleared on power up and the negation of DCOK when SCR<7> is clear.
<6>	IE	Read/Write. When this bit is set, the timer interrupts at IPL 14 when the INT bit is set. This bit is cleared on power up and the negation of DCOK when SCR<7> is clear.
<5>	SGL	Read/Write. Setting this bit causes the timer interval register to be incremented by 1 if the RUN bit is cleared. If the RUN bit is set, then writes to the SGL bit are ignored. This bit is always read as 0. This bit is cleared on power up and the negation of DCOK when SCR<7> is clear.
<4>	XFR	Read/Write. Setting this bit causes the timer next interval register to be copied into the timer interval register. This bit is always read as 0. This bit is cleared on power up and the negation of DCOK when SCR<7> is clear.
<3>	Unused	Read as 0. Must be written as 0.
<2>	STP	Read/Write. This bit determines whether the timer stops after an overflow when the RUN bit is set. If the STP bit is set at overflow, the RUN bit is cleared by the hardware at overflow and counting stops. This bit is cleared on power up and the negation of DCOK when SCR<7> is clear.

Table 3–33 (Cont.) Timer Control Registers Bit Descriptions

Data Bit	Name	Description
<1>	Unused	Read as 0. Must be written as 0.
<0>	RUN	Read/Write. When this bit is set, the timer interval register is incremented once every microsecond. The INT bit is set when the timer overflows. If the STP bit is set at overflow, the RUN bit is cleared by the hardware at overflow and counting stops. When the RUN bit is clear, the timer interval register is not incremented automatically. This bit is cleared on power up and the negation of DCOK when SCR<7> is clear.

3.13.4.2 Timer Interval Registers (TIR0-TIR1)

The KN220-AA has two Timer Interval Registers, one for timer 0 (TIR0), and one for timer 1 (TIR1).

The timer interval register is a read only register containing the interval count (Figure 3–38). When the RUN bit is 0, writing a 1 increments the register. When the RUN bit is one, the register is incremented once every microsecond. When the counter overflows, the INT bit is set and an interrupt is posted at IPL14 if the IE bit is set. If the RUN and STP bits are both set, the RUN bit is cleared and counting stops. Otherwise, the counter is reloaded. The maximum delay that can be specified is approximately 1.2 hours. This register is cleared on power up and the negation of DCOK when SCR<7> is clear.

- R3000 - 10140104 and 10140114
- CVAX - 20140104 and 20140114

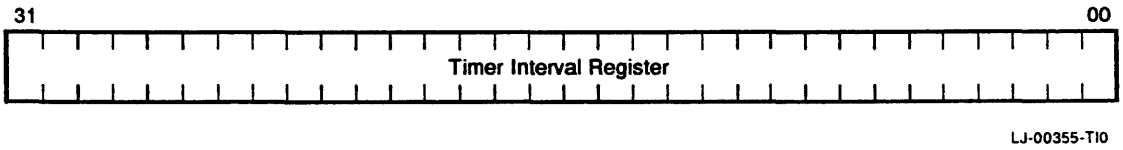


Figure 3–38 Timer Interval Register

3.13.4.3 Timer Next Interval Registers (TNIR0-TNIR1)

The KN220-AA has two Timer Next Interval Registers, one for timer 0 (TNIR0), and one for timer 1 (TNIR1) (Figure 3–39). These registers are implemented in the SSC chip.

This read/write register contains the value that is written into the timer interval register after overflow or in response to a 1 written to the XFR bit. This register is cleared on power up and the negation of DCOK when SCR<7> is clear.

- R3000 - 10140108 and 10140118
- CVAX - 20140108 and 20140118

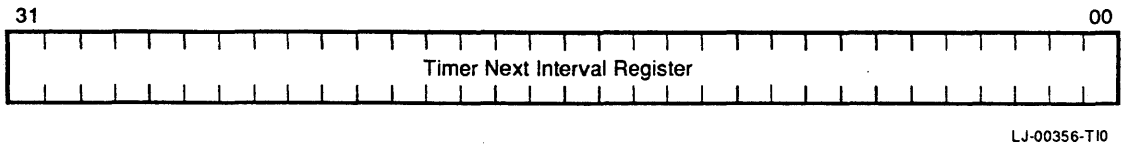


Figure 3–39 Timer Next Interval Registers

3.13.4.4 Timer Interrupt Vector Registers (TIVR0-TIVR1)

The KN220-AA has two Timer Interrupt Vector Registers, one for timer 0 (TIVR0), and one for timer 1 (TIVR1) (Figure 3–40). These registers are implemented in the SSC chip and are set to 78 and 7C respectively by the resident firmware.

This read/write register contains the timer's interrupt vector. Bits <31:10> and <1:0> are read as 0 and must be written as 0. When TCRn <6> (IE) and TCRn<7> (INT) transition to 1, an interrupt is posted at IPL 14. When a timer's interrupt is acknowledged, the content of the interrupt vector register is passed to the CPU and the INT bit is cleared. Interrupt requests can also be cleared by clearing either the IE or INT bit. This register is cleared on power up and the negation of DCOK when SCR<7> is clear.

- R3000 - 1014010C and 1014011C
- CVAX - 2014010C and 2014011C

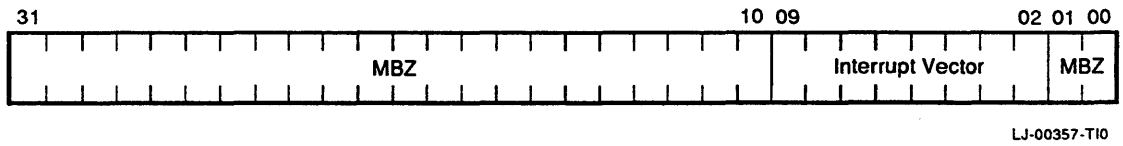


Figure 3–40 Timer Interruot Vector Registers

NOTE
Both timers interrupt at the same IPL (IPL 14) as the console serial line unit. When multiple interrupts are pending, the console serial line has priority over the timers, and timer 0 has priority over timer 1.

3.14 KN220-AA Network Interface

The M7638-Ax includes a network interface that is implemented through the SGEC and SIA. When used in conjunction with the H3602-AC cover panel, this interface allows the M7638-Ax to be connected to either a thinwire or standard Ethernet network.

The SGEC connects directly to the CP-bus, communicating with the host through Command and Status Registers (CSRs) and a host communication area set up in main memory. For data transfer, the SGEC uses an on-chip DMA controller suppporting both VAX virtual and physical memory addresses.

The SGEC features a dual, internal FIFO for decoupled and separate reception and transmission buffering facilitating efficient CP-Bus utilization. The FIFO holds the data until, at least, the collision window is passed.

3.14.1 Ethernet Overview

Ethernet is a serial bus that can support up to 1,024 local nodes with a maximum separation of 2.8 kilometers (1.7 miles). Data is passed over the Ethernet in Manchester encoded format at a rate of 10 million bits per second in variable-length packets. Each packet has the format shown in Figure 3–41.

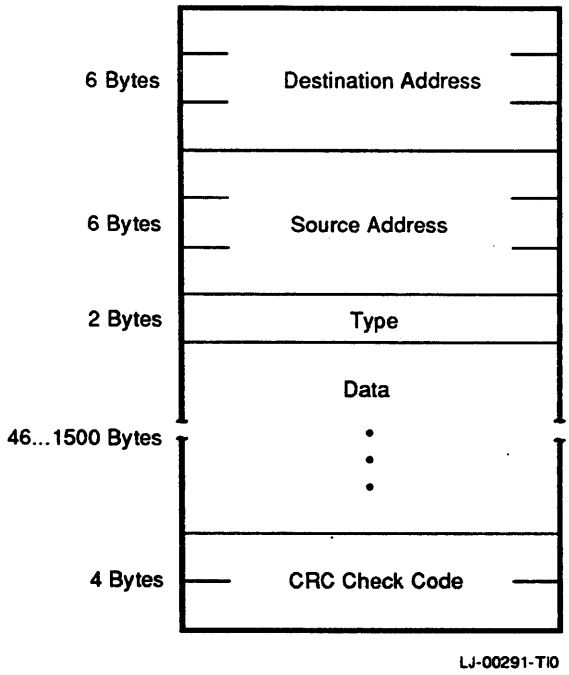


Figure 3–41 Ethernet Packet Format

The minimum size of a packet is 64 bytes, which implies a minimum data length of 46 bytes. Packets shorter than this are called "runt packets" and are treated as erroneous packets when received by the network controller.

All nodes on the Ethernet have equal priority. The technique used to control access to the bus is Carrier Sense Multiple Access, with Collision Detection (CSMA/CD). To access the bus, devices must first wait for the bus to clear (no carrier sensed). Once the bus is clear, all devices that want to access the bus have equal priority (multi-access), so they all attempt to transmit. After starting transmission, devices must monitor the bus for collisions (collision detection). If no collision is detected, the device may continue with transmission. If a collision is detected, then the device waits for a random amount of time and repeats the access sequence.

Ethernet allows point to point communication between two devices, as well as simultaneous communication between multiple devices. To support these two modes of communication, there are two types of network addresses, physical and multicast. These two types of addresses are both 48 bits (6 bytes) long. The addresses are described as follows:

Physical address: Physical address is the unique address associated with a particular station on the Ethernet, which should be distinct from the physical address of any other station on any other Ethernet.

Multicast address: Multicast address is a multi-destination address associated with one or more stations on a given Ethernet (sometimes called a logical address). There are two kinds of multicast addresses:

Multicast-group address: Multicast group address is an address associated by higher-level convention with a group of logically related stations.
Broadcast address: Broadcast address is a predefined multicast address that denotes the set of all the stations on the Ethernet.

Bit 0 (the least significant bit of the first byte) of an address denotes the type. Bit 0 is designated for physical addresses and Bit 1 for multicast addresses. In either case, the remaining 47 bits form the address value. A value of 48 ones (111111...and so on) is always treated as the broadcast address.

The hardware address of the M7638-Ax module is determined at the time of manufacturing and is stored in the network interface station address ROM. Since every device that is intended to connect to an Ethernet network must have a unique physical address, the bit pattern blasted into the network interface station address ROM must be unique for each M7638-Ax. The multicast addresses to which the M7638-Ax responds are determined by the multicast address filter mask in the network interface initialization block.

3.14.2 NI Station Address ROM (NISA ROM)

The network interface includes a byte-wide, 32-byte ROM called the network interface station address ROM (Figure 3–42). One byte of this ROM appears in the low-order byte (byte 0) of each of 32 consecutive longwords in the address range 1012 0000 - 1012 007C (hex). Bytes 1 through 3 of each longword are undefined. The low-order byte of the first six longwords contain the 48-bit Network Physical Address (NPA) of the M7638-Ax. The low-order byte in the remaining 26 longwords are unused. This address range is Read Only. Writes to this address range have no effect on the system.

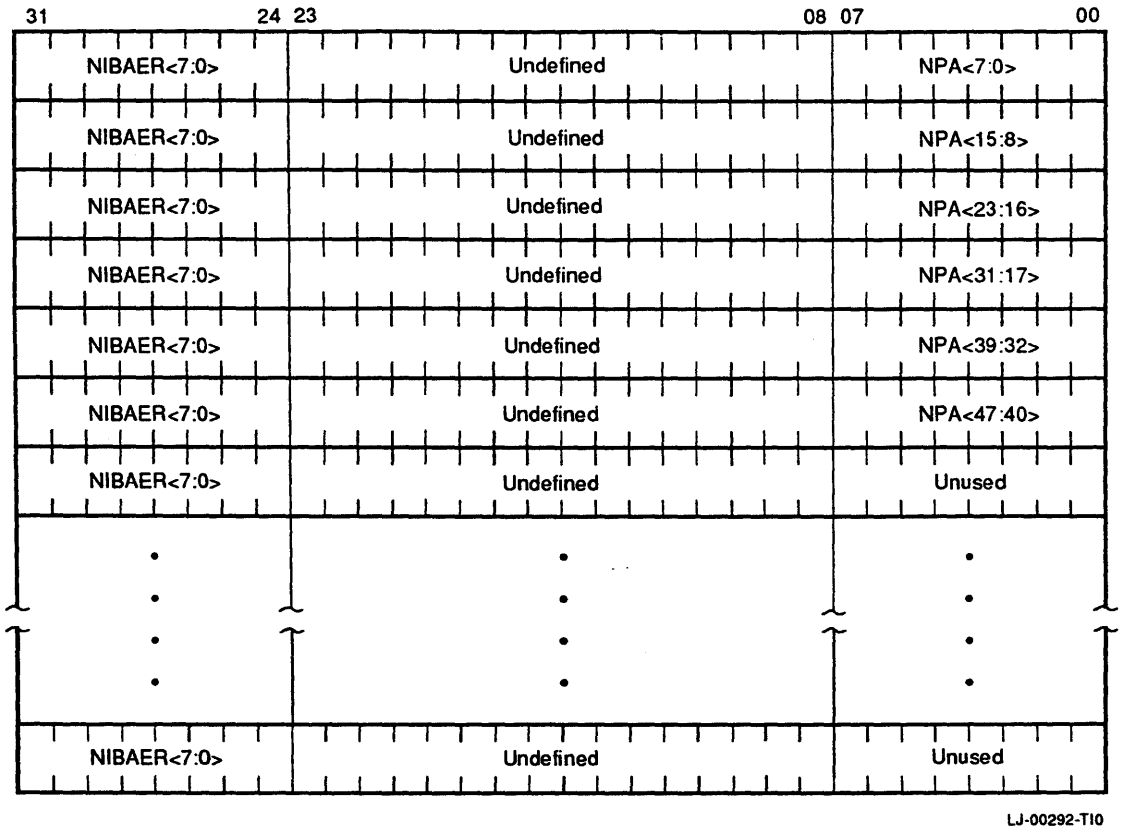


Figure 3–42 Network Interface Station Address ROM

3.15 Programming

The operation of the SGEC is controlled by a program in host memory called the port driver. The SGEC and the port driver communicate through two data structures: Command and Status Registers (CSRs). These CSRs are located in the SGEC and mapped in the host I/O address space, and through descriptor lists and data buffers, collectively called host communication area, in host memory.

The CSRs are used for initialization, global pointers, commands and reporting global errors, while the host memory resident structures handle the actions and statuses related to buffer management.

3.16 Programming Overview

The SGEC can be viewed as two independent concurrently executing processes: reception and transmission. After the SGEC completes its initialization sequence, these two processes alternate between three states: stopped, running, or suspended. State transitions occur as a result of port driver commands (writing to a CSR) or various external events. Some of the port driver commands require the referenced process to be in a specific state.

A simple programming sequence of the chip may be summarized as:

1. After power on (or reset), verify that the self test completed successfully.

2. Write CSRs to set major parameters such as system base register, interrupt vector, address filtering mode and so on.
3. Create the transmit and receive lists in memory and write the CSRs to identify them to the SGEC.
4. Place a setup frame in the transmit list to load the internal reception address filtering table.
5. Start the reception and transmission processes placing them in the running state.
6. Wait for SGEC interrupts. CSR5 contains all the global interrupt status bits.
7. Remedy the suspension cause if either the reception or transmission processes enter the suspended state.
8. Issue a Tx poll demand command to return the transmission process to the running state. To remedy the cause of the reception process suspension, an Rx poll demand could be issued to return the reception process to the running state.

   If the Rx poll demand is not issued, the reception process will return to the running state when the SGEC receives the next recognized incoming frame.

The following sections contain detailed programming and state transitions information.

3.17 Command and Status Registers

The SGEC contains 16 command and status registers that may be accessed by the host.

3.17.1 Host access to CSRs

The SGEC's CSRs are located in the VAX I/O address space.

The CSRs must be longword aligned and can only be accessed using longword instructions. The address of CSRx is the base address plus 4x bytes. For example, if the base address is 2000 8000, then the address of CSR2 is 2000 8008. In the following paragraphs, CSR bits are specified with several access modes. Table 3–34 lists the different bit access modes.

Table 3–34 Bit access modes

Bit marked	Meaning
0	Reserved for future expansion - Ignored on Write. Read as 0.
1	Reserved for future expansion - Ignored on Write. Read as 1.
R	Read only. Ignored on Write.
R/W	Read or Write.
W	Write only. Unpredictable on Read.
R/W1	Read, or Clear by writing a 1. Writing with a 0 has no effect.

To save chip real estate, yet not tie up the host bus for extended periods of time, the 16 CSRs are subdivided into two groups:

1. Physical CSRs - 0 through 7, 15.
2. Virtual CSRs - 8 through 14.

The group the CSR is part of, determines the way the host accesses it.

3.17.1.1 Physical CSRs

These registers are in the chip. Host access to these CSRs is by a single instruction (for example, MOVL). There is no host delay and the instruction completes immediately. Most commonly used SGEC features are contained in the physical CSRs.

NOTE
Read_Modify_Intend instructions are now supported by the SGEC.

3.17.1.2 Virtual CSRs

These registers are not in the SGEC and are incarnated by the on-chip processor. Accesses to SGEC functions implied by these registers may take up to 20 μseconds. So as not to tie up the host bus, virtual CSR access requires several steps by the host, as described below.

CSR5<DN> is used to synchronize access to the virtual CSRs. After the first virtual CSR access, the SGEC deasserts CSR5<DN> until it completes the action.

NOTE
Accessing the virtual CSRs, without first polling the CSR5<DN> reassertion, will cause unpredictable results.

3.17.1.2.1 Virtual CSR write

To write to a virtual CSR, the host takes the following actions:

1. Issues a write CSR instruction. Instruction completes immediately but the data is not yet copied by the SGEC.
2. Waits for CSR5<DN>. No SGEC virtual CSR may be accessed before CSR5<DN> asserts.

3.17.1.2.2 Virtual CSR read

To read a virtual CSR, the host takes the following actions:

1. Issues a read CSR instruction. Instruction completes immediately but no valid data is sent to the host.
2. Waits for CSR5<DN>. No SGEC virtual CSR may be accessed before CSR5<DN> asserts.
3. Reissues a read CSR instruction to the same CSR as in step 1. The host receives valid data.

3.17.2 Vector Address, IPL, Sync/Asynch (CSR0)

Since the SGEC may generate an interrupt on parity errors during host writes to CSRs, this register must be the first one written to by the host (Figure 3–43).

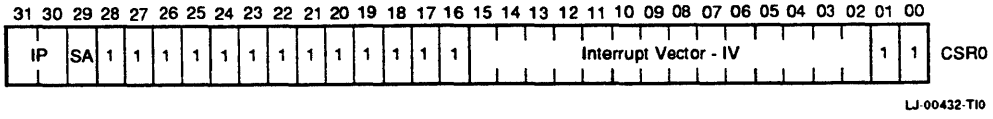


Figure 3–43 CSR0 Format

NOTE

A parity error during CSR0 host write may cause a host system crash because of an erroneous Interrupt Vector. To protect against a crash, CSR0 must be written while the IPL, to which the SGEC is assigned, is disabled. To write to CSR0:

1. **Write CSR0.**
2. **Read CSR0.**
3. **Compare value read to value written. If the values are mismatched, start over from step 1.**
4. **Read CSR5 and examine CSR5<ME> for pending parity interrupt. Should an interrupt be pending, write CSR5 to clear it.**

Table 3–35 lists the CSR0 bit descriptions.

Table 3–35 CSR0 Bits

Bit	Name	Access	Description
<15:00>	IV	R/W	Interrupt Vector - During an Interrupt Acknowledge cycle for an SGEC interrupt, it is the value that the SGEC drives on the host bus CDAL<31:0> pins (CDAL pins <1:0> and <31:16> are set to "0"). Bits <1:0> are ignored when CSR0 is written, and set to "1" when read.
<29>	SA	R/W	Sync/Asynch - This bit determines the SGEC operating mode when it is the bus master. When set, the SGEC will operate as a synchronous device and when clear, the SGEC operates as an asynchronous device.
<31:30>	IP	R/W	Interrupt Priority - is the VAX interrupt priority level that the SGEC will respond to.

IP	IPL (hex)
00	14
01	15
10	16
11	17

Although the SGEC has only one interrupt request pin, that pin might be wired to any of the four IRQ pins on the host. The value in IP should correspond to the IPL level that the pin is wired to.

Table 3–36 lists the CSR0 access rules.

Table 3–36 CSR0 Access

Value after **RESET**:	1FFF0003 hex
Read access rules:	None
Write access rules:	The IPL to which the SGEC is assigned must be disabled.

3.17.3 Transmit Polling Demand (CSR1)

Figure 3–44 shows the register.

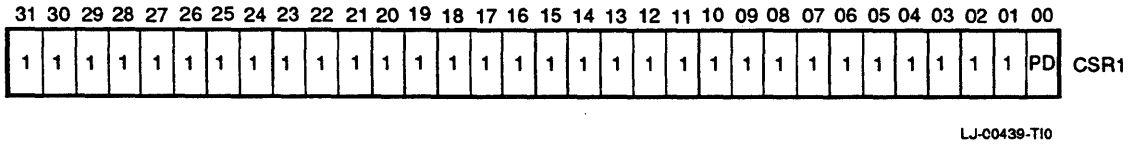


Figure 3–44 CSR1 Format

Table 3–37 lists the CSR1 bit descriptions.

Table 3–37 CSR1 Bits

Bit	Name	Access	Description
<00>	PD	W	Tx Polling Demand - Checks the transmit list for frames to be transmitted. The PD value is meaningless.

Table 3–38 lists the CSR1 access rules.

Table 3–38 CSR1 Access

Value after **RESET**:	Not applicable
Read access rules:	None
Write access rules:	Tx process suspended

3.17.4 Receive Polling Demand (CSR2)

Figure 3–45 shows the register.

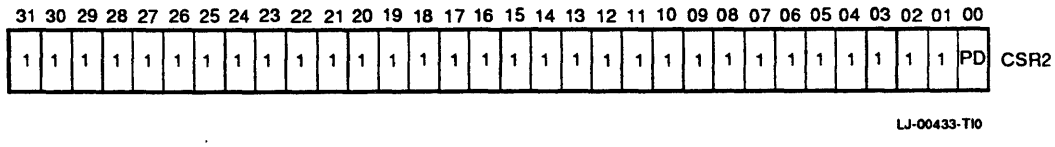


Figure 3–45 CSR2 Format

Table 3–39 lists the CSR2 bit descriptions.

Table 3–39 CSR2 Bits

Bit	Name	Access	Description
<00>	PD	W	Rx Polling Demand - Checks the receive list for receive descriptors to be acquired. The PD value is meaningless.

Table 3–40 lists the CSR2 access rules.

Table 3–40 CSR2 Access

Value after **RESET**:	Not applicable
Read access rules:	None
Write access rules:	Rx process suspended

3.17.5 Descriptor List addresses (CSR3, CSR4)

The two descriptor list head address registers are identical in function (Figure 3–46). One register is being used for the transmit buffer descriptors and one is being used for the receive buffer descriptors. In both cases, the registers are used to point the SGEC to the start of the appropriate buffer descriptor list.

The descriptor lists reside in VAX physical memory space and must be longword aligned.

NOTE
For best performance, it is recommended that the descriptor lists be octaword aligned.

NOTE
If the transmit descriptor list is built as a ring (the chain descriptor points at the first descriptor of the list), the ring must contain at least two descriptors in addition to the chain descriptor.

Initially, these registers must be written before the respective Start command is given (see Section 3.17.7), otherwise the respective process will remain in the stopped state. New list head addresses are only acceptable while the respective process is in the stopped or suspended state. Addresses written while the respective process is in the running state are ignored and discarded.

If the host attempts to read any of these registers before ever writing to them, the SGEC responds with unpredictable values.

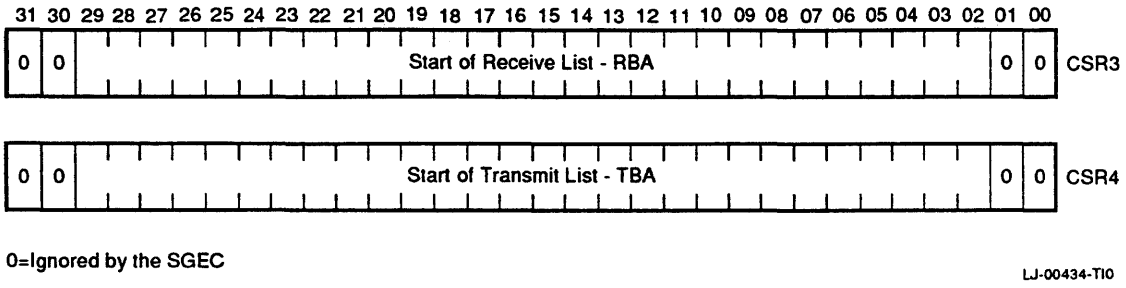


Figure 3–46 Descriptor List Addresses Format

Table 3–41 lists the bit descriptions.

Table 3–41 Descriptor List Addresses Bits

Bit	Name	Access	Description
<29:00>	RBA	R/W	Address of the start of the receive list. This is a 30-bit VAX physical address.
<29:00>	TBA	R/W	Address of the start of the transmit list. This is a 30-bit VAX physical address.

NOTE
The descriptor list must be longword aligned.

Table 3–42 lists the CSR3 access rules.

Table 3–42 CSR3 Access

Value after **RESET**:	Unpredictable
Read access rules:	None
Write access rules:	Rx process stopped or suspended

Table 3–43 lists the CSR4 access rules.

Table 3–43 CSR4 Access

Value after **RESET**:	Unpredictable
Read access rules:	None
Write access rules:	Tx process stopped or suspended

After either CSR3 or CSR4 are written, the new address is readable from the written CSR. If the SGEC status did not match the related write access rules, the new address does not take effect and the written information is lost, even if the SGEC later matches later the right condition.

3.17.6 Status Register (CSR5)

This register contains all the status bits the SGEC reports to the host (Figure 3–47).

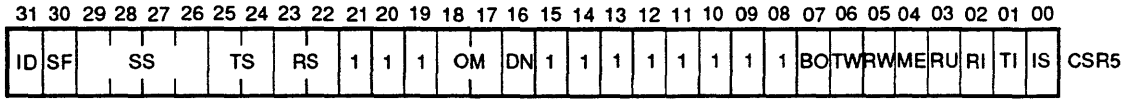


Figure 3–47 CSR5 Bits

Table 3–44 lists the CSR5 bit descriptions.

Table 3–44 CSR5 Bits

Bit	Name	Access	Description
<0>	IS	R/W1	Interrupt Summary - The logical "OR" of CSR5 bits 1 through 6.
<1>	TI	R/W1	Transmit Interrupt - When this bit is set, indicates one of the following: • Either all the frames in the transmit list have been transmitted (next descriptor owned by the host), or a frame transmission was aborted because of a locally induced error. The port driver must scan down the list of descriptors to determine the exact cause. The transmission process is placed in the suspended state. Section 3.19.5 explains the transmission process state transitions. To resume processing transmit descriptors, the port driver must issue the Poll Demand command. • A frame transmission completed, and TDES1<IC> was set. The transmission process remains in the running state unless the next descriptor is owned by the host or the frame transmission aborted because of an error. In the latter cases, the Transmission process is placed in the suspended state.
<2>	RI	R/W1	Receive Interrupt - When this bit is set, it indicates that a frame has been placed on the receive list. Frame-specific status information was posted in the descriptor. The reception process remains in the running state.

Table 3–44 (Cont.) CSR5 Bits

Bit	Name	Access	Description
<3>	RU	R/W1	Receive Buffer Unavailable - When this bit is set, it indicates that the next descriptor on the receive list is owned by the host and could not be acquired by the SGEC. The reception process is placed in the suspended state. Section 3.19.4 explains the reception process state transitions. Once set by the SGEC, this bit will not be set again until the SGEC encounters a descriptor it cannot acquire. To resume processing receive descriptors, the host must flip the ownership bit of the descriptor and can issue the Rx Poll Demand command. If no Rx poll demand is issued, the reception process resumes when the next recognized incoming frame is received.
<4>	ME	R/W1	Memory Error - Is set when any of the following occur: • SGEC is the CP-BUS Master and the ERR_L pin is asserted by external logic (generally indicative of a memory problem). • Parity error detected on a host to SGEC CSR write or SGEC read from memory. When a memory error is set, the reception and transmission processes are **aborted** and placed in the stopped state. **NOTE** **At this point, it is mandatory that the port driver issue a reset command and rewrite all CSRs.**
<5>	RW	R/W1	Receive Watchdog Timer interrupt - When this bit is set, it indicates the receive watchdog timer has timed out, indicating that some other node is babbling on the network. Current frame reception is aborted and RDES0<LE> and RDES0<LS> are set. Bit CSR5<RI> is also set. The reception process remains in the running state.
<6>	TW	R/W1	Transmit Watchdog Timer Interrupt - When this bit is set, it indicates the transmit watchdog timer has timed out, indicating the SGEC transmitter was babbling. The transmission process is aborted and placed in the stopped state. (Also reported into the Tx descriptor status TDES0<TO> flag)

Table 3–44 (Cont.) CSR5 Bits

Bit	Name	Access	Description
<7>	BO	R/W1	Boot_Message - When this bit is set, it indicates that the SGEC has detected a boot_message on the serial line and has set the external pin BOOT_L.
<16>	DN	R	Done - When this bit is set, it indicates the SGEC has completed a requested virtual CSR access. After a reset, this bit is set.
<18:17>	OM	R	Operating Mode - These bits indicate the current SGEC operating mode as in the following table:

Value	Meaning
00	Normal operating mode.
01	Internal Loopback - Indicates the SGEC is disengaged from the Ethernet wire. Frames from the transmit list are looped back to the receive list, subject to address filtering. Section 3.19.6 explains this mode of operation.
10	External Loopback - Indicates the SGEC is working in full duplex mode. Frames from the transmit list are transmitted on the Ethernet wire and also looped back to the receive list, subject to address filtering. Section 3.19.6 explains this mode of operation.
11	Reserved for diagnostic purposes.

<23:22>	RS	R	Reception Process State - Indicates the current state of the reception process as follows:

Value	Meaning
00	Stopped
01	Running
10	Suspended

Section 3.19.4 explains the reception process operation and state transitions.

Table 3–44 (Cont.) CSR5 Bits

Bit	Name	Access	Description
<25:24>	TS	R	Transmission Process State - Indicates the current state of the transmission process as follows: **Value** / **Meaning** 00 Stopped 01 Running 10 Suspended Section 3.19.5 explains the transmission process operation and state transitions.
<29:26>	SS	R	Self-test Status - The self-test completion codes are shown in the following table. Only valid if SF is set. **Value** / **Meaning** 0001 ROM error 0010 RAM error 0011 Address filter RAM error 0100 Transmit FIFO error 0101 Receive FIFO error 0110 Self_test loopback error **NOTE** **Self-test takes 25 ms to complete after hardware or software reset.**
<30>	SF	R	Self-test Failed - When this bit is set, it indicates the SGEC self-test has failed. The self-test completion code bits indicate the failure type.
<31>	ID	R	Initialization Done - When this bit is set, it indicates the SGEC has completed the initialization (reset and self-test) sequences and is ready for further commands. When clear, it indicates the SGEC is performing the initialization sequence and ignoring all commands. After the initialization sequence completes, the transmission and reception processes are in the stopped state.

Table 3–45 lists the CSR5 access rules.

Table 3–45 CSR5 Access

Value after **RESET**:	0039FF00 hex
Read access rules:	None
Write access rules:	CSR5<07:01> bits cleared by 1, other bits not writable.

3.17.6.1 CSR5 Status Report

The Status register CSR5 is split into the following two words:

- High word contains the global status of the SGEC, as the initialization status, the DMA and operation mode, and the receive and transmit process states.
- Low word contains the status related to the receive and transmit frames.

Any change of the CSR5 bits <ID>, <SF>, <OM> or <DN>, which is always the result of a host command, is reported without an interrupt.

Any process state change initiated by a host command CSR6<ST> or CSR6<SR> is reported without an interrupt.

In the above two cases, the driver must poll on CSR5 to receive acknowledgement of its command (for example, polling on <ID, SF> after reset or polling on <TS> after a START_TX command).

Any process state change initiated by the SGEC activity is immediately followed by at least one of the CSR5<6:1> interrupts and the interrupt_summary CSR5<IS>.

The SGEC 16 bits internal processor updates the 32 bits CSR5 register in two phases. The high word is modified first, then the low_word is written, which generates an interrupt to the host. In this case, the driver must scan first the CSR5 low_word to get the interrupt status, then the CSR high_word to get the related process state. For example, <TI> interrupt with <TS> = SUSPENDED reports an end of transmission because of an unavailable Tx descriptor.

If the host polls on the process state change, it may detect a change without interrupt because of the small time window separating the CSR5 high_word and low_word updates.

NOTE
Maximum time window is 4*Tcycles of the host clock.

3.17.7 Command and Mode Register (CSR6)

This register is used to establish operating modes and for port driver commands (Figure 3–48).

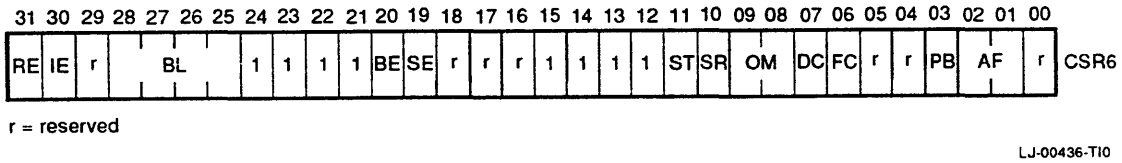


Figure 3–48 CSR6 Format

Table 3–46 lists the CSR6 bit descriptions.

Table 3–46 CSR6 Bits

Bit	Name	Access	Description
<2:1>	AF	R/W	Address Filtering Mode - These bits define the way incoming frames will be address filtered:

Value	Meaning
00	Normal - Incoming frames will be filtered according to the values of the <HP> and <IF> bits of the setup frame descriptor.
01	Promiscuous - All incoming frames will be passed to the host, regardless of the <HP> bit value.
10	All Multicast - All incoming frames with multicast address destinations will be passed to the host. Incoming frames with physical address destinations will be filtered according to the <HP> bit value.
11	Unused - Reserved.

<3>	PB	R/W	Pass Bad Frames Mode - When this bit is set, the SGEC will pass frames that have been damaged by collisions or are too short because of premature reception termination. Both events should have occurred within the collision window (64 bytes), otherwise, other errors will be reported. When clear, these frames will be discarded and never show up in the host receive buffers. **NOTE** **The Pass Bad Frames mode is currently unsupported. Enabling this mode may stack the frame reception.**
<6>	FC	R/W	Force Collision Mode - This bit allows the collision logic to be tested. The chip must be in internal loopback mode for FC to be valid. If FC is set, a collision will be forced during the next transmission attempt. This will result in 16 transmission attempts with excessive collision reported in the transmit descriptor.

Table 3–46 (Cont.) CSR6 Bits

Bit	Name	Access	Description
<7>	DC	R/W	Disable Data Chaining Mode - When this bit is set, no data chaining will occur in reception; frames longer than the current receive buffer will be truncated. RDES0<FS,LS> will always be set. The frame length returned in RDES0<FL> will be the true length of the non-truncated frame while RDES0<BO> will indicate that the frame has been truncated because of buffer overflow. When clear, frames too long for the current receive buffer will be transferred to the next buffer(s) in the receive list.
<9:8>	OM	R/W	Operating Mode - These bits determine the SGEC main operating mode.

Value	Meaning
00	Normal operating mode.
01	Internal Loopback - The SGEC will loopback buffers from the transmit list. The data will be passed from the transmit logic back to the receive logic. The receive logic will treat the looped frame as it would any other frame and subject it to the address filtering and validity check process.
10	External Loopback - The SGEC transmits normally and in addition will enable its receive logic to its own transmissions. The receive logic will treat the looped frame as it would any other frame and subject it to the address filtering and validity check process.
11	Reserved for diagnostic purposes.

Table 3–46 (Cont.) CSR6 Bits

Bit	Name	Access	Description
<10>	SR	R/W	Start/Stop Reception Command - When this bit is set, the reception process is placed in the running state. The SGEC attempts to acquire a descriptor from the receive list and process incoming frames. Descriptor acquisition is attempted from the current position in the list: The address set by CSR3 or the position retained when the Rx process was previously stopped. If no descriptor can be acquired, the reception process enters the suspended state. The start reception command is honored only when the reception process is in the stopped state. The first time this command is issued, CSR3 should already have been written to, otherwise the reception process will remain in the stopped state. When cleared, the reception process is placed in the stopped state after completing reception of the current frame. The next descriptor position in the receive list is saved and becomes the current position after reception is restarted. The stop reception command is honored only when the reception process is in the running or suspended states. Refer to Section 3.19.4 for more information.

Table 3–46 (Cont.) CSR6 Bits

Bit	Name	Access	Description
<11>	ST	R/W	Start/Stop Transmission Command - When this bit is set, the transmission process is placed in the running state and the SGEC checks the transmit list at the current position for a frame to transmit: The address set by CSR4 or the position retained when the Tx process was previously stopped. If it does not find a frame to transmit, the transmission process enters the suspended state. The start transmission command is honored only when the transmission process is in the stopped state. The first time this command is issued, CSR4 should have already been written to, otherwise the transmission process will remain in the stopped state. When cleared, the transmission process is placed in the STOPPED state after completing transmission of the current frame. The next descriptor position in the transmit list is saved and becomes the current position after transmission is restarted. The stop transmission command is honored only when the transmission process is in the running or suspended states. Refer to Section 3.19.5 for more information.
<19>	SE	R/W	Single_cycle Enable Mode - When this bit is set, the SGEC transfers only a single longword or an octaword in a single DMA burst on the host bus.
<20>	BE	R/W	Boot_message Enable Mode - When this bit is set, it enables the boot_message recognition. When the SGEC recognizes an incoming boot message on the serial line, CSR5<BO> is set and the external pin BOOT_L is asserted for a duration of 6*Tcycles (of the host clock).
<28:25>	BL	R/W	Burst Limit Mode - Specifies the maximum number of longwords to be transferred in a single DMA burst on the host bus. When CSR6<SE> is cleared, permissible values are 1,2,4,8 . When SE is set, the only permissible values are 1 and 4: a value of 2 or 8 is respectively forced to 1 or 4. After initialization, the burst limit is set to 1.

Table 3–46 (Cont.) CSR6 Bits

Bit	Name	Access	Description
<30>	IE	R/W	Interrupt Enable Mode - When this bit is set, setting of CSR5 bits 1 through 6 cause an interrupt to be generated.
<31>	RE	R/W	Reset Command - Upon being set, the SGEC aborts all processes and starts the reset sequence. After completing the reset and self test sequence, the SGEC sets bit CSR5<ID>. Clearing this bit has no effect. **NOTE** **The CSR6<RE> value is unpredictable on read after hardware reset.**

Table 3–47 lists the CSR6 access rules.

Table 3–47 CSR6 Access

Value after **RESET**:	83E0F000 hex or 03E0F000 hex
Read access rules:	None
Write access rules:	
<RE, IE, BE>	Unconditional
<BL, SE, OM>	Rx and Tx processes stopped
<FC>	Rx and Tx processes stopped, Internal_Loopback mode
<DC, PB, AF>	Rx stopped
Start_Receive <SR>=1	Rx stopped and CSR3 Initialized
Start_Transmit <ST>=1	Tx stopped and CSR4 Initialized
Stop_Receive <SR>=0	Rx running or suspended
Stop_Transmit <ST>=0	Tx running or suspended

After CSR6 is written, the new value is readable from CSR6. However, if the SGEC status does not match the related write access rules, the new mode setting and command do not take effect and the written information is lost, even if the SGEC matches the right condition later.

3.17.8 System Base Register (CSR7)

This CSR contains the physical starting address of the VAX System Page Table. To make sure that memory will not be corrupt, this register must be loaded by host software before any address translation occurs (Figure 3–49).

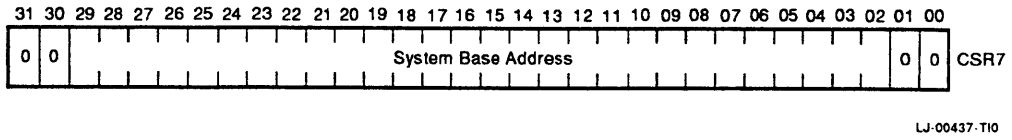


Figure 3–49 CSR7 Format

Table 3–48 lists the CSR7 bit descriptions.

Table 3–48 CSR7 Bits

Bit	Name	Access	Description
<29:00>	SB	R/W	System Base Address - The physical starting address of the VAX System Page Table. Not used if Virtual Addressing (VA) is cleared in all descriptors. This register should be loaded only once after a reset. Subsequent modifications of this register at any other time may cause unpredictable results.

Table 3–49 lists the CSR7 access rules.

Table 3–49 CSR7 Access

Value after **RESET**:	Unpredictable
Read access rules:	None
Write access rules:	Writing once after initialization

3.17.9 Reserved Register (CSR8)

This entire register is reserved.

3.17.10 Watchdog Timers (CSR9)

The SGEC has two timers that restrict the length of time in which the chip can receive or transmit. Figure 3–50 shows the register.

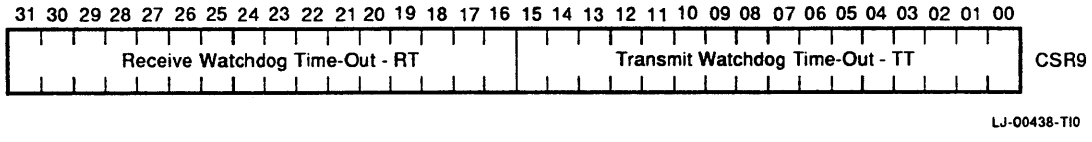


Figure 3–50 CSR9 Format

Table 3–50 lists the CSR9 bit descriptions.

Table 3–50 CSR9 Bits

Bit	Name	Access	Description
<15:00>	TT	R/W	Transmit Watchdog Timeout - The transmit watchdog timer protects the network against babbling SGEC transmissions on top of any such circuitry present in transcievers. If the transmitter stays on for *TT* * 16 cycles of the serial clock, the SGEC will cut off the transmitter and set the (CSR5<TW>) bit. If the timer is set to zero, it will never time-out. The value of TT is an unsigned integer. With a 10 MHz serial clock, this provides a range of 72µs to 100ms. The default value is 1250 corresponding to 2ms. The Tx watchdog timer is programmed only while the transmission process is in the stopped state. **NOTE** **A Tx watchdog value between 1 and 44 is forced to the minimum timeout value of 45 (72µs).**
<31:16>	RT	R/W	Receive Watchdog Timeout - The receive watchdog timer protects the host CPU against babbling transmitters on the network. If the receiver stays on for *RT* * 16 cycles of the serial clock, the SGEC will cut off reception and set the CSR5<RW> bit. If the timer is set to zero, it will never time-out. The value of RT is an unsigned integer. With a 10 MHz serial clock, this provides a range of 72µs to 100ms. The default value is 1250 corresponding to 2ms. The Rx watchdog timer is programmed only while the reception process is in the stopped state. **NOTE** **An Rx watchdog value between 1 and 44 is forced to the minimum timeout value of 45 (72µs).**

Table 3–51 lists the CSR9 access rules.

Table 3–51 CSR9 Access

Value after **RESET**:	00000000 hex
Read access rules:	None
Write access rules:	
Rx watchdog timer	Rx process stopped
Tx watchdog timer	Tx process stopped

These watchdog timers are enabled by default. These timers assume the default values after hardware or software resets.

3.17.11 SGEC Identification and Missed Frame Count (CSR10)

This register contains a missed frame counter and SGEC identification information (Figure 3–51).

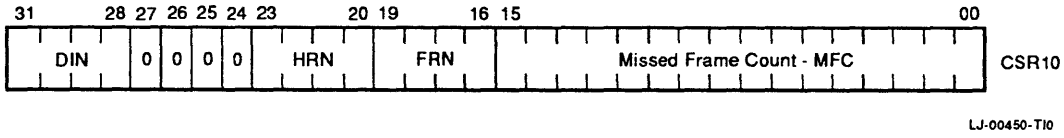


Figure 3–51 SGEC Idendification and Missed Frame Count Format

Table 3–52 lists the CSR10 bit descriptions.

Table 3–52 CSR10 Bits

Data Bit	Name	Access	Description
<15:00>	MFC	R	Missed Frame Count - Counter for the number of frames that were discarded and lost because host receive buffers were unavailable. The counter is cleared when read by the host.
<19:16>	FRN	R	Firmware Revision Number - Internal firmware revision number for this particular SGEC.
<23:20>	HRN	R	Hardware Revision Number - Revision number for this particular SGEC.
<27:24>	0	R	Reserved for future use.
<31:28>	DIN	R	Chip Identification Number - Determines whether the plugged device is a SGEC or another SGEC compatible device (LC_ SGEC,TGEC, and so on). **The SGEC device identification number is 0.**

NOTE

DIN = 0
HRN = 0
FRN = 4

Table 3–53 lists the CSR10 access rules.

Table 3–53 CSR10 Access

Value after **RESET**:	00030000 hex
Read access rules:	Missed_frame counter cleared by read
Write access rules:	Not applicable

3.17.12 Boot Message (CSR11, 12, 13)

These registers contain the boot message verification and processor fields (Figure 3–52).

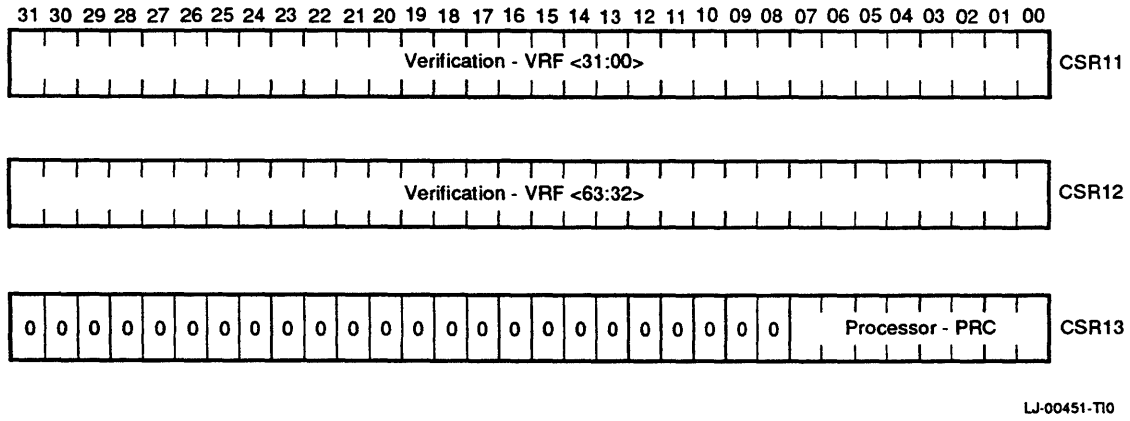


Figure 3–52 Boot Message Registers Format

Table 3–54 lists the CSR11, CSR12, and CSR13 bit descriptions.

Table 3–54 CSR11,12,13 Bits

Data Bit	Name	Access	Description
CSR11 <31:00>	VRF<31:00>	R/W	Boot message verification field <31:00>
CSR12 <31:00>	VRF<63:32>	R/W	Boot message verification field <63:32>
CSR13 <07:00>	PRC	R/W	Boot message processor field <07:00>

NOTE
The least significant bit of the verification field (VRF<0>) corresponds to the first incoming bit of the verification field in the serial boot message.

Table 3–55 lists the CSR11,12,13 access rules.

Table 3–55 CSR11,12,13 Access

Value after **RESET**:	00000000 hex for each of CSR11,CSR12,CSR13
Read access rules:	None
Write access rules:	Boot message disabled (CSR6<BE> = 0)

3.17.13 Diagnostic Registers (CSR14, 15)

These registers are reserved for diagnostic features.

3.18 Descriptors and Buffers Format

The SGEC transfers frame data to and from receive and transmit buffers in host memory. These buffers are pointed to by descriptors which are also resident in host memory.

There are two descriptor lists: one for receive and one for transmit. The starting address of each list is written into CSRs 3 and 4 respectively. A descriptor list is a forward-linked (either implicitly or explicitly) list of descriptors, the last of which may point back to the first entry, thus creating a ring structure. Explicit chaining of descriptors, through setting xDES1<CA> is called *Descriptor Chaining*. The descriptor lists reside in VAX physical memory address space.

NOTE
The SGEC first reads the descriptors, ignoring all unused bits regardless of their state. The only word the SGEC writes back is the first word (xDES0) of each descriptor. Unused bits in xDES0 will be written as "0". Unused bits in xDES1 - xDES3 may be used by the port driver and the SGEC will never disturb them.

A data buffer can contain an entire frame or part of a frame but it cannot contain more than a single frame. Buffers contain only data; buffer status is contained in the descriptor. The term *Data Chaining* is used to refer to frames spanning multiple data buffers. Data Chaining can be enabled or disabled, in reception, through CSR6<DC>. Data buffers reside in either physical or virtual VAX memory space.

NOTE
The virtual to physical address translation is based on the assumption that PTEs are locked in the host memory the time the SGEC owns the related buffer.

NOTE
For best performance in virtual addressing mode, PPTE vectors must not cross a page of the PPTE table.

3.18.1 Receive Descriptors

Figure 3–53 shows the receive descriptor format. The receive descriptors are described in the following sections.

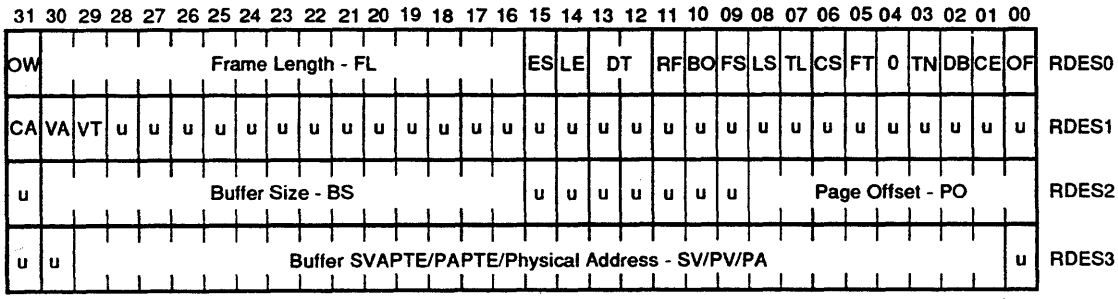


Figure 3–53 Receive Descriptor Format

3.18.1.1 RDES0 Word

RDES0 word contains received frame status, length and descriptor ownership information. Table 3–56 lists the RDES0 bit descriptions.

Table 3–56 RDES0 Bits

Data Bit	Name	Description
<00>	OF	Overflow - When this bit is set, it indicates that received data in this descriptor's buffer was corrupted because of an internal FIFO overflow. This occurs if SGEC DMA requests are not granted before the internal receive FIFO fills up.
<01>	CE	CRC Error - When set, this bit indicates that a CRC error has occurred on the received frame.
<02>	DB	Dribbling Bits - When set, this bit indicates the frame contained a non-integer multiple of eight bits. This error will be reported only if the number of dribbling bits in the last byte is greater than two. Meaningless if RDES0<CS> or RDES0<RF> are set. The CRC check is performed independent of this error. However, only whole bytes are run through the CRC logic. Consequently, received frames with up to six dribbling bits will have this bit set, but if <CE> (or another error indicator) is not set, these frames should be considered valid:

CE	DB	Error
0	0	None
0	1	None
1	0	CRC error
1	1	Alignment error

<03>	TN	Translation Not Valid - When set, this bit indicates that a translation error occurred when the SGEC was translating a VAX virtual buffer address. It will only be set if RDES1<VA> was set. The reception process remains in the running state and attempts to acquire the next descriptor.
<05>	FT	Frame Type - When set, this bit indicates the frame is an Ethernet type frame (Frame Length_Field > 1500). When clear, this bit indicates the frame is an IEEE 802.3 type frame. Meaningless for Runt frames < 14 bytes.
<06>	CS	Collision Seen - When set, this bit indicates the frame was damaged by a collision that occurred after the 64 bytes following the SFD.
<07>	TL	Frame Too Long - When set, this bit indicates the frame length exceeds the maximum Ethernet specified size of 1518 bytes. **NOTE** **Frame Too Long is only a frame length indication and does not cause any frame truncation.**

Table 3–56 (Cont.) RDES0 Bits

Data Bit	Name	Description
<08>	LS	Last Segment - When set, this bit indicates this buffer contains the last segment of a frame and status information is valid.
<09>	FS	First Segment - When set, this bit indicates this buffer contains the first segment of a frame.
<10>	BO	Buffer Overflow - When set, this bit indicates that the frame has been truncated because of a buffer too small to fit the frame size. This bit may only be set if data chaining is disabled (CSR6<DC> = 1).
<11>	RF	Runt Frame - When set, this bit indicates this frame was damaged by a collision or premature termination before the collision window had passed. Runt frames will only be passed on to the host if (CSR6<PB>) is set. Meaningless if RDES0<OF> is set.
<13:12>	DT	Data Type - Indicates the type of frame the buffer contains, according to the following table:

Value	Meaning
00	Serial received frame.
01	Internally looped back frame.
10	Externally looped back frame, serial received frame.
	(The SGEC does not differentiate between looped back and serial received frames. Therefore this information is global and reflects only CSR6<OM>).

<14>	LE	Length Error - When set, this bit indicates a frame truncation caused by one of the following: • The frame segment does not fit within the current buffer and the SGEC does not own the next descriptor. The frame is truncated. • The receive watchdog timer expired. CSR5<RW> is also set.
<15>	ES	Error Summary - The logical "OR" of RDES0 bits OF,CE,TN,CS,TL,LE,RF.
<30:16>	FL	Frame Length - The length in bytes of the received frame. Meaningless if RDES0<LE> is set.
<31>	OW	Own bit - When set, this bit indicates the descriptor is owned by the SGEC. When cleared, this bit indicates the descriptor is owned by the host. The SGEC clears this bit upon completing processing of the descriptor and its associated buffer.

3.18.1.2 RDES1 Word

Table 3–57 lists the RDES1 bit descriptions.

Table 3–57 RDES1 Bits

Data Bit	Name	Description
<29>	VT	Virtual Type - In case of virtual addressing (RDES1<VA> = 1), this bit indicates the type of virtual address translation. When clear, the buffer address RDES3 is interpreted as a System Virtual Address of the Page Table Entry (SVAPTE). When set, the buffer address is interpreted as a Physical Address of the Page Table Entry (PAPTE). Meaningful only if RDES1<VA> is set.
<30>	VA	Virtual Addressing - When set, RDES3 is interpreted as a virtual address. The type of virtual address translation is determined by the RDES1<VT> bit. The SGEC uses RDES3 and RDES2<Page Offset> to perform a VAX virtual address translation process to obtain the physical address of the buffer. When clear, RDES3 is interpreted as the actual physical address of the buffer:

VA	VT	Addressing mode
0	x	Physical
1	0	Virtual - SVAPTE
1	1	Virtual - PAPTE

<31>	CA	Chain Address - When set, RDES3 is interpreted as another descriptor's VAX physical address. This allows the SGEC to process multiple, non-contiguous descriptor lists and explicitly "chain" the lists. Note that contiguous descriptors are implicitly chained. In contrast to what is done for an Rx buffer descriptor, the SGEC does not clear neither the ownership bit RDES0<OW> nor one of the other bits of RDES0 of the chain descriptor after processing. To protect against infinite loop, a chain descriptor pointing back to itself, is seen as **owned by the host**, regardless of the ownership bit state.

3.18.1.3 RDES2 Word

Table 3–58 lists the RDES2 bit descriptions.

Table 3–58 RDES2 Bits

Bit	Name	Description
<08:00>	PO	Page Offset - The byte offset of the buffer within the page. Only meaningful if RDES1<VA> is set. **NOTE** **Receive buffers must be word aligned.**

Table 3–58 (Cont.) RDES2 Bits

Bit	Name	Description
<30:16>	BS	Buffer Size - The size, in bytes, of the data buffer. **NOTE** **Receive buffer size must be an even number of bytes, not shorter than 16 bytes.**

3.18.1.4 RDES3 Word

Table 3–59 lists the RDES3 bit descriptions.

Table 3–59 RDES3 Bits

Data Bit	Name	Description
<31:00>	SV/PV/PA	SVAPTE/PAPTE/Physical Address - When RDES1<VA> is set, RDES3 is interpreted as the address of the Page Table Entry and used in the virtual address translation process. The type of the address System Virtual address (SVAPTE) or Physical Address (PAPTE) is determined by RDES1<VT>. When RDES1<VA> is clear, RDES3 is interpreted as the physical address of the buffer. When RDES1<CA> is set, RDES3 is interpreted as the VAX physical address of another descriptor. **NOTE** **Receive buffers must be word aligned.**

3.18.1.5 Receive Descriptor Status Validity

Table 3–60 summarizes the validity of the receive descriptor status bits regarding the reception completion status.

Table 3–60 Receive Descriptor Status Validity

Reception	Rx Status report						
status	RF	TL	CS	FT	DB	CE	(ES,LE,BO,DT,FS,LS,FL,TN,OF)
Overflow	X	V	X	V	X	X	V
Collision after 512 bits	V	V	V	V	X	X	V
Runt frame	V	V	V	V	X	X	V
Runt frame < 14 bytes	V	V	V	X	X	X	V
Watchdog timeout	V	V	X	V	X	X	V

V - Valid
X - Meaningless

3.18.2 Transmit Descriptors

The transmit descriptor format is shown in Figure 3–54. The transmit descriptors are described in the following sections.

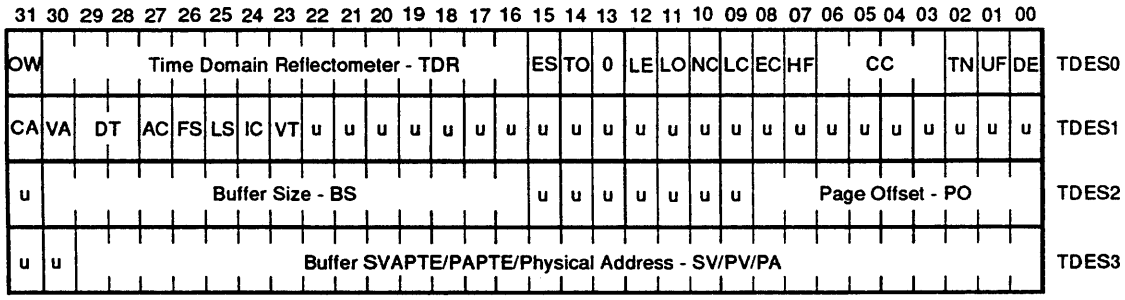


Figure 3–54 Transmit Descriptor Format

3.18.2.1 TDES0 Word

TDES0 word contains transmitted frame status and descriptor ownership information. Table 3–61 lists the TDES0 bit descriptions.

Table 3–61 TDES0 Bits

Data Bit	Name	Description
<00>	DE	Deferred - When set, indicates that the SGEC had to defer while trying to transmit a frame. This condition occurs if the channel is busy when the SGEC is ready to transmit.
<01>	UF	Underflow Error - When set, indicates that the transmitter has truncated a message due to data late from memory. UF indicates that the SGEC encountered an empty transmit FIFO while in the midst of transmitting a frame. The transmission process enters the SUSPENDED state and sets CSR5<TI>.
<02>	TN	Translation Not Valid - When set, indicates that a translation error occurred when the SGEC was translating a VAX virtual buffer address. It may only be set if TDES1<VA> was set. The transmission process enters the SUSPENDED state and sets CSR5<TI>.
<06:03>	CC	Collision Count - A four bit counter indicating the number of collisions that occurred before the transmission attempt succeeded or failed. Meaningless when TDES0<EC> is also set.

Table 3–61 (Cont.) TDES0 Bits

Data Bit	Name	Description
<07>	HF	Heartbeat Fail - When set, this bit indicates heartbeat collision check failure. The transceiver failed to return a collision pulse as a check after the transmission. Some transceivers do not generate heartbeat, and so will always have this bit set. If the transceiver does support heartbeat, this bit indicates transceiver failure. Meaningless if TDES0<UF>.
<08>	EC	Excessive Collisions - When set, this bit indicates that the transmission was aborted because 16 successive collisions occurred while attempting to transmit the current frame.
<09>	LC	Late Collision - When set, this bit indicates frame transmission was aborted because of a late collision. Meaningless if TDES0<UF>.
<10>	NC	No Carrier - When set, this bit indicates the carrier signal from the transceiver was not present during transmission (possible problem in the transceiver or transceiver cable). Meaningless in internal loopback mode (CSR5<OM>=1).
<11>	LO	Loss of Carrier - When set, this bit indicates loss of carrier during transmission (possible short circuit in the Ethernet cable). Meaningless in internal loopback mode (CSR5<OM>=1).
<12>	LE	Length Error - When set, this bit indicates one of the following: • Descriptor unavailable (owned by the host) in the middle of data chained descriptors. • Zero length buffer in the middle of data chained descriptors. • Setup or diagnostic descriptors (Data type TDES1<DT> <> 0) in the middle of data chained descriptors. • Incorrect order of first_segment TDES1<FS> and last_segment TDES1<LS> descriptors in the descriptor list. The transmission process enters the suspended state and sets CSR5<TI>.
<14>	TO	Transmit Watchdog Timeout - When set, this bit indicates the transmit watchdog timer has timed out, indicating the SGEC transmitter was babbling. The interrupt CSR5<TW> is set and the transmission process is *aborted* and placed in the stopped state.
<15>	ES	Error Summary - The logical "OR" of UF, TN, EC, LC, NC, LO, LE and TO.

Table 3–61 (Cont.) TDES0 Bits

Data Bit	Name	Description
<29:16>	TDR	Time Domain Reflectometer - This is a count of bit time and is useful for locating a fault on the cable using the velocity of propagation on the cable. Only valid if TDES0<EC> is also set. Two excessive collisions in a row and with the same or similar (with 20) TDR values indicate a possible cable open.
<31>	OW	Own bit - When set, this bit indicates the descriptor is owned by the SGEC. When cleared, this bit indicates the descriptor is owned by the host. The SGEC clears this bit upon completing processing of the descriptor and its associated buffer.

3.18.2.2 TDES1 Word

Table 3–62 lists the TDES1 bit descriptions.

Table 3–62 TDES1 Bits

Data Bit	Name	Description
<23>	VT	Virtual Type - In case of virtual addressing (TDES1<VA> = 1), this bit indicates the type of virtual address translation. When clear, the buffer address TDES3 is interpreted as a System Virtual Address of the Page Table Entry (SVAPTE). When set, the buffer address is interpreted as a Physical Address of the Page Table Entry (PAPTE). Meaningful only if TDES1<VA> is set.
<24>	IC	Interrupt on Completion - When set, the SGEC will set CSR5<TI> after this frame has been transmitted. To take effect, this bit must be set in the descriptor where LS is set.
<25>	LS	Last Segment - When set, this bit indicates the buffer contains the last segment of a frame.
<26>	FS	First Segment - When set, this bit indicates the buffer contains the first segment of a frame.
<27>	AC	Add CRC disable - When set, the SGEC will not append the CRC to the end of the transmitted frame. To take effect, this bit must be set in the descriptor where FS is set. **NOTE** **If the transmitted frame is shorter than 64 bytes, the SGEC will add the padding field and the CRC regardless of the <AC> flag.**

Table 3–62 (Cont.) TDES1 Bits

Data Bit	Name	Description
<29:28>	DT	Data Type - Indicates the type of data the buffer contains, according to the following table:

Value	Meaning
00	Normal transmit frame data
10	Setup frame - Explained in Section 3.18.3.
11	Diagnostic frame

<30>	VA	Virtual Addressing - When clear, TDES3 is interpreted as the actual physical address of the buffer. The KN220-AA does not support virtual addressing.
<31>	CA	Chain Address - When set, TDES3 is interpreted as another descriptor's VAX physical address. This allows the SGEC to process multiple non-contiguous descriptor lists and explicitly "chain" the lists. Note that contiguous descriptors are implicitly chained. In contrast to what is done for an Rx buffer descriptor, the SGEC neither clears the ownership bit TDES0<OW> nor does it clear any of the other bits of TDES0 of the chain descriptor after processing. To protect against infinite loop, a chain descriptor pointing back itself, is seen as **owned by the host**, regardless of the ownership bit state.

3.18.2.3 TDES2 Word

Table 3–63 lists the TDES2 bit descriptions.

Table 3–63 TDES2 Bits

Data Bit	Name	Description
<08:00>	PO	Page Offset - The byte offset of the buffer within the page. Only meaningful if TDES1<VA> is set. **NOTE** **Transmit buffers may start on arbitrary byte boundaries.**
<30:16>	BS	Buffer Size - The size, in bytes, of the data buffer. If this field is 0, the SGEC will skip over this buffer and ignore it. The frame size is the sum of all BS fields of the frame segments (between and including the descriptors having TDES1<FS> and TDES1<LS> set).

Table 3–63 (Cont.) TDES2 Bits

Data Bit	Name	Description
		NOTE **If the port driver wishes to suppress transmission of a frame, this field must be set to 0 in all descriptors comprising the frame and prior to the SGEC acquiring them. If this rule is not adhered to, corrupted frames might be transmitted.**

3.18.2.4 TDES3 Word

Table 3–64 lists the TDES3 bit descriptions.

Table 3–64 TDES3 Bits

Data Bit	Name	Description
<31:00>	SV/PV/PA	SVAPTE/PAPTE/Physical Address - When TDES1<VA> is set, TDES3 is interpreted as the address of the Page Table Entry and used in the virtual address translation process. The type of the address System Virtual Address (SVAPTE) or Physical Address (PAPTE) is determined by TDES1<VT>. When TDES1<VA> is clear, TDES3 is interpreted as the physical address of the buffer. When TDES1<CA> is set, TDES3 is interpreted as the VAX physical address of another descriptor. **NOTE** **Transmit buffers may start on arbitrary byte boundaries.**

3.18.2.5 Transmit Descriptor Status Validity

Table 3–65 summarizes the validity of the transmit descriptor status bits regarding the transmission completion status.

Table 3–65 Transmit Descriptor Status Validity

Transmission status	Tx Status report						
	LO	NC	LC	EC	HF	CC	(ES,TO,LE,TN,UF,DE)
Underflow	X	X	V	V	X	V	V
Excessive collisions	V	V	V	V	V	X	V
Watchdog timeout	X	V	X	X	X	V	V
Internal Loopback	X	X	V	V	X	V	V

V - Valid
X - Meaningless

3.18.3 Setup Frame

A setup frame defines SGEC Ethernet destination addresses. These addresses will be used to filter all incoming frames. The setup frame is neither transmitted over the Ethernet nor looped back to the receive list. While the setup frame is being processed, the receiver logic will temporarily disengage from the Ethernet wire. The setup frame size is always 128 bytes and must be wholly contained in a single transmit buffer. There are two types of setup frames:

1. Perfect filtering addresses (16) list
2. Imperfect filtering hash bucket (512) heads + one physical address

3.18.3.1 First Setup Frame

A setup frame must be *queued* in place in the transmit list with SGEC ownership to the SGEC before the reception process is started, unless the SGEC operates in promiscuous reception mode.

NOTE

The self-test completes with the SGEC address filtering table fully set to "0." Reception process started without loading a setup frame will reject all the incoming frame except those with a destination physical address = 000000h .

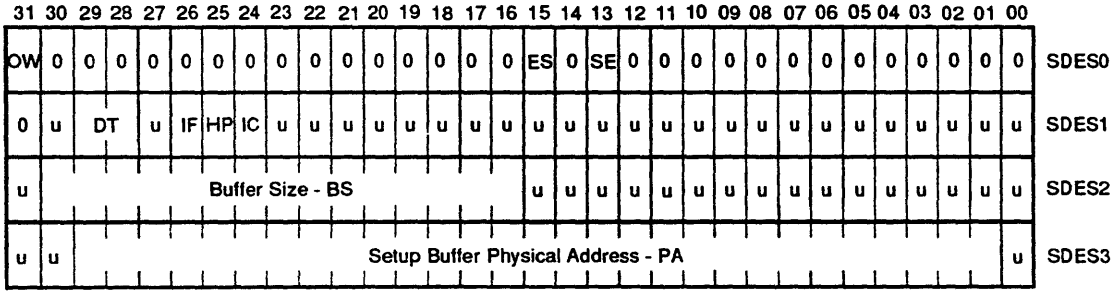
3.18.3.2 Subsequent Setup Frame

Subsequent setup frames may be queued to the SGEC regardless of the reception process state. To process the setup frame, the transmission process must be in the running state. The setup frame will be processed after all preceding frames have been transmitted and after the current frame reception, if any, is completed.

The setup frame does not affect the reception process state, however, during the setup frame processing, the SGEC is disengaged from the Ethernet wire.

3.18.3.3 Setup Frame Descriptor

The setup frame descriptor format is shown in Figure 3–55. The setup frame descriptor is described in the following sections.



0 - SGEC writes as "0"

u - Ignored by the SGEC on read, never written

LJ-00454-TI0

Figure 3–55 Setup Frame Descriptor Format

Table 3–66 lists the setup frame descriptor bit descriptions.

Table 3–66 Setup Frame Descriptor Bits

Frame	Data Bit	Name	Description
SDES0	<13>	SE	Setup Error - When set, this bit indicates the setup frame buffer size is not 128 bytes.
	<15>	ES	Error Summary - Set when SE is set.
	<31>	OW	Own bit - When set, this bit indicates the descriptor is owned by the SGEC. When cleared, this bit indicates the descriptor is owned by the host. The SGEC clears this bit upon completing processing of the descriptor and its associated buffer.
SDES1	<24>	IC	Interrupt on Completion - When set, the SGEC will set CSR5<TI> after this setup frame has been processed.
	<25>	HP	Hash/Perfect filtering mode - When set, the SGEC will interpret the setup frame as a hash table and will do an imperfect address filtering. The imperfect mode is useful when there are more than 16 multicast addresses to listen to. When clear, the SGEC performs a perfect address filter of incoming frames according to the addresses specified in the setup frame.
	<26>	IF	Inverse filtering - When set, the SGEC performs an inverse filtering: the SGEC will receive the incoming frames with destination address not matching the perfect addresses and will reject the frames with destination address matching one of the perfect addresses. Meaningful only for perfect_filtering (SDES1<HP>=0), while promiscuous and all_multicast modes are not selected (CSR6<AF>=0).
	<29:28>	DT	Data Type - The data type must be set to two to indicate setup frame.
SDES2	<30:16>	BS	Buffer Size - The buffer size must be 128.
SDES3	<29:1>	PA	Physical Address - Physical address of setup buffer. **NOTE** **Setup buffer must be word aligned.**

3.18.3.4 Perfect Filtering Setup Frame Buffer

This section describes how the SGEC interprets a setup frame buffer when SDES1<HP> is clear.

The SGEC can store 16 full 48-bit Ethernet destination addresses. It will compare the addresses of any incoming frame to these addresses, and regarding the status of inverse_ filtering flag SDES1<IF>, will reject those which:

- do not match, if SDES1<IF> = 0
- match, if SDES1<IF> = 1

The setup frame must always supply all 16 addresses. Any mix of physical and multicast addresses can be used. Unused addresses should be duplicates of one of the valid addresses. The addresses are formatted as shown in Figure 3–56.

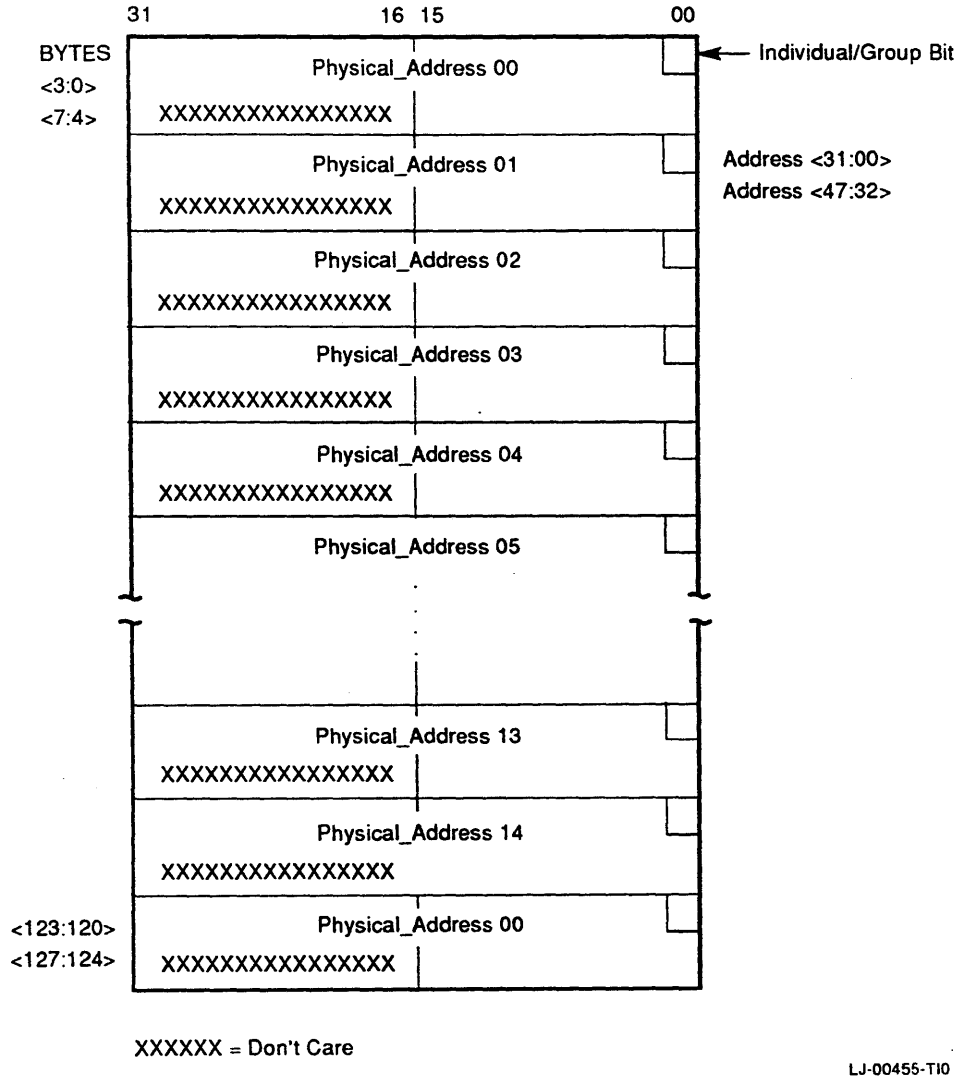


Figure 3–56 Perfect Filtering Setup Frame Buffer Format
The low-order bit of the low-order bytes is the addresses' multicast bit.

Example 3–1 illustrates a perfect filtering setup buffer (fragment).

```
Ethernet addresses to be filtered:
❶  A8-09-65-12-34-76
   09-BC-87-DE-03-15
    .
    .
    .

Setup frame buffer fragment:
❷  126509A8
   00007634
   DE87BC09
   00001503
    .
    .
    .
```

Example 3–1 Perfect Filtering Buffer

❶ Two Ethernet addresses.

❷ Two Ethernet addresses as they would appear in the buffer.

3.18.3.5 Imperfect Filtering Setup Frame Buffer

This section describes how the SGEC interprets a setup frame buffer when SDES1<HP> is set.

The SGEC can store 512 bits, serving as hash bucket heads, and one physical 48-bit Ethernet address. Incoming frames with multicast destination addresses will be subjected to the imperfect filtering. Frames with physical destination addresses will be checked against the single physical address.

For any incoming frame with a multicast destination address, the SGEC applies the standard Ethernet CRC function to the first six bytes containing the destination address, then uses the most significant nine bits of the result as a bit index into the table. If the indexed bit is set, the frame is accepted. If it is cleared, the frame is rejected.

This filtering mode is called imperfect, because multicast frames not addressed to this station may slip through, but it will still cut down the number of frames the host will be presented with.

The format for the hash table and the physical address is shown in Figure 3–57.

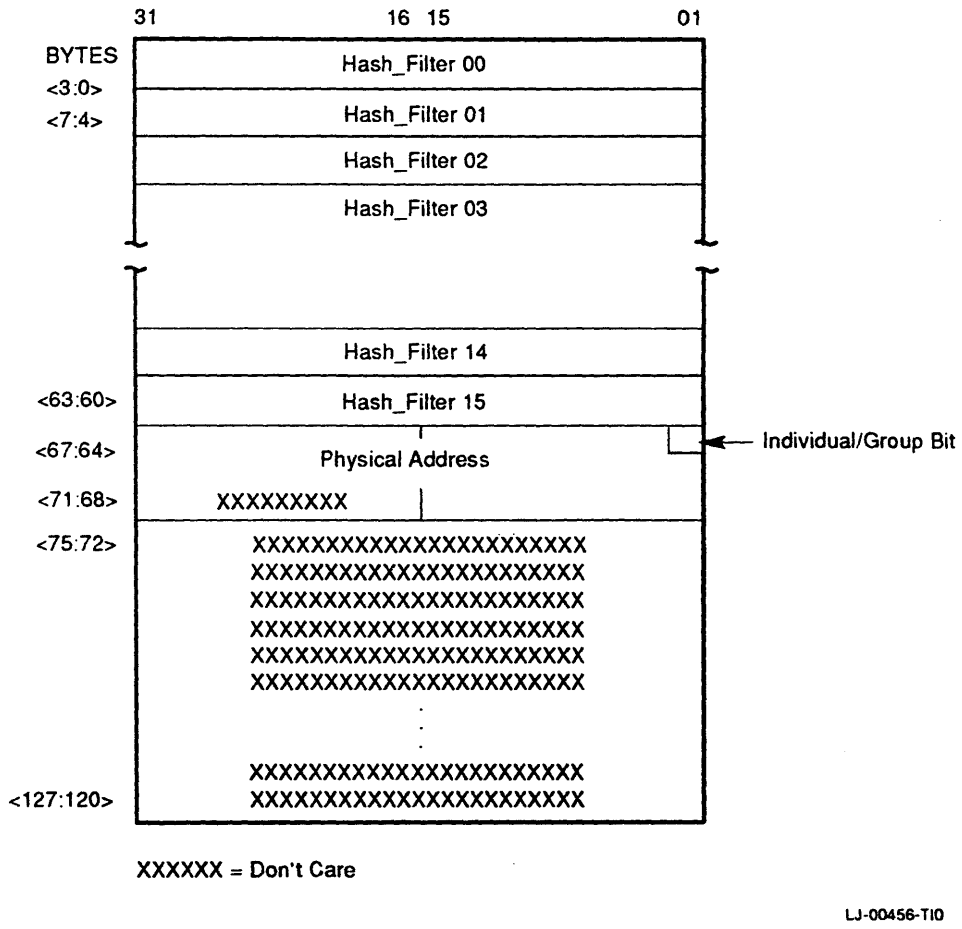


Figure 3–57 Imperfect Filtering Setup Frame Format

Bits are sequentially numbered from right to left and down the table. For example, if CRC (destination address) <8:0> = 33, the SGEC will examine bit #1 in the second longword.

Example 3–2 illustrates an imperfect filtering setup frame buffer.

```
Ethernet addresses to be filtered:
❶  25-00-25-00-27-00
   A3-C5-62-3F-25-87
   D9-C2-C0-99-0B-82
   7D-48-4D-FD-CC-0A
   E7-C1-96-36-89-DD
   61-CC-28-55-D3-C7
   6B-46-0A-55-2D-7E
❷  A8-12-34-35-76-08

Setup frame buffer:
❸  00000000
   10000000
   00000000
   00000000
   00000000
   40000000
   00000080
   00100000
   00000000
   10000000
   00000000
   00000000
   00000000
   00010000
   00000000
   00400000
❹  353412A8
   00000876
```

Example 3–2 Imperfect Filtering Buffer

❶ Ethernet multicast addresses.

❷ An Ethernet physical address.

❸ The first part of an imperfect filter setup frame buffer with set bits for the multicast addresses.

❹ The second part of the buffer with the physical address.

3.19 SGEC Operation

3.19.1 Hardware and Software Reset

The SGEC responds to two types of reset commands: a hardware reset through the RESET_L pin and a software reset command triggered by setting CSR6<RE>. In both cases, the SGEC aborts all ongoing processing and starts the reset sequence. The SGEC restarts and reinitializes all internal states and registers. No internal states are retained, no descriptors are owned, and all the host visible registers are set to "0", except where otherwise noted.

NOTE
The SGEC does not explicitly disown any owned descriptor; therefore, descriptors owned bits might be left in a state indicating SGEC ownership.

The following table indicates the CSR fields which are not set to "0" after reset:

Field	Value
CSR3	Unpredictable
CSR4	Unpredictable
CSR5<DN>	1
CSR6<BL>	1
CSR6<RE>	Unpredictable after hardware reset
	1 after software reset
CSR7	Unpredictable
CSR9	RT = TT = 1250

After the reset sequence completes, the SGEC executes the self-test procedure to do basic sanity checking.

If the self-test completes succesfully, the SGEC initializes the SGEC, then sets the initialization done flag CSR5<ID>.

At the first failure detected in one of the basic tests executed in the self-test routine, the test is aborted and the self-test failure CSR5<SF> is set together with the self-test error status CSR5<SS>which indicates the failure reason.

NOTE
The self-test takes 25ms to complete after hardware or software reset.

If the initialization completes successfully, the SGEC is ready to accept further host commands. Both the reception and transmission processes are placed in the stopped state.

Successive reset commands (either hardware or software) may be issued. The only restriction is that SGEC CSRs should not be accessed during a 1μsecond period following the reset. Access during this period will result in a CP-BUS timeout error. Access to SGEC CSRs during the self-test are permitted; however, only CSR5 reads should be performed.

3.19.2 Interrupts

Interrupts are generated as a result of various events. CSR5 contains all the status bits which may cause an interrupt, provided CSR6<IE> is set. The port driver must clear the interrupt bits (by writing a "1" to the bit position) to enable further interrupts from the same source.

Interrupts are not queued, and if the interrupting event reoccurs before the port driver has responded to it, no additional interrupts will be generated. For example, CSR5<RI> indicates one or more frames were delivered to host memory. The port driver should scan all descriptors from its last recorded position up to the first SGEC owned one.

An interrupt will only be generated once for simultaneous multiple interrupting events. It is the port driver's responsibility to scan CSR5 for the interrupt cause(s). The interrupt will not be regenerated, unless a new interrupting event occurs after the host acknowledged the previous one, and provided the port driver cleared the appropriate CSR5 bit(s). For example, CSR5<TI> and CSR5<RI> may both set, the host acknowledges the interrupt and the port driver begins executing by reading CSR5. Now CSR5<RU> sets. The port driver writes back its copy of CSR5, clearing CSR5<TI>

and CSR5<RI>. After the host IPL is lowered below the SGEC level, another interrupt will be delivered with the CSR5<RU> bit set.

Should the port driver clear all CSR5 set interrupt bits before the interrupt has been acknowledged, the interrupt will be suppressed.

3.19.3 Startup Procedure

A sequence of checks and commands must be performed by the port driver to prepare the SGEC for operation.

1. Wait for the SGEC to complete its initialization sequence by polling on CSR5<ID> and CSR5<SF> (refer to Section 3.17.6 for details).
2. Examine CSR5<SF> to find out whether the SGEC passed its self-test. If it did not, it should be replaced (refer to Section 3.17.6 for details).
3. Write CSR0 to establish system configuration dependent parameters (refer to Section 3.17.2 for details).
4. If the port driver intends to use VAX virtual addresses, CSR7 must be written to identify the system page table to the SGEC (refer to Section 3.17.8 for details).
5. In order to change the default settings of the watchdog timers, the port driver must write to CSR9 (refer to Section 3.17.10 for details).
6. Port driver must create the transmit and receive descriptor lists, then write to CSR3 and CSR4 to provide the SGEC with the starting address of each list. The first descriptor on the transmit list will usually contain a setup frame (refer to Section 3.17.5 for details).
7. Write CSR6 to set global operating parameters and start the reception and transmission processes. The reception and transmission processes enter the running state and attempt to acquire descriptors from the respective descriptor lists and begin processing incoming and outgoing frames (refer to Section 3.17.7 for details). The reception and transmission processes are independent of each other and can be started and stopped separately.

   CAUTION
   If address filtering (either perfect or imperfect) is desired, the reception process should only be started after the setup frame has been processed.
8. The port driver now waits for any SGEC interrupts. If either the reception or transmission processes were suspended, the port driver must issue the poll demand command after it has rectified the suspension cause.

3.19.4 Reception Process

While in the running state, the reception process polls the receive descriptor list, attempting to acquire free descriptors. Incoming frames are processed and placed in acquired descriptor's data buffers, while status information is written to the descriptor RDES0 words. The SGEC always tries to acquire an extra descriptor in anticipation of incoming frames. Descriptor acquisition is attempted under the following conditions:

- Immediately after being placed in the running state through setting of CSR6<SR>.
- The SGEC begins writing frame data to a data buffer pointed to by the current descriptor.
- The last acquired descriptor chained (RDES1<CA> = 1) to another descriptor.

- A virtual translation error was encountered RDES0<TN> while the SGEC was translating the buffer base address of the acquired descriptor .

As incoming frames arrive, the SGEC strips the preamble bits and stores the frame data in the receive FIFO. Concurrently, it performs address filtering according to CSR6 fields AF, HP, and its internal filtering table. If the frame fails the address filtering, it is ignored and purged from the FIFO. Frames which are shorter than 64 bytes because of collision or premature termination are also ignored and purged from the FIFO, unless the CSR6<PB> is set.

After 64 bytes have been received, the SGEC begins transferring the frame data to the buffer pointed to by the current descriptor. If data chaining is enabled (CSR6<DC> clear), the SGEC will write frame data overflowing the current data buffer into successive buffer(s). The SGEC sets the RDES0<FS> and RDES0<LS> in the first and last descriptors, respectively, to delimit the frame. Descriptors are released (RDES0<OW> bit cleared) as their data buffers fill up or the last segment of a frame has been transferred to a buffer.

The SGEC sets RDES0<LS> and the RDES0 status bits in the last descriptor it releases for a frame. After the last descriptor of a frame is released, the SGEC sets CSR5<RI>.

This process is repeated until the SGEC encounters a descriptor flagged as owned by the host. After filling up all previously acquired buffers, the reception sets CSR5<RU> and enters the suspended state. The position in the receive list is retained.

Any incoming frames while in this state will cause the SGEC to fetch the current descriptor in the host memory. If the descripor is now owned by the SGEC, the reception reenters the running state and starts the frame reception.

If the descripor is still owned by the host, the SGEC increments the missed frames counter (CSR10<MFC>) and discards the frame.

Table 3–67 summarizes the reception process state transitions and resulting actions:

Table 3–67 Reception Process State Transitions

From state	Event	To state	Action
Stopped	Start reception command	Running	Receive polling begins from the last list position or from the list head if this is the first start command issued, or if the receive descriptor list address (CSR3) was modified by the port driver.
Running	SGEC attempts acquisition of a descriptor owned by the host	Suspended	CSR5<RU> is set when the last acquired descriptor buffer is consumed. The position in the list is retained.
Running	Stop reception command	Stopped	Reception process is stopped after the current frame, if any, is completely transferred to data buffer(s). The position in the list is retained.
Running	Memory or host bus parity error encountered	Stopped	Reception is cut off and CSR5<ME> is set.

Table 3–67 (Cont.) Reception Process State Transitions

From state	Event	To state	Action
Running	Reset command	Stopped	Reception is cut off.
Suspended	Rx poll demand or incoming frame and available descriptor	Running	Receive polling resumes from the last list position or from the list head if CSR3 was modified by the port driver.
Suspended	Stop reception command	Stopped	None.
Suspended	Reset command	Stopped	None.

3.19.5 Transmission Process

While in the running state, the transmission process polls the transmit descriptor list for any frames to transmit. Frames are built and transmitted on the Ethernet wire. Upon completing frame transmission (or giving up), status information is written to the TDES0 words. Once polling starts, it continues (in sequential or descriptor chained order) until the SGEC encounters a descriptor flagged as owned by the host or an error condition. At this point, the transmission process is placed in the suspended state and CSR5<TI> is set.

CSR5<TI> will also be set after completing transmission of a frame that has TDES1<IC> set in its last descriptor. In this case, the transmission process remains in the running state.

Frames may be data chained and span several buffers. Frames must be delimited by TDES1<FS> and TDES1<LS> in the first and last descriptors, respectively, containing the frame. While in the running state, as the transmission process starts, it first expects a descriptor with TDES1<FS> set. Frame data transfer from the host buffer to the internal FIFO is initiated. Concurrently, if the current frame had TDES1<LS> clear, the transmission process attempts to acquire the next descriptor, expecting TDES1<FS> and TDES1<LS> to be clear indicating an intermediary buffer, or expecting TDES1<LS> to be set, indicating the end of the frame. After the last buffer of the frame has been transmitted, the SGEC writes back final status information to the TDES0 word of the descriptor having TDES1<LS> set, optionally sets CSR5<TI> if TDES1<IC> was set, and repeats the process with the next descriptor(s). Actual frame transmission begins after at least 72 bytes have been transferred to the internal FIFO or a full frame is contained in the FIFO. Descriptors are released (TDES0<OW> bit cleared) as soon as the SGEC is through processing a descriptor.

Transmit polling suspends under the following conditions:

- SGEC reaches a descriptor with TDES0<OW> clear. To resume, the port driver must give descriptor ownership to the SGEC and issue a poll demand command.
- TDES1<FS> and TDES1<LS> are incorrectly paired or out of order. TDES0<LE> will be set.
- A frame transmission is given up because of a locally induced error. The appropriate TDES0 bit is set.

The transmission process enters the suspended state and sets CSR5<TI>. Status information is written to the TDES0 word of the descriptor causing the suspension. The position in the transmit list, in all of the above cases, is retained. The retained position is that of the descriptor following the last descriptor closed (set to host ownership) by the SGEC.

NOTE
The SGEC does not automatically poll the Tx descriptor list and the port driver must explicitly issue a Tx poll demand command after rectifying the suspension cause.

Table 3–68 summarizes the transmission process state transitions:

Table 3–68 Transmission Process State Transitions

From state	Event	To state	Action
Stopped	Start transmission command	Running	Transmit polling begins from the last list position or from the head of the list if this is the first start command issued, or if the transmit descriptor list address (CSR4) was modified by the port driver.
Running	SGEC attempts acquisition of a descriptor owned by the host	Suspended	CSR5<TI> is set. The position in the list is retained.
Running	Out of order delimiting flag (TDES0<FS> or TDES0<LS>) encountered.	Suspended	TDES0<LE> and CSR5<TI> are set. The position in the list is retained.
Running	Frame transmission aborts because of a locally induced error (refer to Table 3–61 for details).	Suspended	Appropriate TDES0 and CSR5<TI> bits are set. The position in the list is retained.
Running	Stop transmission command	Stopped	Transmission process is stopped after the current frame, if any, is transmitted. The position in the list is retained.
Running	Transmit watchdog expires	Stopped	Transmission is cut off and CSR5<TW>, TDES0<TO> are set. The position in the list is retained.
Running	Memory or host bus parity error encountered	Stopped	Transmission is cut off and CSR5<ME> is set.
Running	Reset command	Stopped	Transmission is cut off.

Table 3–68 (Cont.) Transmission Process State Transitions

From state	Event	To state	Action
Suspended	Tx poll demand command	Running	Transmit polling resumes from the last list position or from the list head if CSR4 was modified by the port driver.
Suspended	Stop transmission command	Stopped	None.
Suspended	Reset command	Stopped	None.

3.19.6 Loopback Operations

The SGEC supports two loopback modes:

- Internal loopback

  This mode is generally used to verify correct operations of the SGEC internal logic. While in this mode, the SGEC will take frames from the transmit list and loop them back internally to the receive list. The SGEC is disengaged from the Ethernet wire while in this mode.

- External loopback

  This mode is generally used to verify correct operations up to the Ethernet cable. While in this mode, the SGEC will take frames from the transmit list and transmit them on the Ethernet wire. Concurrently, the SGEC listens to the line which carries its own transmissions and places incoming frames in the receive list.

  NOTE
  Caution should be exercised in this mode as transmitted frames are placed on the Ethernet wire. Furthermore, the SGEC does not check the origin of any incoming frames. Consequently, frames not necessarily originating from the SGEC might make it to the receive buffers.

In either of these modes, all the address filtering and validity checking rules apply. The port driver needs to take the following actions:

1. Place the reception and transmission processes in the stopped state. The port driver must wait for any previously scheduled frame activity to cease. This is done by polling the TS and RS fields in CSR5.
2. Prepare appropriate transmit and receive descriptor lists in host memory. These may follow the existing lists at the point of suspension, or may be new lists that will have to be identified to the SGEC by appropriately writing CSR3 and CSR4.
3. Write to CSR6<OM> according to the desired loopback mode and place the reception and transmission processes in the running state through Start commands.
4. Respond and process any SGEC interrupts, as in normal processing.

To restore normal operations:

1. The port driver must execute step 1 above.
2. Write the OM field in CSR6 with "00".

3.20 DNA CSMA/CD Counters and Events Support

Table 3–69 describes the SGEC features that support the port driver in implementing and reporting the specified counters and events.

Table 3–69 CSMA/CD Counters

Counter	SGEC Feature
Time since counter creation	Supported by the host driver.
Bytes received	Port driver must add up the RDES0<FL> fields of all successfully received frames.
Bytes sent	Port driver must add up the TDES2<BS> fields of all successfully transmitted buffers.
Frames received	Port driver must count the successfully received frames in the receive descriptor list.
Frames sent	Port driver must count the successfully transmitted frames in the transmit descriptor list.
Multicast bytes received	Port driver must add up the RDES0<FL> fields of all successfully received frames with multicast address destinations.
Multicast frames received	Port driver must count the successfully received frames with multicast address destinations.
Frames sent, initially deferred	Port driver must count the successfully transmitted frames with TDES0<DE> set.
Frames sent, single collision	Port driver must count the successfully transmitted frames with TDES0<CC> equal to one.
Frames sent, multiple collisions	Port driver must count the successfully transmitted frames with TDES0<CC> greater than one.
Send failure- Excessive collisions	Port driver must count the transmit descriptors having TDES0<EC> set.
Send failure- Carrier check failed	Port driver must count the transmit descriptors having TDES0<LC> set.
Send failure- Short circuit	Two successive transmit descriptors with the No_ carrier flag TDES0<NC> set, indicates a short circuit.
Send failure- Open circuit	Two successive transmit descriptors with the excessive_collisions flag TDES0<EC> set with the same time domain reflectometer value TDES0<TDR>, indicates an open circuit.
Send failure- Remote Failure to Defer	Flagged as a late collision TDES0<LC> in the transmit descriptors.
Receive failure- Block Check Error	Port driver must count the receive descriptors having RDES0<CE> set with RDES0<DB> cleared.
Receive failure- Framing Error	Port driver must count the receive descriptors having both RDES0<CE> and RDES0<DB> set.
Receive failure- Frame too long	Port driver must count the receive descriptors having RDES0<TL> set.
Unrecognized frame destination	Not applicable.

Table 3–69 (Cont.) CSMA/CD Counters

Counter	SGEC Feature
Data overrun	Port driver must count the receive descriptors having RDES0<OF> set.
System buffer unavailable	Reported in the missed_frame counter CSR10<MFC> (refer to Table 3–52).
User buffer unavailable	Not applicable.
Collision detect check failed	Port driver must count the transmit descriptors having TDES0<HF> set.

CSMA/CD specified events can be reported by the port driver based on Table 3–69. The initialization failed event is reported through CSR5<SF>.

3.21 KN220-AA Q22-bus Interface

The KN220-AA includes a Q22-bus interface implemented through a single VLSI chip called the CQBIC. It contains a CDAL bus to a Q22-bus interface that supports the following:

- A programmable mapping function (scatter-gather map) for translating 22-bit, Q22-bus addresses into 29-bit CDAL bus addresses that allow any page in the Q22-bus memory space to be mapped to any page in main memory.
- A direct mapping function for translating 29-bit CDAL addresses in the local Q22-bus address space and local Q22-bus I/O page into 22-bit, Q22-bus addresses.
- Masked and unmasked longword reads and writes from the CPU to the Q22-bus memory and I/O space and the Q22-bus interface registers. Longword reads and writes of the local Q22-bus memory space are buffered and translated into two-word block mode transfers on the Q22-bus. Longword reads and writes of the local Q22-bus I/O space are buffered and translated into two single-word transfers on the Q22-bus.
- Up to 16-word block mode writes from the Q22-bus to main memory. These words are buffered then transferred to main memory using two asynchronous DMA octaword transfers. For block mode writes of less than 16 words, the words are buffered and transferred to main memory using the most efficient combination of octaword, quadword, and longword asynchronous DMA transfers. The maximum write bandwidth for block mode references is 3.3 Mbytes per second. Block mode reads of main memory from the Q22-bus cause the Q22-bus interface to perform an asynchronous DMA quadword read of main memory and buffer all four words. Therefore on block mode reads the next three words of the block mode read can be delivered without any additional CDAL bus cycles. The maximum read bandwidth for Q22-bus block mode references is 2.4 Mbytes per second. Q22-bus burst mode DMA transfers result in single-word reads and writes of main memory.
- Transfers from the CPU to the local Q22-bus memory space that result in the Q22-bus map translating the address back into main memory (local-miss, global-hit transactions).

NOTE
Q-22 and LMGH transactions that write to R3000 memory do not invalidate the R3000 data cache. Software must make the cache coherent by invalidating tags for any locations that were written to.

The Q22-bus interface contains several registers for Q22-bus control and configuration, interprocessor communication, and error reporting.

The interface also contains Q22-bus interrupt arbitration logic that recognizes Q22-bus interrupt requests BR7-BR4 and translates them into CPU interrupts at levels 17-14.

The Q22-bus interface detects Q22-bus "no sack" timeouts, Q22-bus interrupt acknowledge timeouts, Q22-bus non-existent memory timeouts, main memory errors on DMA accesses from the Q22-bus, and Q22-bus parity errors.

3.21.1 Q22-bus to Main Memory Address Translation

On DMA references to main memory, the 22-bit Q22-bus address must be translated into a 29-bit main memory address. This translation process is performed by the Q22-bus interface by using the Q22-bus map. This map contains 8192 mapping registers (one for each page in the Q22-bus memory space), each of which can map a page (512 bytes) of the Q22-bus memory address space into any of the 128K pages in main memory. Since local I/O space addresses cannot be mapped to Q22-bus pages, the local I/O page is unaccessible to devices on the Q22-bus.

Q22-bus addresses are translated to main memory addresses as shown in Figure 3–58.

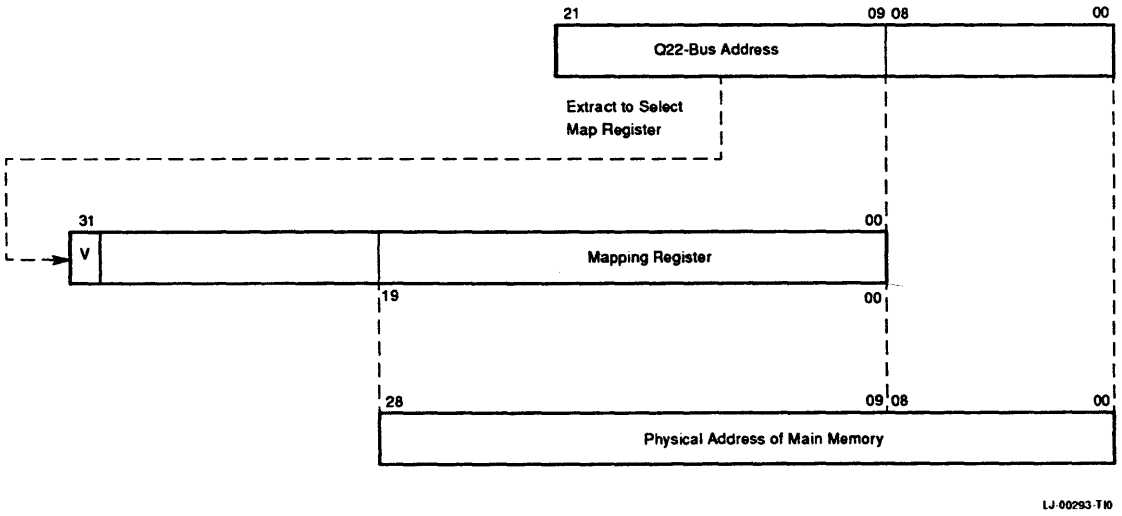


Figure 3–58 Main Memory Addres Translation

At power up time, the Q22-bus map registers, including the valid bits, are undefined. External access to main memory is disabled as long as the interprocessor communication register LM EAE bit is cleared. The Q22-bus interface monitors each Q22-bus cycle and responds if the following three conditions are met:

1. The interprocessor communication register LM EAE bit is set.
2. The valid bit of the selected mapping register is set.
3. During read operations, the mapping register must map into existent main memory, or a Q22-bus timeout occurs. (During write operations, the Q22-bus interface returns Q22-bus BRPLY before checking for existent local memory; the response depends only on conditions 1 and 2 above.)

NOTE
In the case of local-miss, global-hit, the state of the LM EAE bit is ignored.

If the map cache does not contain the needed Q22-bus map register, the Q22-bus interface performs an asychronous DMA read of the Q22-bus map register before proceeding with the Q-22 bus DMA transfer.

3.21.1.1 Q22-bus Map Registers (QMRs)

The Q22-bus map contains 8192 registers that control the mapping of Q22-bus addresses into main memory. Each register maps a page of the Q22-bus memory space into a page of main memory. These registers are implemented in a 32 Kbyte block of main memory but are accessed through the CQBIC chip through a block of addresses in the I/O page.

The local I/O space address of each register was chosen so that register address bits <14:2> are identical to Q22-bus address bits <21:9> of the Q22-bus page that the register maps.

R3000 Register Address	CVAX Register Address	Q22-bus Addresses Mapped (Hex)	Q22-bus Addresses Mapped (Octal)
1008 8000	2008 8000	00 0000 - 00 01FF	00 000 000 - 00 000 777
1008 8004	2008 8004	00 0200 - 00 03FF	00 001 000 - 00 001 777
1008 8008	2008 8008	00 0400 - 00 05FF	00 002 000 - 00 002 777
1008 800C	2008 800C	00 0600 - 00 07FF	00 003 000 - 00 003 777
1008 8010	2008 8010	00 0800 - 00 09FF	00 004 000 - 00 004 777
1008 8014	2008 8014	00 0A00 - 00 0BFF	00 005 000 - 00 005 777
1008 8018	2008 8018	00 0C00 - 00 0DFF	00 006 000 - 00 006 777
1008 801C	2008 801C	00 0E00 - 00 0FFF	00 007 000 - 00 007 777
.	.	.	.
.	.	.	.
.	.	.	.
1008 FFF0	2008 FFF0	3F F800 - 3F F9FF	17 774 000 - 17 774 777
1008 FFF4	2008 FFF4	3F FA00 - 3F FBFF	17 775 000 - 17 775 777
1008 FFF8	2008 FFF8	3F FC00 - 3F FDFF	17 776 000 - 17 776 777
1008 FFFC	2008 FFFC	3F FE00 - 3F FFFF	17 776 000 - 17 777 777

The Q22-bus map registers (QMRs) format is shown in Figure 3–59.

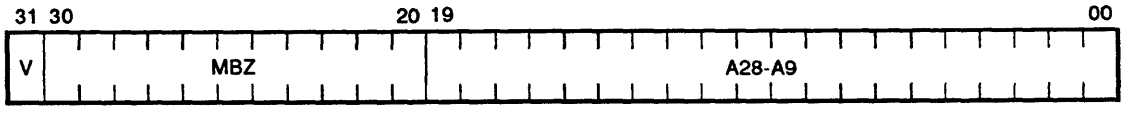


Figure 3–59 Q22-bus Map Registers Format

Table 3–70 lists the Q22-bus map register bit descriptions.

Table 3–70 Q22-bus Map Registers Format Bit Descriptions

Data Bit	Name	Description
<31>	V	Valid. Read/Write. When a Q22-bus map register is selected by bits <21:9> of the Q22-bus address, the valid bit determines whether mapping is enabled for that Q22-bus page. If the valid bit is set, the mapping is enabled and Q22-bus addresses within the page controlled by the register are mapped into the main memory page determined by bits <28:9>. If the valid bit is clear, the mapping register is disabled and the Q22-bus interface does not respond to addresses within that page. This bit is undefined on power up and the negation of DCOK when SCR<7> is clear.
<30:20>	Unused	These bits always read as 0 and must be written as 0.
<19:0>	A28-A9	Address Bits <28:9>. Read/Write. When a Q22-bus map register is selected by a Q22-bus address and if that register's valid bit is set, then these 20 bits are used as main memory address bits <28:9>. Q22-bus address bits <8:0> are used as main memory address bits <8:0>. QMR<19 should always be set to 0 since only 512 Mbytes of main memory is supported. These bits are undefined on power up and the negation of DCOK when SCR<7> is clear.

3.21.1.2 Accessing the Q22-bus Map Registers

Although the CPU accesses the Q22-bus map registers through aligned unmasked longword references to the local I/O page (addresses R3000 10088000-1008FFFC, CVAX 20088000-2008FFFC), the map actually resides in a 32 Kbytes block of main memory. The starting address of this block is controlled by the contents of the Q22-bus map base register. The Q22-bus interface also contains a 16-entry fully associative memory accesses required for address translation.

NOTE

The system software must protect the pages of memory that contain the Q22-bus map from direct accesses that corrupt the map or cause the entries in the Q22-bus map cache to become stale. Either of these conditions will result in the incorrect operation of the mapping function.

When the CPU accesses the Q22-bus map through the local I/O page addresses, the Q22-bus interface reads or writes the map in main memory. The Q22-bus interface does not have to gain Q22-bus mastership when accessing the Q22-bus map. Since these addresses are in the local I/O space, they are not accessible from the Q22-bus.

On a Q22-bus map read by the CPU, the Q22-bus interface decodes the local I/O space address (R3000 10088000-1008FFFC, CVAX 20088000-2008FFFC). If the register is in the Q22-bus map cache, the Q22-bus interface internally resolves any conflicts between CPU and Q22-bus transactions (if both are attempting to access the Q22-bus map cache entries at the same time), then return to the data. If the map register is not in the map cache, the Q22-bus interface forces the CPU to retry, acquire the CDAL bus and perform an asynchronous DMA read of the map register. On completion of the read, the CPU is provided with the data when its read operation is retried. A map read by the CPU does not cause the register that was read to be stored in the map cache.

On a Q22-bus map write by the CPU, the Q22-bus interface latches the data, then on the completion of the CPU write, acquires the CDAL bus and performs an asynchronous DMA write to the map register. If the map register is in the Q22-bus map cache, then the CAMValid bit for that entry will be cleared to prevent the entry from becoming stale. A Q22-bus map write by the CPU does not update any cached copies of the Q22-bus map register.

3.21.1.3 The Q22-bus Map Cache

To speed up the process of translating Q22-bus addresses to main memory addresses, the Q22-bus interface utilizes a fully associative 16-entry Q22-bus map cache that is implemented in the CQBIC chip.

If a DMA transfer ends on a VAX page (512 bytes) boundry, the Q22-bus interface prefetchs the mapping register required to translate the next page and load it into the cache before starting a new DMA transfer. This allows Q22-bus block mode DMA transfers that cross page boundries to proceed without delay. The replacement algorithm for updating the Q22-bus map cache is FIFO.

The cached copy of the Q22-bus map register is used for the address translation process. If the required map entry for a Q22-bus address (as determined by bits <21:9> of the Q22-bus address) is not in the map cache, then the Q22-bus interface uses the contents of the map base register to access main memory and retrieve the required entry. After obtaining the entry from main memory, the valid bit is checked. If it is set, the entry is stored in the cache and the Q22-bus cycle continues.

The format of a Q22-bus map cache entry is shown in Figure 3–60.

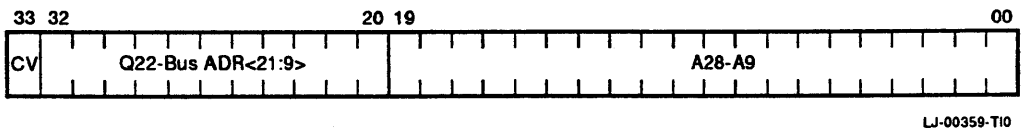


Figure 3–60 Q22-bus Map Cache

Table 3–71 lists the Q22-bus map cache bit descriptions.

Table 3–71 Q22-bus Map Cache Bit Descriptions

Data Bit	Name	Descriptions
<33>	CAMValid	When a mapping register is selected by a Q22-bus address, the CAMValid bit determines whether the cached copy of the mapping register for that address is valid. If the CAMValid bit is set, the mapping register is enabled and addresses within that page can be mapped. If the CAMValid bit is clear, the Q22-bus interface must read the map in local memory to determine if the mapping register is enabled. This bit is cleared on the following: • Power up, the negation of DCOK when SCR <7>is clear • By setting the QMCIA (Q22-bus Map Cache Invalidate All) bit in the interprocessor communication register • On writes to IORESET register (IPR 55) • By a write to the Q22-bus map base register • By writing to the QMR that is being cached
<32:20>	QBUS ADR	These bits contain the Q22-bus address bits <21:9> of the page that this entry maps. This is the content addressable field of the 16 entry cache for determining if the map register for a particular Q22-bus address is in the map cache. These bits are undefined on power up.
<19:0>	Address bits A28-A9	When a mapping register is selected by a Q22-bus address and if that registers CAMValid bit is set, then these 20 bits are used as main memory address bits 28 through 9. Q22-bus address bits 8 through 0 are used as local memory address bits 8 through 0. These bits are undefined on power up.

3.21.2 CDAL Bus to Q22-bus Address Translation

CDAL bus addresses within the local Q22-bus I/O space, R3000 address 10000000-10001FFF and CVAX addresses 20000000-20001FFF, are translated into Q22-bus I/O space addresses by using bits <12:0> of the CDAL address as bits <12:0> of the Q22-bus address and asserting BBS7. Q22-bus address bits <21:13> are driven as 0.

CDAL bus addresses within the local Q22-bus memory space, R3000 addresses 14000000-143FFFFF and CVAX addresses 30000000-303FFFFF, are translated into Q22-bus memory space addresses by using bits <21:0> of the CDAL address as bits <21:0> of the Q22-bus address.

3.21.3 Interprocessor Communications Facility

3.21.3.1 Interprocessor Communication Register (IPCR)

NOTE
The KN220-AA does not support auxiliary mode.

The interprocessor communication register is a 16-bit register that resides in the Q22-bus I/O page address space and can be accessed by any device that can become Q22-bus master (including the KN220-AA itself) (Figure 3–61). The IPCR is implemented in the CQBIC chip and is byte accessible, meaning that a write byte instruction can write to either the low or high byte without affecting the other byte.

The I/O page address of the IPCR varies with the four configurations of arbiter and unsupported auxiliary KN220-AA:

R3000 Address	CVAX Address	Octal 22-Bit Address	Register
1000 1F40	2000 1F40	17 777 500	IPCR
1000 1F42	2000 1F42	17 777 502	Reserved
1000 1F44	2000 1F44	17 777 504	Reserved
1000 1F46	2000 1F46	17 777 506	Reserved

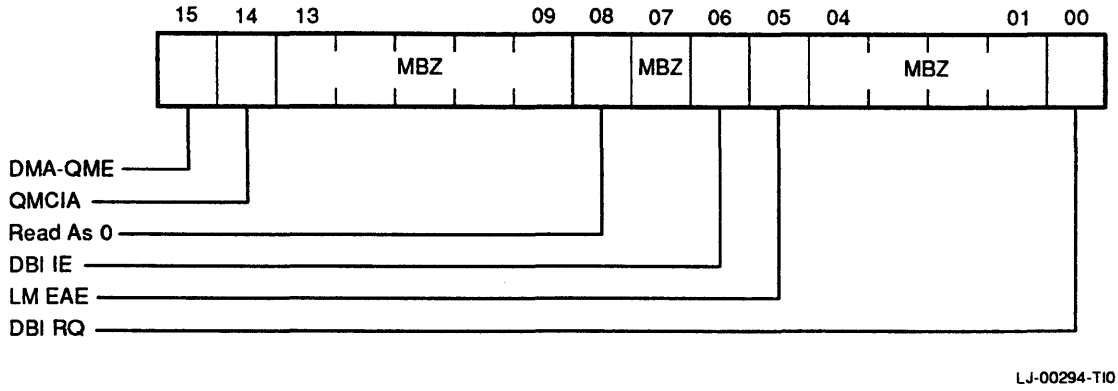


Figure 3–61 Interprocessor Communication Register

Table 3–72 lists the bit descriptions.

Table 3–72 Interprocessor Communication Register Bit Description

Data Bit	Name	Description
<15>	DMA QME	DMA Q22-bus address space memory error. Read/Write to clear. This bit indicates that an error occured when a Q22-bus device was attempting to read main memory. It is set if DMA system error register bit DSER <4> (main memory error) is set or the CDAL bus timer expires. The main memory error bit indicates that an uncorrectable error occurred when an external device was accessing the KN220-AA local memory. The CDAL bus timer expiring indicates that the memory controller did not respond when the Q22-bus interface initiated a DMA transfer. This bit is cleared by the following: • Writing a 1 to it • On power up, by the negation of DCOK when SCR <7> is clear • By writes to the IORESET register (IPR 55) • Whenever DSER<4> is cleared

Table 3–72 (Cont.) Interprocessor Communication Register Bit Description

Data Bit	Name	Description
<14>	QMCIA	Q22-bus Map Cache Invalidate All. Write Only. Writing a 1 to this bit clears the CAMValid bits in the cached copy of the map. This bit always reads as 0. Writing a 0 has no effect.
<13:09>	Unused	Read as 0. Must be written as 0.
<8>	Reserved	Read as 0.
<6>	DBI IE	Doorbell Interrupt Enable. Read/Write when the KN220-AA is Q22-bus master. Read only when another device is Q22-bus master. When set, this bit enables interprocessor doorbell interrupt requests through IPCR<0>. Cleared on power up, by the negation of DCOK when SCR <7> is clear, and writes to IORESET register (IPR 55).
<5>	LM EAE	Local Memory External Access Enable. Read/Write when the KN220-AA is Q22-bus master. Read only when another device is Q22-bus master. When set, this bit enables external access to local memory (through the Q22-bus map). Cleared on power up, by the negation of DCOK when SCR <7> is clear.
<4:1>	Unused	Read as 0. Must be written as 0.
<0>	DBI RQ	Doorbell Interrupt Request. Read/Write. If IPCR<6> (DBI IE) is set, setting this bit generates a doorbell interrupt request. If IPCR<6> is clear, setting this bit has no effect. Clearing this bit has no effect. DBI RQ is cleared when the CPU grants the doorbell interrupt request. DBI RQ is held clear whenever DBI IE is clear. This bit is cleared on power up and the negation of DCOK.

3.21.3.2 Interprocessor Doorbell Interrupts

If the interprocessor communication register DBI IE bit is set, any Q22-bus master can request an interprocessor doorbell interrupt by writing a 1 into IPCR bit <0>.

Interrupt vector	204 (Hex)
Interrupt priority	IPL 14 (Hex); same as BR4 on Q-bus (the interprocessor doorbell is the third highest priority IPL 14 device, directly after the console serial line unit and the programmable timers).

NOTE

Following an interprocessor doorbell interrupt, the KN220-AA CPU sets the IPL to 14. The IPL is set to 17 for external Q22-bus BR4 interrupts.

3.21.4 Q22-bus Interrupt Handling

The KN220-AA responds to interrupt requests BR7-4 with the standard Q22-bus interrupt acknowledge protocol (DIN followed by IAK). The console serial line unit, the programmable timers, and the interprocessor doorbell request interrupts at IPL 14 have priority over all Q22-bus BR4 interrupt requests. After responding to any interrupt request BR7-4, the CVAX CPU sets the processor priority to IPL 17 if it is in control. All BR7-4 interrupt requests are disabled unless software lowers the interrupt priority level.

Interrupt requests from the KN220-AA CVAX interval timer are handled directly by the CVAX CPU. Interval timer interrupt requests have a higher priority than BR6 interrupt requests. After responding to an interval timer interrupt request, the CPU sets the processor priority to IPL 16. Thus, BR7 interrupt requests remain enabled.

3.21.5 Configuring the Q22-bus Map

The KN220-AA implements the Q22-bus map in an 8K longword (32 Kbytes) block of main memory. This map must be configured by the KN220-AA firmware during a processor initialization by writing the base address of the uppermost 32 Kbytes block of good main memory into the Q22-bus map base register. The base of this map must be located on a 32 Kbytes boundary.

NOTE
This 32 Kbytes block of main memory must be protected by the system software. The only access to the map should be through R3000 addresses 10088000-1008FFFC and through CVAX address 20088000-2008FFFC.

3.21.5.1 Q22-bus Map Base Address Register (QBMBR)

The Q22-bus Map Base Address Register controls the main memory location of the 32 Kbytes block of Q22-bus map registers (Figure 3–62). This Read/Write register is accessible by the CPU on a longword boundary only. Bits <31:29,14:0> are unused and should be written as 0 and returns 0 when read.

A write to the map base register flushes the Q22-bus map cache by clearing the CAMValid bits in all the entries.

The contents of this register are undefined on power up, at the negation of DCOK when SCR<7> is clear, and is not affected by BINIT being asserted on the Q22-bus.

- R3000 - 20080010
- CVAX - 20080010

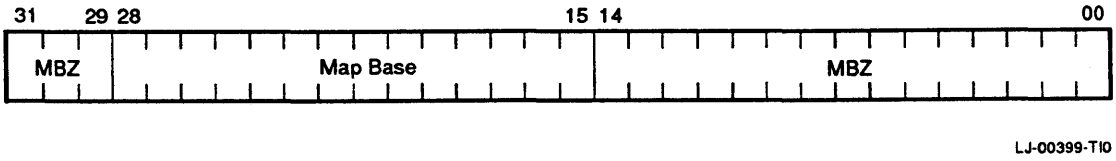


Figure 3–62 Q22-bus Map Base Address Register

3.21.6 System Configuration Register (SCR)

The System Configuration Register contains the processor number that determines the address of the IPCR register, a BHALT enable bit, a power OK flag, and an AUX flag (unused) (Figure 3–63).

The system configuration register (SCR) is longword, word, and byte accessible. Programmable option fields are cleared on power up and by the negation of DCOK when SCR <7> is clear.

- R3000 - 10080000
- CVAX - 20080000

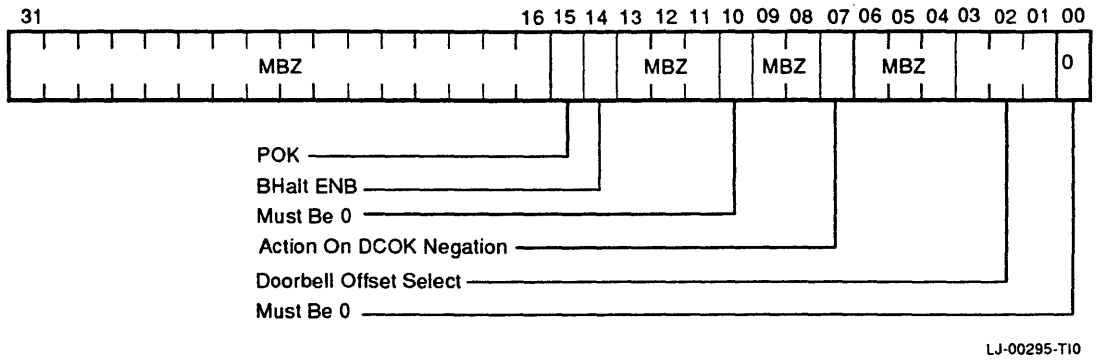


Figure 3–63 System Configuration Register

Table 3–73 lists the system configuration register bit descriptions.

Table 3–73 System Configuration Register Bit Descriptions

Data Bit	Name	Description
<31:16>	Unused	Read as 0. Must be written as 0.
<15>	POK	Power OK. Read Only. Writes have no effect. This bit is set if the Q22-bus BPOK signal is asserted and this bit is clear if it is negated. This bit is cleared on power up and by the negation of DCOK when SCR<7> is clear.
<14>	BHALT EN	BHALT Enable. Read/Write. This bit controls the effect the Q22-bus BHALT signal has on the CPU. When set, asserting the Q22-bus BHALT signal halts the CPU and assert DSER <15>. When cleared, the Q22-bus BHALT signal has no effect. This bit is cleared on power up and by the negation of DCOK when SCR<7> is clear.
<13:11>	Unused	Must be written as 0.
<10>	AUX	Unused. Must be written as 0.
<9:8>	Unused	
<7>	ACTION ON DCOK NEGATION	Read/Write. When cleared, the Q22-bus interface asserts SYSRESET (causing a hardware reset of the board and control to be passed to the resident firmware through the hardware halt procedure with a halt code of 3) when DCOK is negated on the Q22-bus. When set, the Q22-bus interface asserts HALTIN (causing control to be passed to the resident firmware through the hardware halt procedure with a halt code of 2) when DCOK is negated on the Q22-bus. Cleared on power up and the negation of DCOK when SCR<7> is clear.
<6:3>	Unused	Must be written as 0.

Table 3–73 (Cont.) System Configuration Register Bit Descriptions

Data Bit	Name	Description
<3:1>	DOORBELL OFFSET SELECT	Read/Write. These bits determine the IPCR address if the KN220-AA is configured as an auxiliary processor. If the KN220-AA is configured as an arbiter processor, these bits have no effect. If the KN220-AA is configured as an auxiliary processor, programming all zeros disable access of the IPCR from the Q22-bus, including local processor accesses. A doorbell address is selected by programming a non-zero offset into SCR bits <3:1>. A CPU write to the SCR arbitrates for the Q22-bus mastership before allowing the write data to be updated. These bits are cleared on power up and by the negation of DCOK when SCR<7> is clear.
<0>	Unused	Read as 0. Must be written as 0.

3.21.7 DMA System Error Register (DSER)

The DMA System Error Register is one of three registers associated with Q22-bus interface error reporting (Figure 3–64). These registers are located in the local VAX I/O address space and can only be accessed by the local processor. The DMA system error register is implemented in the CQBIC chip and it logs main memory errors on DMA transfers, Q22-bus parity errors, Q22-bus non-existent memory errors, and Q22-bus no-Grant errors. The DMA master error address register contains the address of the page in Q22-bus space, which caused a parity error during an access by the local processor. The slave error address register contains the address of the page in local memory, which caused a memory error during an access by an external device or the processor during a local-miss global-hit transaction. An access by the local processor that the Q22-bus interface maps into main memory provide error status to the processor when the processor does a RETRY for a READ local miss-global hit, or by a MEMERR interrupt in the case of a local-miss global-hit write.

The DSER is a longword, word, or byte accessible Read/Write register available to the local processor. The bits in this register are cleared to 0 on power up, by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register. All bits are set to 1 to record the occurrence of an event. They are cleared by writing a 1, writing a 0 has no effect.

- R3000 - 10080004
- CVAX - 20080004

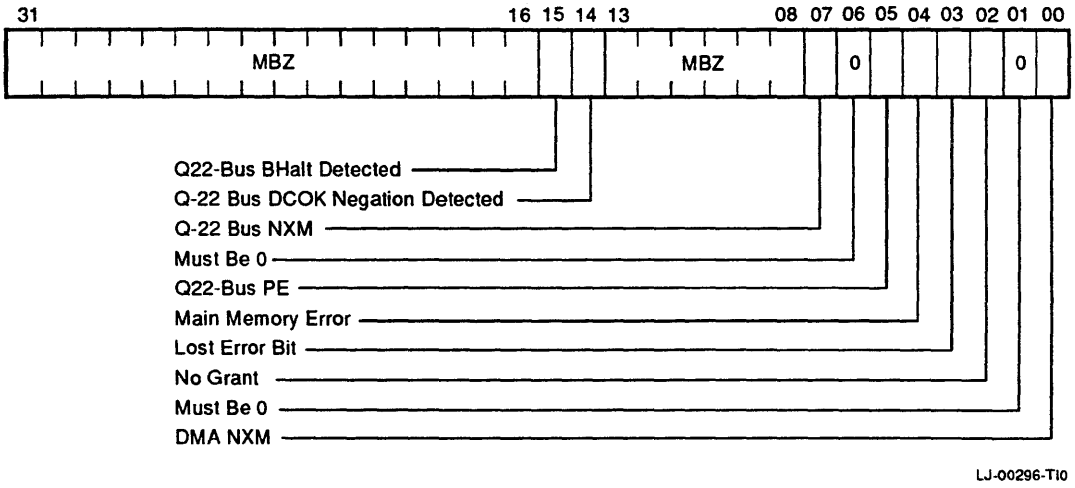


Figure 3–64 DMA System Error Register

Table 3–74 lists the DMA system error register bit descriptions.

Table 3–74 DMA System Error Register Bit Descriptions

Data Bit	Name	Description
<31:16>	Unused	Must be written as 0.
<15>	Q22-bus BHALT DETECTED	Read/Write to clear. This bit is set when the Q22-bus interface detects that the Q22-bus BHALT line was asserted and SCR <14> (BHALT ENABLE) is set. These bits are cleared on power up by the negation of DCOK when SCR <7> is clear, and by writes to the IORESET register (IPR55).
<14>	Q22-bus DCOK NEGATION DETECTED	Read/Write to clear. This bit is set when the Q22-bus interface detects the negation of DCOK on the Q22-bus and SCR <7> (ACTION ON DCOK NEGATION) is set. These bits are cleared on power up by the negation of DCOK when SCR <7> is clear, and by writes to the IORESET register (IPR55).
<13:8>	Unused	Must be written as 0.
<7>	MASTER DMA NXM	Read/Write to clear. This bit is set when the CPU performs a demand Q22-bus read cycle or write cycle that does not reply after 10us. During interrupt acknowledge cycles or request read cycles, this bit is not set. These bits are cleared on power up by the negation of DCOK when SCR <7> is clear, and by writes to the IORESET register (IPR55).
<6>	Unused	Must be written as 0.
<5>	Q22-bus PARITY ERROR	Read/Write to clear. This bit is set when the CPU performs a Q22-bus demand read cycle which returns a parity error. During interrupt acknowledge cycles or request read cycles, this bit is not set. These bits are cleared on power up by the negation of DCOK when SCR <7> is clear, and by writes to the IORESET register (IPR55).

Table 3–74 (Cont.) DMA System Error Register Bit Descriptions

Data Bit	Name	Description
<4>	MAIN MEMORY ERROR	Read/Write to clear. This bit is set if an external Q22-bus device or local miss global hit receives a memory error while reading local memory. The IPCR<15> reports the memory error to the external Q22-bus device. These bits are cleared on power up by the negation of DCOK when SCR <7> is clear, and by writes to the IORESET register (IPR55).
<3>	LOST ERROR	Read/Write to clear. This bit indicates that an error address has been lost because of DSER<7,5,4,0> having been previously set and a subsequent error of either type occurs, which would have normally captured an address and set either DSER<7,5,4,0> flag. These bits are cleared on power up by the negation of DCOK when SCR <7> is clear, and by writes to the IORESET register (IPR55).
<2>	NO GRANT TIMEOUT	Read/Write to clear. This bit is set if the Q22-bus does not return a bus grant within 10ms of the bus request from a CPU demand read cycle or write cycle. During interrupt acknowledge or request read cycles, this bit is not set. These bits are cleared on power up by the negation of DCOK when SCR <7> is clear, and by writes to the IORESET register (IPR55).
<1>	Unused	Must be written as 0.
<0>	DMA NXM	Read/Write to clear. This bit is set on a DMA transfer to a non-existent main memory location. This includes local-miss global-hit cycles and map accesses to non-existent memory. These bits are cleared on power up by the negation of DCOK when SCR <7> is clear, and by writes to the IORESET register (IPR55).

3.21.8 Q22-bus Error Address Register (QBEAR)

The Q22-bus Error Address Register is a read only longword accessible register which is implemented in the CQBIC chip (Figure 3–65). Its contents are valid only if DSER<5> (Q22-bus PARITY ERROR) is set, or if DSER<7> (Q22-bus TIMEOUT) is set.

Reading this register when DSER<5> and DSER<7> are clear returns UNDEFINED results. Additional Q22-bus parity errors that could have set DSER<5>, or Q22-bus timeout errors that could have caused DSER<7> to set, cause DSER<3> to set.

The QBEAR contains the address of the page in Q22-bus space, which caused a parity error during an access by the on-board CPU that set DSER<5> or a master timeout that set DSER<7>.

Q22-bus address bits <21:9> are loaded into QBEAR bits <12:0>. QBEAR bits <31:13> always read as 0.

- R3000 - 10080008
- CVAX - 20080008

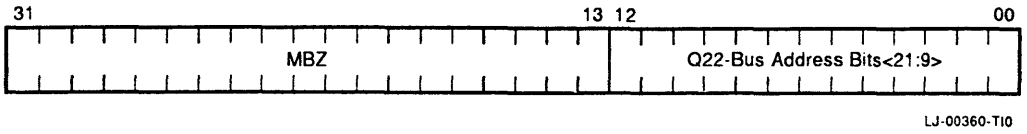


Figure 3–65 Q22-bus Erroe Address Register

NOTE
This is a read only register. If a write is attempted, either a CVAX machine check is generated or an R3000 bus error exception is taken.

3.21.9 DMA Error Address Register (DEAR)

The DMA Error Address Register is a read only longword accessible register that is implemented in the CQBIC chip (Figure 3–66). It contains valid information only when DSER<4> (MAIN MEMORY ERROR is set or when DSER<0> (DMA NXM) is set. Reading this register when DSER<4> and DSER<0> are clear, returns UNDEFINED data.

The DEAR contains the map translated address of the page in local memory which caused a memory error or non existent memory error during an access by an external device or the Q22-bus interface for the CPU during a local-miss global-hit transaction or Q22-bus Map access.

The contents of this register are latched when DSER <4> or DSER<0> is set. Additional main memory errors or non-existent memory errors have no effect on the DEAR until software clears DSER <4> and DSER<0>.

Mapped Q22-bus address bits <28:9> are loaded into DEAR bits <19:0>. DEAR bits <31:20> always read as 0.

- R3000 - 1008000C
- CVAX - 2008000C

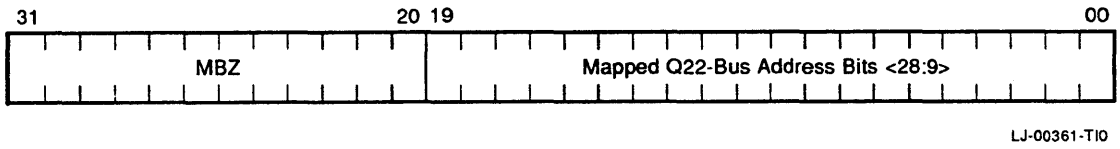


Figure 3–66 DMA Error Address Register

NOTE
This is a read only register. If a write is attempted, either a CVAX machine check is generated or an R3000 bus error exception is taken.

3.21.10 Error Handling

The Q22-bus interface does not generate or check CDAL bus parity.

The Q22-bus interface checks all CPU references to Q22-bus memory and I/O spaces to ensure that nothing but masked and unmasked longword accesses are attempted. Any other type of reference causes a machine check abort (CVAX) or bus error exception (R3000) to be initiated.

The Q22-bus interface maintains several timers to prevent incomplete accesses from hanging the system indefinitely. The timers include:

- A 10us non-existent memory timer for accesses to the Q22-bus memory and I/O spaces
- A 10us "no sack" timer for acknowledgement of Q22-bus DMA grants
- A 10ms "no grant" timer for acquiring the Q22-bus

If there is a non-existent memory (NXM) error (10us timeout) while accessing the Q22-bus on a demand read reference, the following occurs:

1. The associated row in the cache is invalidated
2. DSER<7> is set
3. The address of the Q22-bus page being accessed is captured in QBEAR<12:0>
4. A machine check abort (CVAX) or bus error exception (R3000) is initiated

If there is an NXM error on a prefetch read or an interrupt acknowledge vector read, then the prefetch or interrupt acknowledge reference is aborted but no information is captured and no machine check or bus error exception occurs.

If there is an NXM error on a masked write reference, then DSER<7> is set. The address of the Q22-bus page being accessed is captured in QBEAR<12:0> and an interrupt is generated at IPL 1D through vector 60 (hex).

If the Q22-bus interface does not receive an acknowledgement within 10us after it has granted the Q22-bus ownership of the grant, then the grant is withdrawn. No errors are reported and the Q22-bus interface waits 500ns to clear the Q22-bus grant daisy chain before beginning arbitration again.

If the Q22-bus interface tries to obtain Q22-bus mastership on a CPU demand read reference and does not obtain it within 10ms, then the associated row in the cache is invalidated, DSER<2> is set, and a machine check abort (CVAX) or bus error exception (R3000) is initiated.

The Q22-bus interface also monitors Q22-bus signals BDAL<17:16> while reading information over the Q22-bus so that parity errors, detected by the device where they are read from, are recognized.

If a parity error is detected by another Q22-bus device on a CPU demand read reference to Q22-bus memory or I/O space, then the following occurs:

1. Associated row in the cache is invalidated
2. DSER<5> is set
3. The address of the Q22-bus page being accessed is captured in QBEAR<12:0>
4. A machine check abort is initiated

If a parity error is detected by another Q22-bus device on a prefetch request read by the CPU, the following occurs:

1. The prefetch is aborted
2. The associated row in the cache is invalidated
3. DSER<5> is set
4. The address of the Q22-bus page being accessed is captured in QBEAR<12:0>
5. No exception is generated

The Q22-bus interface also monitors the backplane BPOK signal to detect power failures. If BPOK is negated on the Q22-bus, a power fail trap is generated and the CVAX CPU traps through vector 0C (hex). The R3000 will get an IRQ 4 interrupt request. The state of the Q22-bus BPOK signal can be read from SCR<15>. The Q22-bus interface continues to operate after generating the powerfail trap until DCOK is negated.

3.22 KN220-AA Diagnostic Processor

The diagnostic central processor of the KN220-AA supports the MicroVAX chip subset (plus six additional string instructions) of the VAX instruction set and data types and full VAX memory management. It is implemented through a single VLSI chip called the CVAX. For a complete description of the operation of the diagnostic processor, consult the "MicroVAX Architecture Reference Manual."

3.22.1 Interrupts And Exceptions

Both interrupts and exceptions divert execution from the normal flow of control. An exception is caused by the execution of the current instruction and is typically handled by the current process (for example, an arithmetic overflow). An interrupt is caused by some activity outside the current process and typically transfers control outside the process (for example, an interrupt from an external hardware device). Interrupts can be divided into two classes: non-maskable and maskable.

Non-maskable interrupts cause a halt through the hardware halt procedure that saves the PC, PSL, MAPEN<0> and a halt code in IPRs, raises the processor IPL to 1F and then passes control to the resident firmware. The firmware dispatches the interrupt to the appropriate service routine based on the halt code and hardware event indicators. Non-maskable interrupts cannot be blocked by raising the processor IPL but can be blocked by running out of the halt protected address space (except those non-maskable interrupts that generate a halt code of 3). Non-maskable interrupts with a halt code of 3 cannot be blocked since this halt code is generated after a hardware reset.

Maskable interrupts cause the following:

1. PC and PSL to be saved
2. The processor IPL to be raised to the priority level of the interrupt (except for Q22-bus, mass storage and network interface interrupts where the processor IPL is set to 17 independent of the level at which the interrupt was received)
3. The interrupt to be dispatched to the appropriate service routine through the SCB

The various interrupt conditions for the KN220-AA diagnostic processor are listed on the following table along with their associated priority levels and SCB offsets.

Priority Level	Interrupt Condition	SCB Offset
Non-maskable	BDCOK and BPOK negated then asserted on Q22-bus (power up)	*
	BDCOK negated then asserted while BPOK asserted on Q22-bus (SCR<7> clear)	*
	BINIT asserted on Q22-bus when configured as an auxiliary	**
	BDCOK negated then asserted while BPOK asserted on Q22-bus (SCR<7> set)	**
	BHALT asserted on Q22-bus	**
	BREAK generated by the console device IPCR <8> set when configured as an auxiliary	**
1F	Unused	
1E	BPOK negated on Q22-bus	0C
1D	CDAL Bus parity error	60
	Q22-bus NXM on a write	60
	CDAL Bus timeout during DMA	60
	Main memory NXM errors	60
	Uncorrectable main memory errors	60
1C:1B	Unused	
19:18	Unused	
17	BR7 L asserted	QBus Vector plus 200(hex)
	Memory Interrupts (ECC or NXM)	54
16	Interval Timer Interrupt	C0
	BR6 L asserted	QBus Vector plus 200(hex)
15	BR5 L asserted	QBus Vector plus 200(hex)
	DSSI Mass Storage Interface	C4
	SCSI Mass Storage Interface	1FC
	Network Interface	programmable
14	Console Terminal	F8,F6
	Programmable Timers	78,7C
	Interprocessor Doorbell	204
	BR4 L asserted	QBus Vector plus 200(hex)
13:10	Unused	
0F:01	Software interrupt requests	84-BC

* These conditions generate a hardware halt procedure with a halt code of 3 (hardware reset).

** These conditions generate a hardware halt procedure with a halt code of 2 (external halt).

NOTE

Because the Q22-bus does not allow differentiation between the four bus grant levels (for example, a level 7 device could respond to a level 4 bus grant), the KN220-AA CPU raises the IPL to 17 after responding to interrupts generated by the assertion of either BR7 L, BR6 L, BR5 L, or BR4 L. The KN220-AA maintains the IPL at the priority of the interrupt for all other interrupts.

3.22.1.1 System Control Block (SCB)

The System Control Block (SCB) consists of two pages in main memory or ROM that contains the vectors by which interrupts and exceptions are dispatched to the appropriate service routines. The SCB is pointed to by IPR 17, the system control block base register (Figure 3–67).

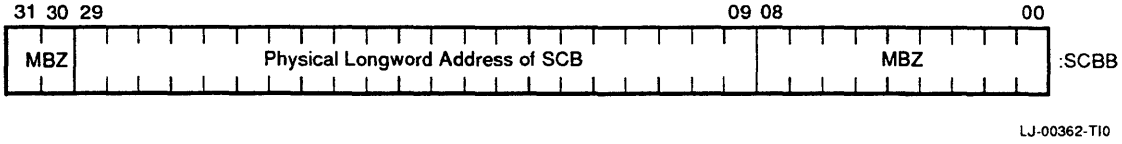


Figure 3–67 System Control Block Base Register

Table 3–75 lists the system control block format.

Table 3–75 System Control Block Format

SCB Offset	Interrupt/ Exception Name	Type	#Param	Notes
00	Unused	-	-	IRQ passive release on other VAX systems
04	Machine check	Abort	4	Parameters depend on error type
08	Kernel stack not valid	Abort	0	Must be serviced on interrupt stack
0C	Power fail	Interrupt	0	IPL is raised to 1E
10	Reserved/privileged instruction	Fault	0	–
14	Customer reserved instruction	Fault	0	XFC instruction
18	Reserved operand	Fault/ Abort	0	Not always recoverable
1C	Reserved addressing mode	Fault	0	–
20	Access control violation	Fault	2	Parameters are virtual address, status code
24	Translation not valid	Fault	2	Parameters are virtual address, status code
28	Trace pending (TP)	Fault	0	–
2C	Breakpoint instruction	Fault	0	–
30	Unused	–	–	Compatibility mode in other VAX systems
34	Arithmetic	Trap/ Fault	1	Parameter is type code
38:3C	Unused	–	–	–

Table 3–75 (Cont.) System Control Block Format

SCB Offset	Interrupt/ Exception Name	Type	#Param	Notes
40	CHMK	Trap	1	Parameter is sign-extended operand word
44	CHME	Trap	1	Parameter is sign-extended operand word
48	CHMS	Trap	1	Parameter is sign-extended operand word
4C	CHMU	Trap	1	Parameter is sign-extended operand word
50	Unused	–	–	–
54	Memory error	Interrupt	3	IPL is 17
58:5C	Unused	–	–	–
60	Memory error	Interrupt	0	IPL is 1D (MEMERR L)
64:6C	Unused	–	–	–
78	Programmable Timer 0	Interrupt	0	IPL is 14
7C	Programmable Timer 1	Interrupt	0	IPL is 14
80	Unused	–	–	–
84	Software level 1	Interrupt	0	–
88	Software level 2	Interrupt	0	Ordinarily used for AST delivery
8C	Software level 3	Interrupt	0	Ordinarily used for process scheduling
90:BC	Software levels 4-15	Interrupt	0	–
C0	Interval timer	Interrupt	0	IPL is 16 (INTTIM L)
C4	DSSI mass storage interface	Interrupt	0	Arbitrated at IPL=15 and Serviced at IPL=17
C8	Emulation start	Fault	10	Same mode exception, FPD = 0; parameters are opcode, PC, specifiers
CC	Emulation continue	Fault	0	Same mode exception, FPD = 1: no parameters
D0	Unused	–	–	–
D4	Network Interface	Interrupt	0	Arbitrated at IPL=15 and serviced at IPL=17
D8:DC	Unused	–	–	–

Table 3–75 (Cont.) System Control Block Format

SCB Offset	Interrupt/ Exception Name	Type	#Param	Notes
E0:EC	Reserved for customer or CSS use	–	–	–
F0:F4	Unused	–	–	Console storage registers on 11/750 and 11/730
F8	Console receiver	Interrupt	0	IPL is 14
FC	Console transmitter	Interrupt	0	IPL is 14
1FC	SCSI mass storage interface	Interrupt	0	Arbitrated at IPL=15 and serviced at IPL=17
400:FFC	Unused	Interrupt	0	–

3.22.1.2 Diagnostic Processor Hardware Detected Errors

The KN220-AA is capable of detecting these types of error conditions during program execution:

- Q22-bus NXM errors indicated by DSER<7> being set.
- Q22-bus no sack errors (no indicator).
- Q22-bus no grant errors indicated by DSER<2> being set.
- Q22-bus the KN220-AA does not support virtual device parity errors indicated by DSER<5> being set.
- CDAL-bus timeout errors indicated by DSER<4>(only on DMA) being set.
- CDAL-bus timeout errors indicated by MEAR<29:28>=01 (only on DMA).
- Main memory NXM errors indicated by DSER<0> (only on DMA) being set.
- Main memory NXM errors indicated by MEAR<0> set.
- Main memory correctable ECC errors indicated by MSER<9> or MSER<0> being set.
- Main memory uncorrectable ECC errors indicated by MSER<10> or MSER<1> being set.

These errors will cause either a machine check exception, a memory error interrupt, or a corrected read data interrupt depending on the severity of the error and the reference type that caused the error.

3.23 KN220-AA Mass Storage Interfaces

3.23.1 DSSI Bus Interface

The KN220-AA I/O module (M7638-Ax) contains a DSSI bus interface that is implemented through the SII chip and four 32K x 8 static RAMs. The interface allows the M7638-Ax to transmit packets of data to, and receive packets of data from, up to seven other DSSI devices (typically RF type disk drives and TF type streaming tape drives). The M7638-Ax also provides the DSSI Bus termination resistors.

This interface contains 27 registers (of which only 16 are used) and 128 Kbytes of 32-bit wide RAM (DSSI Buffer RAM). The SII chip transfers data between the DSSI bus and the DSSI buffer RAM. The processor transfers data between the DSSI buffer RAM and main memory at transfer rate which is 6.5 MByte/sec.

NOTE
All addresses are shown as seen from the R3000 CPU. From the CVAX diagnostic processor, the addresses have to be 2xxx xxxx instead of 1xxx xxxx.

3.23.1.1 DSSI Bus Overview

Some of the major characteristics of the DSSI bus are:

— Eight bit data path

— Eight devices supported

— Parity checking

— Distributed arbitration

— Synchronous operation

— Maximum bandwidth of 4M Bytes/sec

Communication on the DSSI bus is limited to two devices at a time. Each device has a unique ID assigned to it.

When two devices communicate on the DSSI bus, one acts as the initiator, the other as the target. The initiator is the device that starts a DSSI bus transaction. The target device controls the remainder of the DSSI Bus transaction. The direction of data flow is from the initiator to the target.

A DSSI bus transaction consists of six phases:

— WAIT: During this phase, the initiator waits for the bus to become free.

— ARBITRATION: During this phase, control of the bus is taken by the initiator with the highest ID.

— SELECTION: During this phase, the initiator tries to make a logical connection with the target.

— COMMAND OUT: During this phase, the initiator sends the six bytes of command information specified in the command block to the target.

— DATA OUT: During this phase, the initiator sends one to four Kbytes of data to the target.

— STATUS IN: During this phase, the target sends one byte of status information on the transaction to the initiator. The initiator writes this byte to the status word in the command block.

Figure 3–68 shows the DSSI bus sequence.

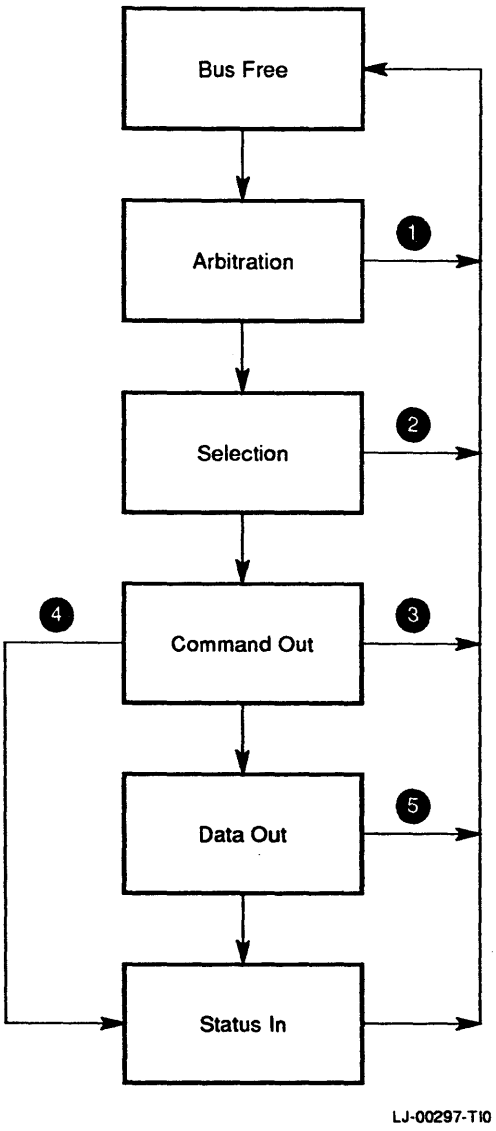


Figure 3–68 DSSI Bus Sequences

The normal path follows vertically downward. Exception paths are as follows:

- The initiator arbitrates and loses.
- The target failed to respond or the target responded with an unexpected bus phase.
- The operation was timed out or the target responded with unexpected phase.
- The target detected a parity error or information mismatch in the command, or the target did not have any buffer space available.
- The operation was timed out or the target responded with an unexpected phase.

3.23.1.2 Target Operation

When the M7638-Ax is functioning as a target device, the SII chip expects receive buffers to be established in the 128 Kbytes DSSI BUFFER RAM (addresses 1010 0000 through 101F FFFF (hex)). Receive buffers must be set up by the processor and start on quadword boundaries (Figure 3–69). These buffers consist of a command block and a receive data block. These buffers are linked together by the first word in the command block and the DSSI_TLP register is used to point to the first buffer in the list.

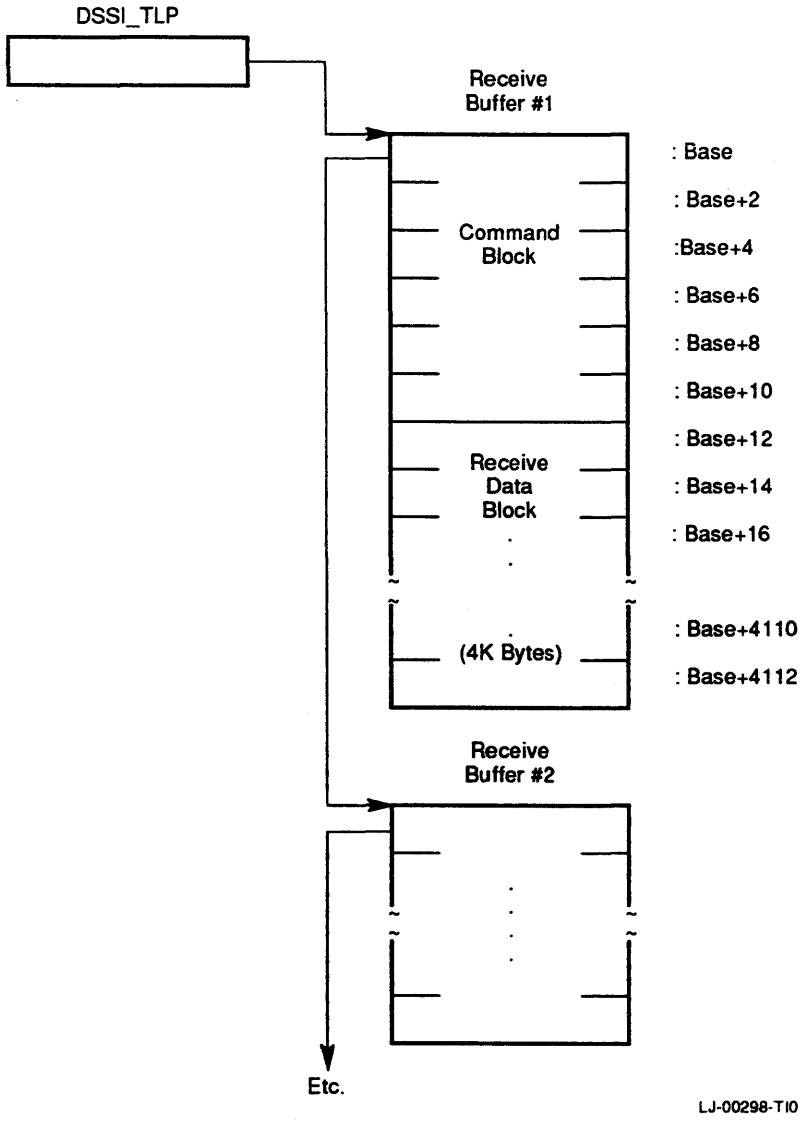


Figure 3–69 Receive Buffers

During target operation, the SII chip uses the DSSI_TLP register to determine the address of the next free receive buffer to be used for this DSSI bus transaction. As the SII chip fills the buffer, it reloads the DSSI_TLP for the next target transaction with the buffer's thread word (the first word in the command block). The target then places the DSSI bus in the Status In phase, sends a status byte to the initiator, and updates the status byte in its buffer's command block.

3.23.1.3 Initiator Operation

When the M7638-Ax is functioning as an initiator device, the SII chip expects transmit buffers to be established in the 128 Kbytes DSSI BUFFER RAM (addresses 1010 0000 through 101F FFFF (hex)). These buffers must be set up by the processor and start on quadword boundaries. These buffers consist of a command block and a transmit data block. These buffers are linked together by the first word in the command block and the DSSI_ILP register is used to point to the first buffer in the list.

3.23.1.3.1 Transmit Data Segment Links

The transmit data block is broken into one or more segments. These segments do not need to reside in contiguous locations in the DSSI buffer RAM and are connected together by the link. Pictorially, the link appears as shown in Figure 3–70.

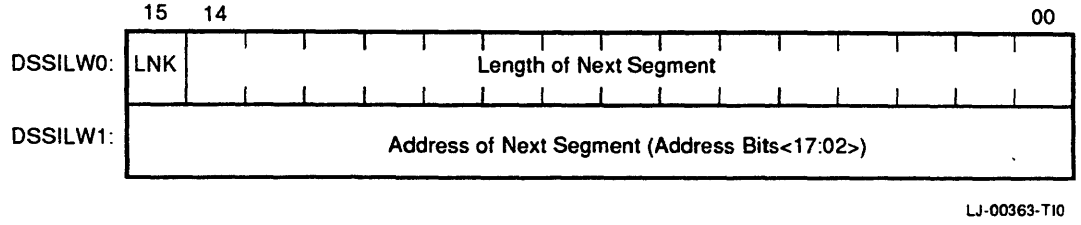


Figure 3–70 Transmit Data Segment Link

3.23.1.3.1.1 DSSI Link Word 0 (DSSILW0)

Table 3–76 lists the transmit data segment link (DSSILW0) bit descriptions.

Table 3–76 DSSI Link Word 0 Bit Descriptions

Data Bit	Name	Description
<15>	LNK	When set, this bit indicates that there is a data segment following the next one. When clear, the next data segment is the last in this data block.
<14:0>	LENGTH OF NEXT SEGMENT	This field contains the number of bytes in the next data segment.

3.23.1.3.1.2 DSSI Link Word 1 (DSSILW1)

Table 3–77 lists the transmit data segment link (DSSILW1) bit descriptions.

Table 3–77 DSSI Link Word 1 Bit Descriptions

Data Bit	Name	Description
<15:0>	ADDRESS OF NEXT SEGMENT	This field contains bits <17:2> of the next quadword aligned data segment.

Each segment of data must be preceded by the above described link. The number of linked segments is only limited by the maximum size of the data block (4 Kbytes). Figure 3–71 shows the transmit buffer.

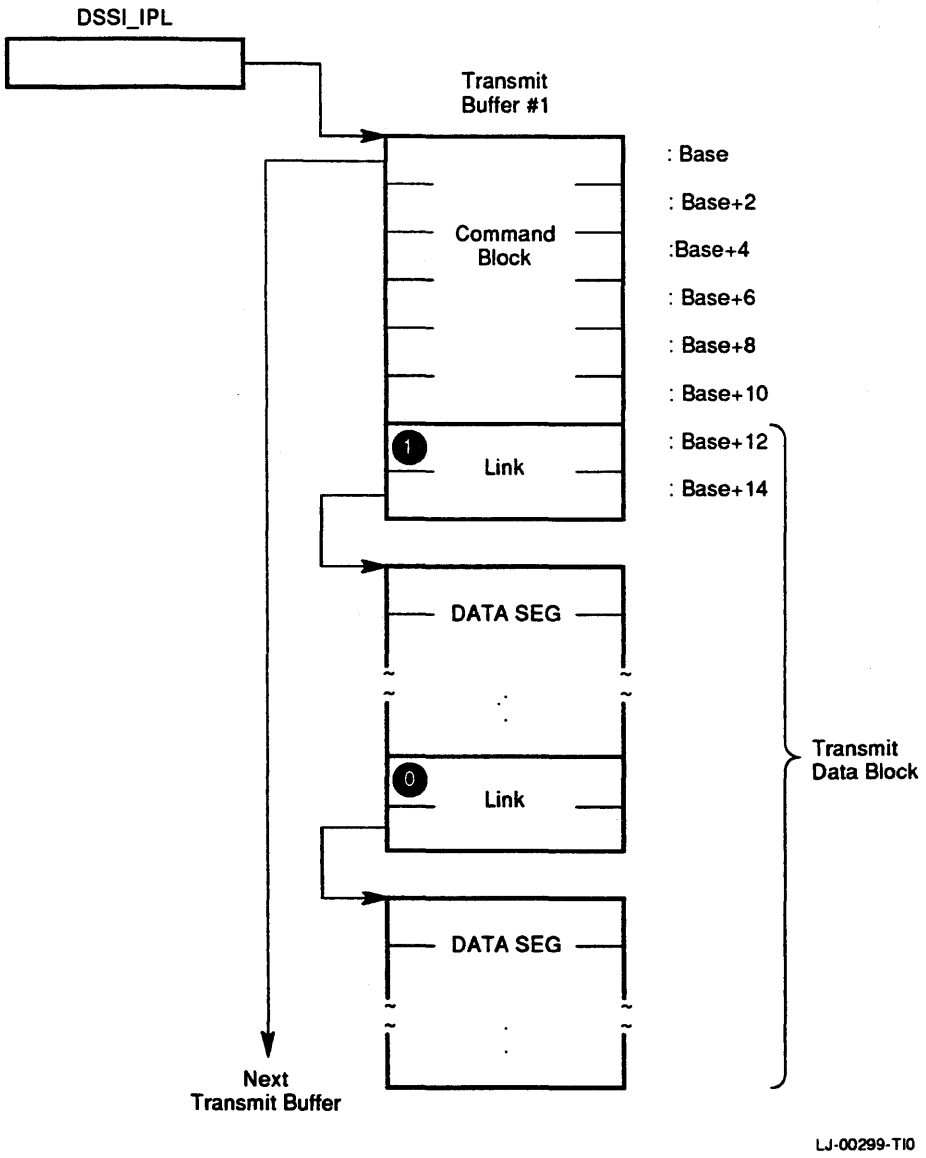


Figure 3–71 Transmit Buffer

When the M7638-Ax is the initiator, the SII chip uses the DSSI_IPL register to determine the address of the transmit buffer to be used for this DSSI transaction. As the SII chip is processing a transmit buffer, it loads an internal register with the second word of the link word as long as the LNK is enabled. This chaining continues until a LNK value of 0 is encountered. Then the SII transfers the next segment and deposits the status of the entire transfer in the status area of the command block. The DSSI_IPL register is then loaded with the buffer's thread word (the first word in the command block) for the next initiator operation. If an error of any kind occurs during the processing of a transmit buffer, the SII stops the transmit operation by clearing the Output Enable bit DSSI_DSCTRL<14>.

3.23.1.4 Adding To A Buffer List

To add new buffers to the DSSI_TLP and DSSI_IPL lists:

1. Fill in the new buffer command block and ensure that the MSB of the status word is 0.
2. Make the thread word of the new buffer 0.
3. Replace the thread word of the last item on either the DSSI_IPL or DSSI_TLP list with the new thread word pointing to the new buffer.
4. If the DSSI_IPL or DSSI_TLP is 0, load it with the address of the new buffer.

3.23.1.5 DSSI Command Block (DSSICB)

The DSSI Command Block is a 12-byte data structure that the processor has to build at the start of all transmit and receive buffers in the DSSI buffer RAM (Figure 3–72).

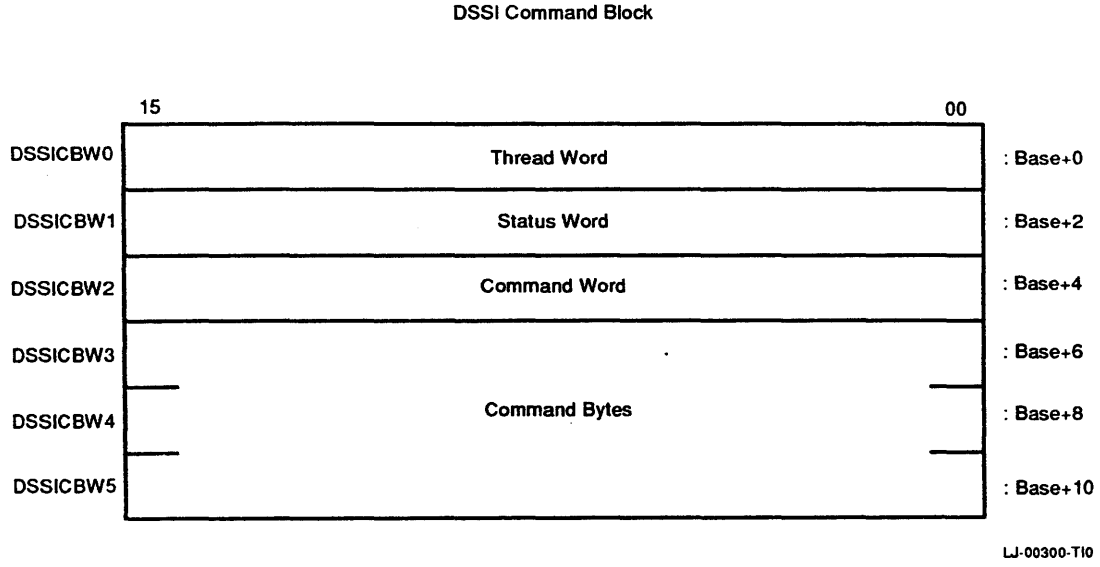


Figure 3–72 DSSI Command Block

3.23.1.5.1 DSSI Command Block Word 0 (DSSICBW0)

Word 0 of the DSSI Command Block also referred to as the thread word, resides in the DSSI buffer RAM at the base address of the DSSI command block (Figure 3–73). The thread word contains bits 17:2 of the base address of the next buffer. Bit 0 of this field must always be set to 0 since buffers must start on a quadword boundary. A thread word of 0 indicates that there are no more buffers.

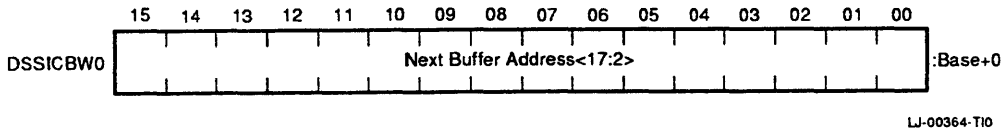


Figure 3–73 DSSI Command Block Word 0

3.23.1.5.2 DSSI Command Block Word 1 (DSSICBW1)

Word 1 of the DSSI Command Block, also referred to as the Status Word, resides in the DSSI buffer RAM at the base+2 address of each DSSI command block (Figure 3–74). This word indicates the status of the current DSSI transaction and is used by the processor to find out which buffers the SII chip have finished processing.

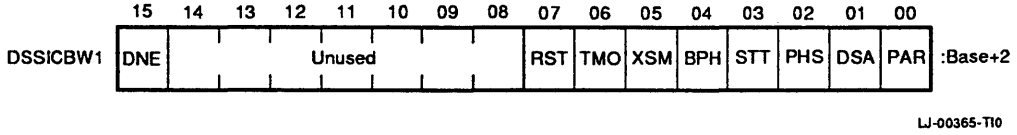


Figure 3–74 DSSI Command Block Word 1

Table 3–78 lists the DSSICBW1 bit descriptions.

Table 3–78 DSSI Command Block Word 1 Bit Descriptions

Data Bit	Name	Description
<15>	DNE	DONE. When set, this bit indicates that the SII chip has used this buffer (either successfully or not). When clear the SII chip has not used this buffer. Note, if this bit is set when the SII chip begins processing a buffer, the buffer is not used.
<14:8>	UNUSED	–
<7>	RST	RESET. When this bit is set, a DSSI device reset the DSSI bus during this buffer's transaction. **NOTE** **If a DSSI bus RESET occurred before the SII chip reached Status In phase, the SII chip clears DSSI_ DSCTRL <7> (OUTPUT ENABLE bit) and interrupts the processor without writing any status.**
<6>	TMO	TIMEOUT. When this bit is set, one of the DSSI_DSTMO timers has expired. **NOTE** **If the timeout occurred before the SII chip reached Status In phase, the SII chip clears DSSI_DSCTRL <7> (OUTPUT ENABLE bit) and interrupts the processor without writing any status.**
<5>	XSM	CHECKSUM. When this bit is set, the received checksum does not agree with that computed by the SII chip. Note the XSM bit is only valid when the M7638-Ax is a target.
<4>	BPH	BAD PHASE. When this bit is set, an illegal DSSI phase was entered by the target. Note the BPH bit is only valid when the M7638-Ax is the initiator.
<3>	STT	STATUS. When this bit is set, ACK was not returned by the target. Note the STT bit is only valid when the M7638-Ax is the initiator.
<2>	PHS	PHASE. When this bit is set, the DSSI bus phase changed before the initiator expected. Note the PHS bit is only valid when the M7638-Ax is the initiator.

Table 3–78 (Cont.) DSSI Command Block Word 1 Bit Descriptions

Data Bit	Name	Description
<1>	DSA	DSSI. When this bit is set, the target detected an error in the command bytes. Note the DSA bit is only valid when the M7638-Ax is the target.
<0>	PAR	PARITY. When this bit is set, a parity error on the DSSI bus was detected.

Please note that the following cases will not cause status to be written in memory:

- DSSI bus RESET occurred before Status In phase was reached.
- Initiator selects a non-existent device (initiator timeout will cause a DSSI bus RESET).
- Target disconnects from the DSSI bus before Status In phase was reached.

3.23.1.5.3 DSSI Command Block Word 2 (DSSICBW2)

Word 2 of the DSSI Command Block, also referred to as the command word, resides in the DSSI buffer RAM at the base+4 address of each DSSI command block (Figure 3–75). This word contains information regarding the transfer.

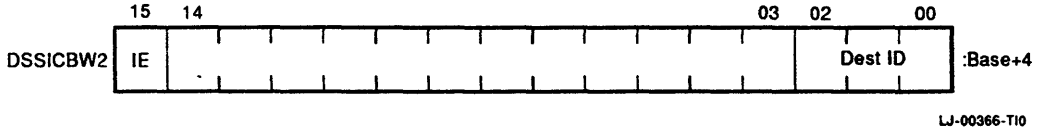


Figure 3–75 DSSI Command Block word 2

Table 3–79 lists the DSSICBW2 bit descriptions.

Table 3–79 DSSI Command Block Word 2 Bit Descriptions

Data Bit	Name	Description
<15>	IE	INTERRUPT ENABLE. When this bit is set, the SII chip interrupts the M7638-Ax upon the completion (successful or not) of this transaction. When this bit is clear, the SII chip does not generate an interrupt. Interrupts are posted at IPL14 with a vector offset of C4 (hex).
<14:3>	UNUSED	–
<2:0>	DEST ID	DESTINATION ID. The ID of the target to be selected. This field is only used when the M7638-Ax is the initiator.

3.23.1.5.4 DSSI Command Block Words 3-5 (DSSICBW3-5)

Words 3-5 of the DSSI Command Block, also referred to as command bytes, reside in the DSSI buffer RAM at the base+6 through base+10 address of each DSSI command block. These six bytes are sent out during the command out phase by the initiator. Some of the information contained in these bytes are: the target and initiator IDs, the number of data bytes that will be transferred by the initiator in the data out phase, and the DSSI opcode.

3.23.1.6 DSSI Registers

The SII chip is very powerful and diverse. The M7638-Ax does not use all its functionality. As a result of this, the M7638-Ax does not use all of the SII's 27 processor visible registers. The following sections contain a description of the 16 registers needed to control the SII chip during DSSI bus operations.

NOTE

Five of the registers are described in the following sections. The other 11 out of 16 registers are not used during DSSI operations and should not be accessed.

3.23.1.6.1 DSSI Control and Status Registers

These five registers are used to configure, control, and monitor the SII chip.

3.23.1.6.1.1 DSSI Control/Status Register (DSSI_CSR)

The Mass Storage Interface Control/Status Register, address 1016 000C (hex), contains control and status information about the general operation of the SII chip in regard to the DSSI bus, including various enable bits (Figure 3–76).

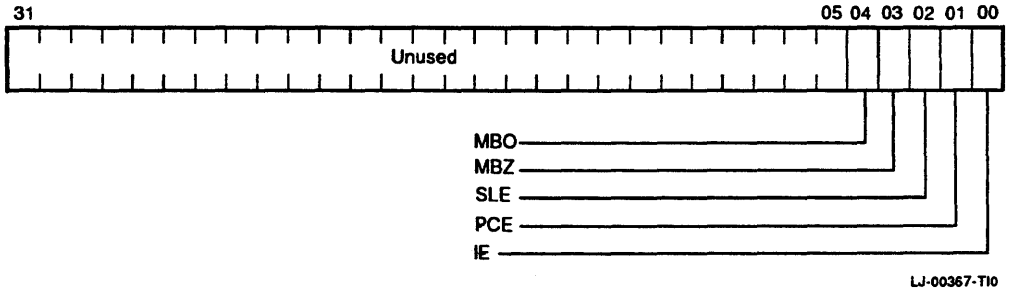


Figure 3–76 DSSI Control/Status Register

Table 3–80 lists the DSSI control/status register bit descriptions.

Table 3–80 DSSI Control/Status Register Bit Description

Data Bit	Name	Description
<31:5>	UNUSED	Reads return undefined results. Writes have no effect.
<4>	MBO	MUST BE ZERO. Read/Write. This bit must read as 0 and be written as 0.
<3>	MBZ	MUST BE ZERO. Read/Write. This bit must read as 0 and be written as 0.
<2>	SLE	SELECTIONS. Read/Write. When this bit is set, the SII chip responds to selections. When this bit is clear, the SII chip will not respond to an initiator trying to select it. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<1>	PCE	PARITY CHECK. Read/Write. When this bit is set, the SII chip reports parity errors. When this bit is clear, the SII chip continues to check parity but will not report any errors during the Status In phase. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).

Table 3–80 (Cont.) DSSI Control/Status Register Bit Description

Data Bit	Name	Description
<0>	IE	INTERRUPT ENABLE. Read/Write. When this bit is set, interrupts are enabled. The SII chip posts interrupts when an error occurs or at the end of a transaction (successful or not). Interrupts are posted at IPL14 with an offset of C4 (hex). These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).

3.23.1.6.1.2 DSSI Control Register (DSSI_DSCTRL)

The Mass Storage Interface DSSI Control Register (DSSI_DSCTRL), address 1016 0044 (hex), contains information to control the SII chip. Figure 3–77 shows the DSSI control register.

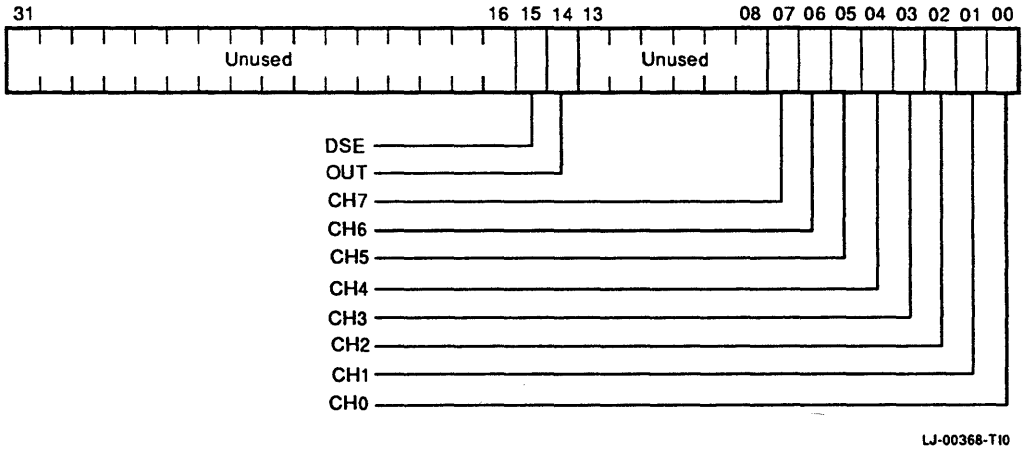


Figure 3–77 DSSI Control Register

Table 3–81 lists the DSSI control register bit descriptions.

Table 3–81 DSSI Control Register Bit Descriptions

Data Bit	Name	Description
<31:16>	UNUSED.	Reads return undefined results. Writes have no effect.
<15>	DSE	DSSI ENABLE. Read/Write. This bit must be set to 1 by the processor for the SII chip to work on a DSSI bus. This bit is cleared by the SII chip if the SII chip selects or is selected by a non-DSSI device, the SII chip is selected with Attention. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).

Table 3–81 (Cont.) DSSI Control Register Bit Descriptions

Data Bit	Name	Description
<14>	OUT	OUTPUT ENABLE. Read/Write. When this bit is set, the SII chip is enabled to send transmit buffers. This bit is cleared by the SII chip if the DSSI_IPL becomes 0, the initiator timer DSSI_DSTMO <3:0> expires, or a transmit buffer is not terminated with ACK. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<13:8>	UNUSED	Reads return undefined results. Writes have no effect.
<7>	CH7	Channel 7. Read/Write. This bit is used to determine if device 7 is a DSSI device. This bit must be set to 1 by the processor. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<6>	CH6	Channel 6. Read/Write. This bit is used to determine if device 6 is a DSSI device. This bit must be set to 1 by the processor. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<5>	CH5	Channel 5. Read/Write. This bit is used to determine if device 5 is a DSSI device. This bit must be set to 1 by the processor. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<4>	CH4	Channel 4. Read/Write. This bit is used to determine if device 4 is a DSSI device. This bit must be set to 1 by the processor. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<3>	CH3	Channel 3. Read/Write. This bit is used to determine if device 3 is a DSSI device. This bit must be set to 1 by the processor. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<2>	CH2	Channel 2. Read/Write. This bit is used to determine if device 2 is a DSSI device. This bit must be set to 1 by the processor. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<1>	CH1	Channel 1. Read/Write. This bit is used to determine if device 1 is a DSSI device. This bit must be set to one by the processor. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<0>	CH0	Channel 0. Read/Write. This bit is used to determine if device 0 is a DSSI device. This bit must be set to 1 by the processor. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).

3.23.1.6.1.3 DSSI Connection Register (DSSI_CSTAT)

The Mass Storage Interface DSSI Connection Register, address 1016 0048 (hex), contains interrupt status related to SII chip connections (Figure 3–78).

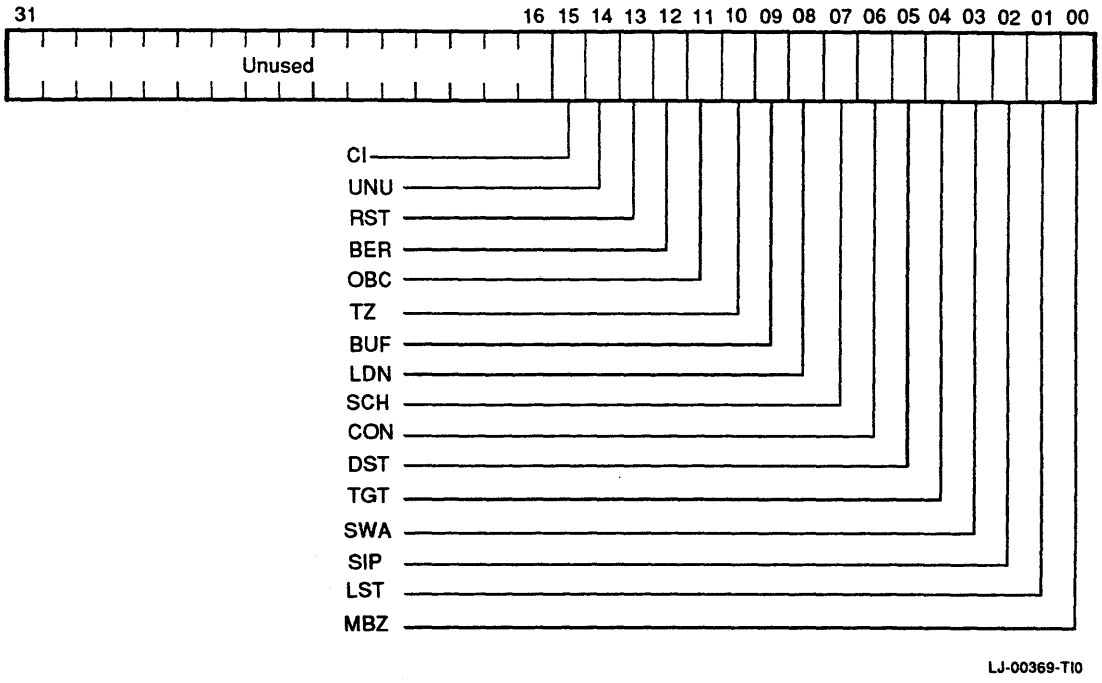


Figure 3–78 DSSI Connection Register

Table 3–82 lists the DSSI connection register bit descriptions.

Table 3–82 DSSI Connection Register Bit Descriptions

Data Bit	Name	Description
<31:16>	UNUSED	Reads return undefined results. Writes have no effect.
<15>	CI	COMPOSITE INTERRUPT. Read only. This bit is the composite error bit of the DSSI_CSTAT register. It is the logical 'OR' of bits DSSI_CSTAT <13:11> and DSSI_CSTAT<9:7>. When this bit is set, the processor will be interrupted at IPL14 with an offset of C4 (hex) if interrupts are enabled. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<14>	UNU	UNUSED. Reads return undefined results. Writes have no effect.
<13>	RST	RESET ASSERTED. Read/Write 1 to clear. When this bit is set, the DSSI bus was reset by one of the eight DSSI devices. The SII chip automatically disconnects itself from the bus and interrupts the processor at IPL14 with an offset of C4 (hex). This bit is write 1 to clear. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).

Table 3–82 (Cont.) DSSI Connection Register Bit Descriptions

Data Bit	Name	Description
<12>	BER	BUS ERROR. Read/Write 1 to clear. This bit is set to 1 on any of the following conditions: 1. Buffer overflow 2. Req/Ack offset exceeded 3. Illegal phase change While this bit is asserted, the SII chip will not receive or transmit data. This bit is write 1 to clear. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<11>	OBC	OUT_EN BIT CLEARED. Read/Write One to clear. This bit is set to 1 on any of the following conditions: 1. The SII chip has received RSTIN. (The DSSI bus has been reset). 2. The DSSI_DSTMO (DSSI_DSTMO<3:0> or DSSI_ DSTMO<7:4>) has expired. 3. As an initiator, the attached target disconnects unexpectedly. This bit is write 1 to clear. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<10>	TZ	TARGET POINTER ZERO. Read only. When set, the DSSI_TLP register contains a value of zero. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<9>	BUF	BUFFER SERVICE. Read/Write 1 to Clear. When this bit is set, the SII chip has begun processing a transmit buffer destined for non-DSSI device. Note, this bit should always be 0 since all devices must be DSSI. This bit is write 1 to clear. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<8>	LDN	LIST ELEMENT DONE. Read/Write 1 to clear. When interrupts are enabled, this bit is set if the SII chip has completed a buffer successfully or not. This bit is write 1 to clear. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<7>	SCH	STATE CHANGE. Read/Write 1 to clear. Set if DSSI_ DSCTRL<15> is cleared causing the SII chip to leave DSSI mode. This bit is write 1 to clear. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).

Table 3–82 (Cont.) DSSI Connection Register Bit Descriptions

Data Bit	Name	Description
<6>	CON	CONNECTED. Read only. When this bit is set, the SII is connected to another device on the DSSI bus. Clear while the SII chip is not connected to another device on the DSSI bus. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<5>	DST	DESTINATION. Read only. When this bit is set, the SII is the destination of the current transaction. In other words, this bit is set if the SII chip was selected by another device on the DSSI bus. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<4>	TGT	TARGET. Read only. When set, the SII chip is operating as a target during the current transaction. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<3>	SWA	SELECTED WITH ATTENTION. Read only. When set, the SII chip was selected with attention. This bit is write 1 to clear. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<2>	SIP	SELECTION IN PROGRESS. Read only. When this bit is set, the SII chip is currently in a selection process. This is useful in determining if the desired target is unavailable. This bit is write 1 to clear. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<1>	LST	LOST. Read only. When this bit is set, the SII lost arbitration. It is cleared by the SII chip when it begins a selection process and on power up the negation of DCOK when SCR<7> is clear or writes to IPR55 (IORESET).
<0>	MBZ	MUST BE ZERO. Read Only. This bit will be read as 0.

3.23.1.6.1.4 DSSI ID Register (DSSI_ID)

The Mass Storage Interface ID Register (DSSI_ID), address 1016 0010 (hex), contains the three bit ID number of the M7638-Ax on the DSSI bus. This value is placed on the DSSI bus during the selection phase so the target knows who selected it. Figure 3–79 shows the DSSI ID register.

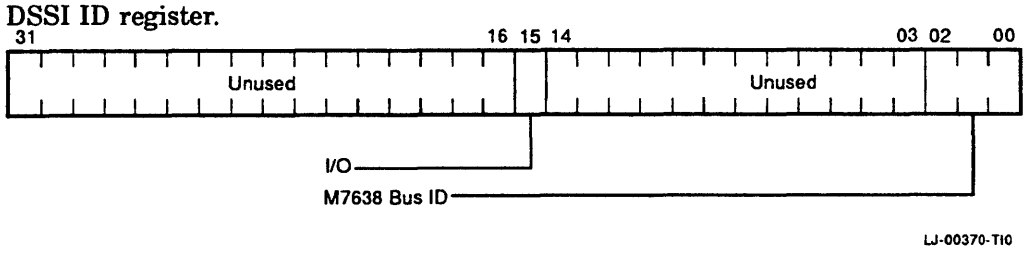


Figure 3–79 DSSI ID Register

Table 3–83 lists the bit descriptions.

Table 3–83 DSSI ID Register Bit Descriptions

Data Bit	Name	Description
<31:16>	UNUSED	Reads return undefined results. Writes have no effect.
<15>	I/O	INPUT/OUTPUT. Read/Write. When this bit is set, the M7638-Ax's ID is determined by DSSI_ID<2:0>.When this bit is clear, the M7638-Ax's ID is determined by on board jumpers and DSSI_ID<2:0> reflects the one's complement of the M7638-Ax's DSSI ID. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<14:3>	UNUSED	Reads return undefined results. Writes have no effect.
<2:0>	M7638-Ax BUS ID	Read/Write. When DSSI_ID<31> is set, this field contains the DSSI ID of the M7638-Ax. When DSSI_ID <31> is clear, this field contains the one's complement of the M7638-Ax's ID. Indeterminate on power up, the negation of DCOK when **SCR<7> is clear or writes to IPR55 (IORESET)**.

3.23.1.6.1.5 DSSI Timeout Register (DSSI_DSTMO)

The Mass Storage Interface DSSI Timeout Register (DSSI_DSTMO), address 1016 001C (hex), contains the timeout values of the SII chip for both the initiator and target roles. Also contained in this register is a single enable bit that governs both timers (Figure 3–80).

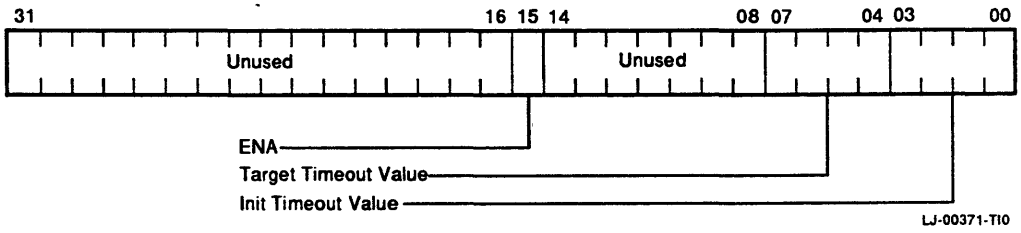


Figure 3–80 DSSI Timeout Register

Table 3–84 lists the bit descriptions.

Table 3–84 DSSI Timeout Register Bit Descriptions

Data Bit	Name	Description
<31:16>	UNUSED	Reads return undefined results. Writes have no effect.
<15>	ENA	ENABLE. Read/Write. When this bit is set, both the DSSI target and DSSI initiator timers are enabled. When clear, both the DSSI target and DSSI initiator timer are disabled. These bits are cleared on power up by the negation of DCOK which SCR<7> is clear, and by writes to the IORESET register (IPR55).
<14:8>	UNUSED	Reads return undefined results. Writes have no effect.

Table 3–84 (Cont.) DSSI Timeout Register Bit Descriptions

Data Bit	Name	Description
<7:4>	TARGET TIMEOUT VALUE	Read/Write. This field contains the number of 200 microsecond intervals which may elapse while the M7638-Ax is the target. The timer starts from the point when the M7638-Ax was selected and ends at the next observed bus free phase. These bits are cleared on power up by the negation of DCOK which SCR<7> is clear, and by writes to the IORESET register (IPR55).
<3:0>	INITIATOR TIMEOUT VALUE	Read/Write. This field contains the number of 200 microsecond intervals which may elapse, from the last observed bus free phase until the next observed bus free phase, while the M7638-Ax is in the initiator role; or the number of 200 microsecond intervals which may elapse before the M7638-Ax, acting as a potential initiator, detects a bus free phase. Should the timer expire under either of these two conditions, the SII chip asserts a DSSI bus reset. These bits are cleared on power up by the negation of DCOK which SCR<7> is clear, and by writes to the IORESET register (IPR55).

3.23.1.6.2 List Pointer Registers

These are the two registers used as address pointers for next incoming and outgoing data buffers.

3.23.1.6.2.1 DSSI Target List Pointer Register (DSSI_TLP)

The Mass Storage Interface Target List Pointer register, address 1016 003C (hex), contains the address to which the SII chip writes the next free receive buffer (Figure 3–81). The SII chip automatically reloads the register with the receive buffer's thread word upon completion of the current transaction. Note this register must contain bits <17:2> of a quadword aligned address; therefore, bit 0 will always be 0. The SII chip interprets an address of 0000 (hex) as the end of a linked list.

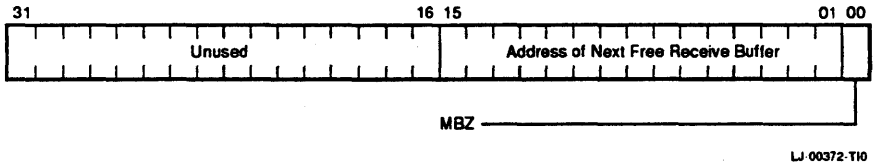


Figure 3–81 DSSI Target List Pointer Register

Table 3–85 lists the bit descriptions.

Table 3–85 DSSI Target List Pointer Register Bit Descriptions

Data Bit	Name	Description
<31:16>	UNUSED	Reads return undefined results. Writes have no effect.

Table 3–85 (Cont.) DSSI Target List Pointer Register Bit Descriptions

Data Bit	Name	Description
<15:1>	ADDRESS OF NEXT INCOMING BUFFER	Read/Write. This field contains bits <17:3> of the quadword aligned address to where the SII chip will find the next free receive buffer. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<0>	MBZ	MUST BE ZERO. Read/Write. This bit is read as 0 and must be written as 0.

NOTE
This register can only be written by the processor when the register is 0 or when DSSI_DSCTRL<15> (DSE) is clear. All other attempts to write to this register have no effect.

3.23.1.6.2.2 DSSI Initiator List Pointer Register (DSSI_IPL)

The Mass Storage Interface Initiator List Pointer register, address 1016 0040 (hex), contains the address from which the SII chip finds the next transmit buffer (Figure 3–82). The SII chip automatically reloads this register with the transmit buffer's thread word upon completion of the current transaction. Note, this register must contain bits <17:2> of a quadword aligned address; therefore, bit 0 must always be 0. The SII chip interprets an address of 0000 (hex) as the end of a linked list.

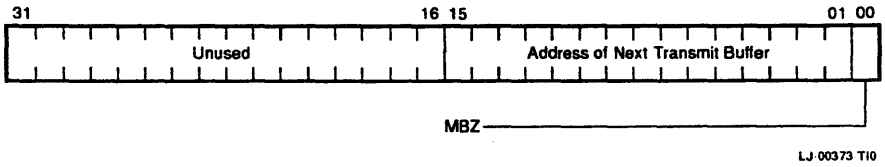


Figure 3–82 DSSI Initiator List Pointer Register

Table 3–86 lists the bit descriptions.

Table 3–86 DSSI Initiator List Pointer Register Bit Descriptions

Data Bit	Name	Description
<31:16>	UNUSED	Reads return undefined results. Writes have no effect.
<15:1>	ADDRESS OF NEXT OUTGOING BUFFER	Read/Write. This field contains bits <17:3> of the quadword aligned address of where the SII chip finds the next transmit buffer. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<0>	MBZ	MUST BE ZERO. Read/Write. This bit reads as 0 and must be written as 0.

NOTE
This register can only be written to by the processor when the register is 0 or when DSSI_DSCTRL<15> (DSE) is clear. All other attempts to write to this register have no effect.

3.23.1.6.3 Diagnostic and Test Registers

This group of registers is used for test and diagnostic purposes only. They should never be used during normal operation.

3.23.1.6.3.1 DSSI Diagnostic Control Register (DSSI_DICTRL)

The Mass Storage Interface Diagnostic Control Register, at address 1016 0054 (hex), allows the SII chip to be placed in one of three diagnostic test modes (Figure 3–83).

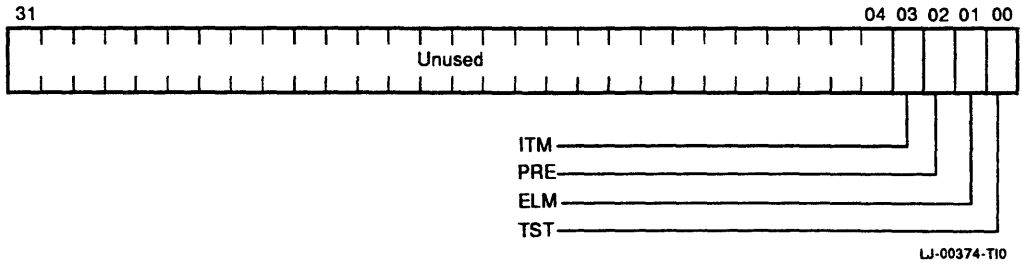


Figure 3–83 DSSI Diagnostic Control Register

Table 3–87 lists the bit descriptions.

Table 3–87 DSSI Diagnostic Control Register Bit Descriptions

Data Bit	Name	Description
<31:4>	UNUSED	Reads return undefined results. Writes have no effect.
<3>	ITM	INTERNAL TEST MODE. Read/Write. When this bit is set, the values written to DSSI_DR0, DSSI_DR1 and DSSI_DR2 are to be looped back into the chip. This enables the processor to insert test vectors into the chip during power up diagnostics. Note that the DSSI_DICTRL <1> (ELM) must be deasserted for this test to be meaningful. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<2>	PRE	PORT ENABLE. Read/Write. When this bit is set, the off-chip drivers to the DSSI port are enabled. After a reset, the M7638-Ax will be disconnected from the bus (this bit will be zero). The primary purpose of this bit is to allow SII chip diagnostics to run without affecting the rest of the DSSI bus (PRE=0). These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).
<1>	ELM	EXTERNAL LOOP-BACK MODE. Read/Write. When this bit is set, the SII chip is in external loop-back mode. In this mode, DSSI_DR0, DSSI_DR1, and DSSI_DR2 are used to directly control the DSSI data and control lines, as well as the external bus transceiver. Note, an external loop-back connector must be in place when using this test mode. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).

Table 3–87 (Cont.) DSSI Diagnostic Control Register Bit Descriptions

Data Bit	Name	Description
<0>	TST	TEST MODE. Read/Write. When this bit is set to 1, the SII chip is in test mode. This enables the user to replace the 20 MHz clock. The new clock is pulsed each time the DSSI_CLOCK register is written. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to the IORESET register (IPR55).

3.23.1.6.3.2 DSSI Diagnostic Register 0 (DSSI_DR0)

The Mass Storage Interface Diagnostic Register 0 (DSSI_DR0), address 1016 0000 (hex) is used during internal and external loop back diagnostic tests. The fields in this register are used to emulate the data lines of the DSSI (Figure 3–84).

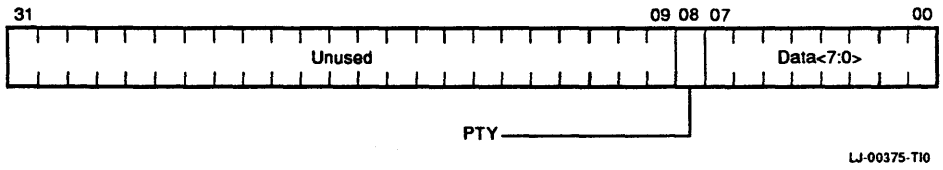


Figure 3–84 DSSI Diagnostic Register 0

Table 3–88 lists the bit descriptions.

Table 3–88 DSSI Diagnostic Register 0 Bit Descriptions

Data Bit	Name	Description
<31:9>	UNUSED	Reads return undefined results. Writes have no effect.
<8>	PTY	Parity. Read/Write. This bit contains the parity bit for the data byte DSSI_DR0<7:0>. Indeterminate on power up, the negation of DCOK when SCR<7> is clear or writes to IPR55 (IORESET). Note, parity checking is only enabled if DSSI_CSR<1> PCE is set to 1. The SII chip uses odd parity checking.
<7:0>	DATA	Read/Write. This field contains the current byte on the data bus. Indeterminate on power up, the negation of DCOK when SCR<7> is clear or writes to IPR55 (IORESET).

NOTE
This register should not be used during normal operation.

3.23.1.6.3.3 DSSI Diagnostic Register 1 (DSSI_DR1)

The Mass Storage Interface Diagnostic Register 1, address 1016 0004 (hex), is used during internal and external loopback tests. In external loopback mode, an external loopback connector in place allows values written into DSSI_DR0 to be read back in DSSI_DR1 and values written into DSSI_DR1 to be read back in DSSI_DR0. In internal loopback mode, it acts as the DSSI bus emulating some of the DSSI control lines. Note that all the control lines are asserted high in internal loopback test mode. Figure 3–85 shows the DSSI diagnostic register 1.

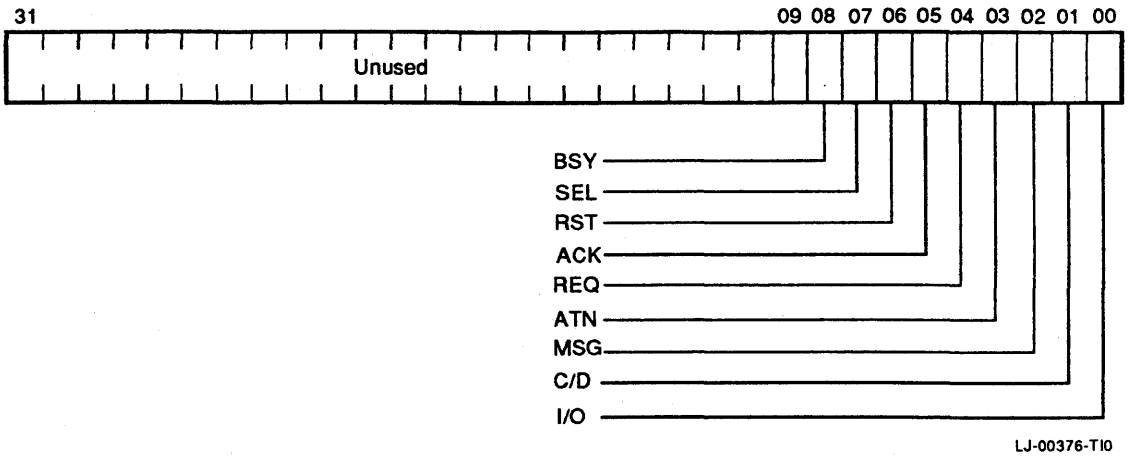


Figure 3–85 DSSI Diagnostic Register 1

Table 3–89 lists the bit descriptions.

Table 3–89 DSSI Diagnostic Register 1 Bit Descriptions

Data Bit	Name	Description
<31:9>	UNUSED	Reads return undefined results. Writes have no effect.
<8>	BSY	BUSY. Read/Write. In internal loopback test mode, DSSI_DICTRL<3> set, this bit emulates the DSSI BSY bus signal. In external loopback mode, this bit is linked to DSSI_DR0<8> (PTY) for driver testing. Indeterminate on power up, the negation of DCOK when SCR<7> is clear or writes to IPR55(IORESET).
<7>	SEL	SELECT. Read/Write. In internal loopback test mode, DSSI_DICTRL <3> set, this bit emulates the DSSI SEL bus signal. In external loopback mode, this bit is linked to DSSI_DR0 <7> (DATA<7>) for driver testing. Indeterminate on power up, the negation of DCOK when SCR <7> is clear or writes to IPR55(IORESET).
<6>	RST	RESET. Read/Write. In internal loopback test mode, DSSI_DICTRL <3> set, this bit emulates the DSSI RST bus signal. In external loopback mode, this bit is linked to DSSI_DR0 <6> (DATA<6>) for driver testing. Indeterminate on power up, the negation of DCOK when SCR <7> is clear or writes to IPR55(IORESET).
<5>	ACK	ACKNOWLEDGE. Read/Write. In internal loopback test mode, DSSI_DICTRL <3> set, this bit emulates the DSSI ACK bus signal. In external loopback mode, this bit is linked to DSSI_DR0 <5> (DATA<5>) for driver testing. Indeterminate on power up, the negation of DCOK when SCR <7> is clear or writes to IPR55(IORESET).

Table 3–89 (Cont.) DSSI Diagnostic Register 1 Bit Descriptions

Data Bit	Name	Description
<4>	REQ	REQUEST. Read/Write. In internal loopback test mode, DSSI_DICTRL <3> set, this bit emulates the DSSI REQ bus signal. In external loopback mode, this bit is linked to DSSI_DR0 <4> (DATA<4>) for driver testing. Indeterminate on power up, the negation of DCOK when SCR <7> is clear or writes to IPR55(IORESET).
<3>	ATN	ATTENTION. Read/Write. In internal loopback test mode, DSSI_DICTRL <3> set, this bit emulates the DSSI ATN bus signal. In external loopback mode, this bit is linked to DSSI_DR0 <3> (DATA<3>) for driver testing. Indeterminate on power up, the negation of DCOK when SCR <7> is clear or writes to IPR55(IORESET).
<2>	MSG	MESSAGE. Read/Write. In internal loopback test mode, DSSI_DICTRL <3> set, this bit emulates the DSSI MSG bus signal. In external loopback mode, this bit is linked to DSSI_DR0<2> (DATA <2>) for driver testing. Indeterminate on power up, the negation of DCOK when SCR<7> is clear or writes to IPR55(IORESET).
<1>	C/D	CONTROL/DATA. Read/Write. In internal loopback test mode, DSSI_DICTRL <3> set, this bit emulates the DSSI C/D bus signal. In external loopback mode, this bit is linked to DSSI_DR0 <1> (DATA<1>) for driver testing. Indeterminate on power up, the negation of DCOK when SCR <7> is clear or writes to IPR55(IORESET).
<0>	I/O	INPUT/OUTPUT. Read/Write. In internal loopback test mode, DSSI_DICTRL <3> set, this bit emulates the DSSI I/O bus signal. In external loopback mode, this bit is linked to DSSI_DR0 <0> (DATA<0>) for driver testing. Indeterminate on power up, the negation of DCOK when SCR <7> is clear or writes to IPR55(IORESET).

NOTE

The data written to this register in internal test mode may differ from that read back from it, since only certain bits are driven when configured as a target or initiator. See Section 3.22.1.6.3.4 for more information on the internal test mode.

3.23.1.6.3.4 DSSI Diagnostic Register 2 (DSSI_DR2)

The Mass Storage Interface Diagnostic Register 2, address 1016 0008 (hex) is used by diagnostics to directly control the DC563 transceiver chip. Figure 3–86 shows the DSSI diagnostic register 2.

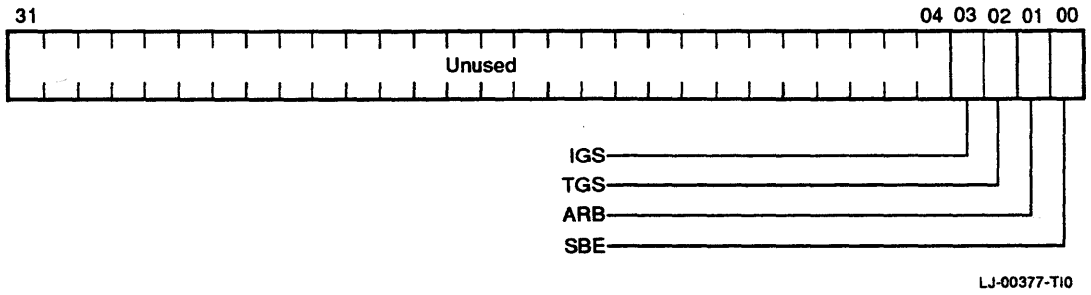


Figure 3–86 DSSI Diagnostic Register 2

Table 3–90 lists the bit descriptions.

Table 3–90 DSSI Diagnostic Register 2 Bit Descriptions

Data Bit	Name	Description
<31:4>	UNUSED	Reads return undefined results. Writes have no effect.
<3>	IGS	Read/Write. This bit enables the DSSI bus drivers for ACK and ATN, placing the SII chip in the initiator role. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to IORESET register (IPR55).
<2>	TGS	Read/Write. When set, this bit enables the DSSI bus drivers for I/O, C/D, MSG, and ATN, placing the SII chip in the target role. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to IORESET register (IPR55).
<1>	ARB	ARBITRATE. Read/Write. This bit enables the decoding of ID0..ID2, putting the SII chip in the arbitration phase. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to IORESET register (IPR55).
<0>	SBE	Read/Write. When this bit is set, the DC563 transceiver drives the DSSI data bus and parity lines. These bits are cleared on power up by the negation of DCOK when SCR<7> is clear, and by writes to IORESET register (IPR55).

NOTE
Special care should be taken when writing to this register to avoid disturbing the DSSI bus during power up diagnostics. This register should only be used when an external loopback connector is in place and not during normal operation.

3.23.1.6.3.5 DSSI Clock Control Register (DSSI_CLOCK)

Writing to the Mass Storage Interface Clock Control Register, address 1016 0058 (hex), generates a pulse which, in test mode (DSSI_DICTRL<0> set to 1), replaces the 20 MHz clock input (Figure 3–87). This can be used to allow the CVAX CPU to observe and sequence the various state machines inside the SII chip.

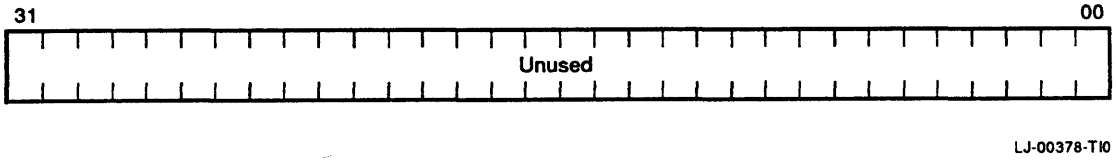


Figure 3–87 DSSI Clock Control Register

Table 3–91 lists the bit descriptions.

Table 3–91 DSSI Clock Control Register Bit Descriptions

Data Bit	Name	Description
<31:0>	UNUSED	Write Only. Writing to this register generates a pulse which, in test mode (DSSI_DICTRL<0> set to 1), replaces the 20 MHz clock input.

3.23.1.6.3.6 DSSI Internal State Registers (0-3)

These registers, at addresses 1016 005C, 1016 0060, 1016 0064, and 1016 0068 (hex), reflect the status of the SII chip's internal state machine when used in test mode (for example, DSSI_DICTRL<0> set to 1).

3.23.2 SCSI Bus Interface

The KN220-AA I/O module contains a SCSI bus interface that is implemented through the NCR 53C94 Advanced SCSI controller, 32Kx32 DMA buffer and RIO interface/ DMA controller.

The 53C94 reduces protocol overhead by performing common SCSI algorithms or sequences in response to a single command. NCR specs the 53C94 to operate at sustained data transfer rates of up to 5 Mbytes per second in synchronous mode and 5 Mbytes per second in asynchronous mode. The 53C94 is an 84-pin PLCC CMOS device conforming to the ANSI standard, X3.131-1986, for small computer systems interface. It includes a 16-bit bus used, in this implementation, to support both DMA and register access. It has on-chip 48 mA drivers for single-ended SCSI transmission. Current implementation operates the 53C94 at its highend frequency limit of 25MHz.

A 32Kx32 DMA buffer is implemented using 4 highspeed 32Kx8 static RAMs. The buffer is accessed by the 53C94 using the 16-bit DMA path. The processor uses the 32-bit RIO bus to load the buffer prior to kicking off SCSI write transfers (or unload the buffer after completion of SCSI reads).

The RIO interface/controller is implemented using programmable array logic and several LSI registers/counters and tranceivers. The controller responds to DMA requests by the 53C94 while allowing the processor access to control and status registers. The 53C94 may address the buffer on halfword (16-bit) boundaries only. The processor may read on word boundaries and write on byte (8-bit) boundaries.

3.23.2.1 DMA Address Register - DMAAR

The DMA Address Register , address 1714 0000, is used to define the current halfword (16-bit) address and direction of DMA activity between the 53C94 and the DMA buffer (Figure 3–88). The DMA address register is a write only register.

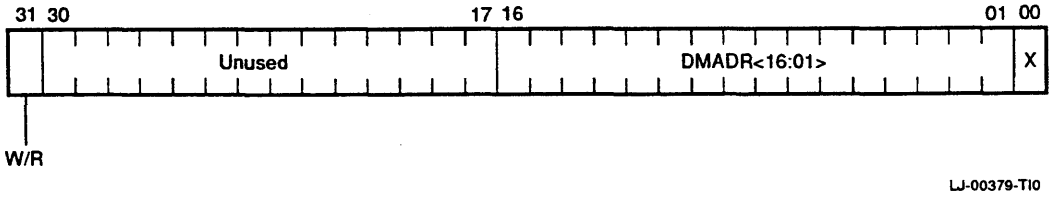


Figure 3–88 DMA Address Register

Table 3–92 lists the bit descriptions.

Table 3–92 DMA Address Register Bit Descriptions

Data Bit	Name	Description
<31>	DMAW	Used to define the direction of DMA activity. If this bit is set, SCSI DMA activity will be out of the DMA buffer.
<30:17>	UNUSED	Writes have no effect.
<16:01>	DMADR	Used to define the halfword address at which DMA activity is to begin. DMA transfers, between the 53C94 and the DMA buffer RAM, start at this address and increment upward on subsequent transfers.
<00>	UNUSED	Writes have no effect.

3.23.2.2 DMA Buffer RAM - DMABR

The dual port DMA Buffer RAM (128 Kbytes), address 17180000-1719FFFF, is used to buffer data being transferred across the SCSI bus. Data to be written is preloaded into the buffer, by the processor, using the 32-bit wide RIO bus. The DMA address register is loaded with the halfword start address of the current data and the DMAW bit is set. The 53C94 is then issued the proper instructions to initiate a write to a SCSI device .When SCSI protocols have been met, the 53C94 begins to DMA data out of the buffer using the 16-bit wide SDATA bus. The SCSI bus write then proceeds independent of the processor. Completion is indicated by a processor interrupt from the 53C94.

SCSI bus reads also use the DMA buffer to store incoming data. The read is set up by loading the DMA address register with the halfword address at which data is to be stored and the DMAW bit is cleared. The 53C94 is then issued the proper instructions to initiate a read from a SCSI device. When SCSI protocols have been met, the 53C94 begins to DMA data into the buffer using the 16-bit wide SDATA bus. The SCSI bus read then proceeds independent of the processor. Completion is indicated by a processor interrupt from the 53C94. The processor then accesses the data across the RIO bus.

3.23.2.3 53C94 Registers - 53C94Rx

The 53C94 Registers, address 17100000 - 1710003c, are 16-bit registers used to configure the controller for SCSI bus transfers, and indicate status and load/ unload the internal FIFO (Figure 3–89). The registers are addressed on word (32-bit) boundaries but data to and from these registers will only be transferred on the lower 16 bits of the RIO bus.

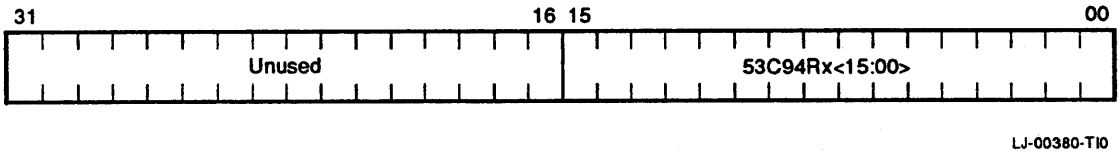


Figure 3–89 53C94 Register

4
Firmware Specification

4.1 Environment

The following is a description of the intended environment in which KN220-AA firmware will be used. This environment includes not only the surrounding hardware but also the field of use.

4.1.1 Users

Engineering, Manufacturing, Customer Services and customers will use this program to test, configure, and boot their KN220-AA module. This firmware will be used to provide an easy means to bootstrap a KN220-AA based system and to detect and isolate system malfunctions.

4.1.2 Hardware

The firmware described in this document resides on the KN220-AA CPU and I/O modules. The KN220-AA is an R3000 and CVAX-based Q22-bus CPU module. Additionally, the KN220-AA is designed to be directly interfaced to other system components; in particular, the Ethernet adapter for network communications, a DSSI port, and a SCSI port for connection of mass storage devices. Naturally, the KN220-AA processor provides a local serial I/O port for support of a standard console.

The KN220-AA support communications with other Q22-bus modules through its Q22-bus interface consisting of a scatter/gather map, a direct map of the Q22-bus I/O page, and memory through I/O space.

The KN220-AA provides a maximum of 384 Kbytes of EPROM for the firmware. Firmware resides on the KN220-AA processor and I/O modules in three 128 Kbytes EPROMs which are arranged as words (32 bits) and located at the R3000 restart location at physical address 1FC00000 and the CVAX restart location at physical address 20040000. The R3000 uses 256 Kbytes and the CVAX uses 128 Kbytes.

The firmware uses the KN220-AA module LEDs and a console terminal to communicate diagnostic progress, display error conditions, and indicate the current mode of operation. Additionally, the console terminal is used as the primary operator interface when in the console I/O mode.

NOTE
A console terminal is not required for operation since the module can be configured to bootstrap automatically. However, in most scenarios, a console terminal is a recommended part of a standard configuration.

4.1.3 Software

The KN220-AA firmware runs standalone and does not require other software products for normal operation in console I/O mode. However, in most situations, an operating system (or other image) is loaded in KN220-AA local memory and control is transferred to it. When the operating system is running, the processor is in program I/O mode. (The terms console I/O mode and program I/O mode refer to the context of the console terminal.)

The KN220-AA supports ULTRIX-32. Additionally, the firmware supports bootstrap of MDM and other diagnostics images. Furthermore, the console supports communication with an APT host in manufacturing environments through the console serial line.

4.1.4 Services

KN220-AA firmware is an integral part of the module and does not require installation or support services.

4.2 Firmware Capabilities

The KN220-AA firmware provides the following services:

- Diagnostics that test all components on the CPU and I/O boards and verify correct operation.
- Automatic/manual bootstrap of an operating system following processor halts.
- Automatic/manual restart of an operating system following processor halts.
- An interactive command language that allows the user to examine and alter the state of the processor.
- The capability to operate in both a secure and insecure fashion to control access to the system.

The remainder of this Chapter describes in detail the functions and external characteristics of the KN220-AA firmware.

4.3 Power up

On power up, the KN220-AA firmware performs initial power up tests, locates and identifies a console device, and performs the remaining diagnostics. Certain power up actions are dependent on the state of the processor select jumpers and the operation and function switches on the H3602-AC panel. The processor select jumpers select the processor (R3000 or CVAX) that will start running at power up. The operation and function switches determine various functions described in sections 4.3.4 and 4.3.6.

4.3.1 Processor Select

The processor select jumpers specify which processor starts executing at power up. The operation switch controls secure operation if security is enabled. See section on security for a description of the security features of the firmware.

When the CVAX is selected, the operation switch has three settings: normal/secure, maintenance, and action. When in normal/secure mode, the function switch positions are break enabled and break disabled. If the CVAX is in maintenance mode, the function switch positions are break enabled and break disabled. If the position of the operation switch is set to action, the function switch positions are test and query.

When the R3000 is selected, the OPERATION switch has one valid setting: normal mode. When the R3000 is in normal/secure mode, the function switch positions are break enabled and break disabled. The maintenance and action modes are disabled.

The CVAX switches are described in Table 4–1.

Table 4–1 CVAX Operation and Function Switches

Mode	Operation	Mode	Function
Normal /secure	→	Break enabled	⊙
		Break disabled	Ȯ
Maintenance	⊕	Break enabled	⊙
		Break disabled	Ȯ
Action	{°	Test	⊙
		Query	Ȯ

The R3000 switches are described in Table 4–2.

Table 4–2 R3000 Operation and Function Switches

Mode	Operation	Mode	Function
Normal /secure	→	Break enabled	⊙
		Break disabled	Ȯ
Maintenance	⊕	Disabled	
Action	{°	Disabled	

NOTE
The H3602-AA is shipped with the processor select jumpers set for the CVAX to be the default power-up processor.

4.3.2 CVAX Initial Power-Up Test

When the CVAX is selected to start executing at power up, it performs the Initial Power-up Test (IPT). The purpose of the IPT is to verify that the console private NVRAM (in the SSC) is valid; and if invalid, the purpose of the IPT is to test and initialize it.

If the NVRAM in the SSC contains invalid data, the IPT initializes certain non-volatile data to a known state. The IPT then initializes other data structures and performs a processor initialization. If the operation switch is set to action and the function switch is set to test, the IPT then tests the console serial line as described in Section 4.3.4.3.1.

NOTE
All IPT failures are considered fatal, and the KN220-AA will hang with a value on the LEDs indicating the point of failure. Refer to Table 4–3 for the meaning of the LEDs.

4.3.3 Locating a Console Device

After the CVAX IPT has completed successfully, the firmware attempts to locate a console device and find out what type of device it is. Normally, this is the device attached to the console serial line. In this case, the firmware will send out a device attributes escape sequence to the console serial line to determine the type of terminal attached and the functions it supports. Terminals that do not respond to the device attributes request correctly, are assumed to be hardcopy devices.

NOTE
The KN220-AA is always an arbiter CPU and the console serial line is unconditionally treated as the system console.

Once a console device has been found the firmware displays the KN220-AA banner message as follows:

```
KN220-A Vn.n
```

The banner message contains the processor name and the version of the firmware. The letter code in the firmware version indicates whether the firmware is pre-field test (X), field test (T), or an official release (V). The first digit indicates the major release number and the trailing digit indicates the minor release number.

4.3.4 CVAX Operation and Function Switches

The operation and function switches tell the CVAX to perform maintenance functions or to continue with the bootstrap process.

4.3.4.1 Operation Switch Set to Normal

When the OPERATION switch is set to normal/secure, the CVAX executes the power up diagnostics. In addition to the message text, a countdown is displayed to indicate diagnostic test progress. A successful diagnostic countdown is as follows:

```
KN220-A Vn.n
Performing normal system tests.
83..82..81..80..79..78..77..76..75..74..73..72..71..70..69..68..67..
66..65..64..63..62..61..60..59..58..57..56..55..54..53..52..51..50..
49..48..47..46..45..44..43..42..41..40..39..38..37..36..35..34..33..
32..31..30..29..28..27..26..25..24..23..22..21..20..19..18..17..16..
15..14..13..12..11..10..09..08..07..06..05..04..03..
Tests completed.
Memory Size: 33554432 (0x2000000) bytes
Ethernet Address: 08-00-2b-16-eb-65
>>
```

After the diagnostics are executed and completed successfully, control is passed to the R3000 processor by writing to bits 30 and 31 of the Select Processor Register (SPR) in the KN220-AA processor (Section 4.3.7.1). Then, the R3000 processor continues with the power-up sequence as described in Section 4.3.6.1.

If there are diagnostic failures, a diagnostic register dump is performed similarly to the following example. The remaining diagnostics execute the countdown continues and the KN220-AA enters maintenance mode if security is not enabled, (Section 4.3.4.2). For a detailed description of the register dump, refer to Section 4.6.

```
KN220-A Vn.n
Performing normal system tests.
83..82..81..80..79..78..77..76..75..74..73..72..71..70..69..68..67..
66..65..64..63..62..61..60..59..58..57..56..55..54..53..52..51..50..
49..48..47..46..45..44..43..42..41..40..39..38..37..36..35..34..33..
32..31..30..29..28..27..26..25..24..23..22..21..20..19..18..17..16..
15..14..13..12..

?66 2 12 FF 01FC 000A

P1 =00000100 P2 =00000000 P3 =B7100010 P4 =FFFFFF87 P5 =00000000
P6 =00000000 P7 =00000000 P8 =00000000 P9 =00000000 P10=00000000
P11=00000000 P12=00000000 P13=00000000 P14=00000000 P15=00000000
P16=00000000 P17=00000000 P18=00000000 P19=00000000 P20=00000000
9p =83620242 sp =B8001B1C fp =00000000 sr =B048FF04
epc=BFC2DD1C badvaddr =00000000 cause =00000000
11..10..09..08..07..06..05..04..03..
Normal operation not possible.

>>
```

4.3.4.2 Operation Switch Set to Maintenance

When the operation switch is set to maintenance, the CVAX executes the power-up diagnostics. In addition to message text, a countdown is displayed to indicate diagnostic test progress. A successful diagnostic countdown is shown in Section 4.3.4.1.

After the diagnostics are executed, the KN220-AA enters maintenance mode and prompts the user for commands. Note that there are two console prompts, normal R3000 (>>) and maintenance CVAX (>>>).

4.3.4.3 Operation Switch Set to Action

Two different operations are possible when the operation switch is set to action: test and query.

4.3.4.3.1 Function Switch Set to Test

If the function switch is set to test, the console serial line external loopback test is executed at the end of the IPT. The purpose of this test is to verify that the console serial line connections from the KN220-AA through the H3602-AC panel are intact.

NOTE

An external loopback connector should be inserted in the serial line connector on the H3602-AC panel before cycling power to invoke this test.

During this test, the firmware toggles between the two states, active and passive, for a few seconds each and each state displays a different number on the LEDs.

During the active state (about three seconds long), the LEDs are set to "7". In this state, the firmware reads the baud rate and operation switch, then transmits and receives a character sequence. If the operation switch has been moved from the action position, the firmware exits the test and continues as if on a normal power up.

During the passive state (about seven seconds long), the LEDs are set to "4".

If at any time the firmware detects an error (parity, framing, overflow, or no characters), the firmware hangs with a "7" on the LEDs.

4.3.4.3.2 Function Switch Set to Query

If the KN220-AA designated console device supports the DEC Multinational Character Set (MCS) and either the NVRAM data is invalid or the FUNCTION switch is set to query the firmware prompts for the console language. The firmware displays the language selection menu shown below.

After the language query, the firmware invokes the ROM-based diagnostics. The firmware enters normal mode if the diagnostics are completed successfully. However, if the diagnostics fail, the firmware enters maintenance mode.

```
KN220-A Vn.n

 1) Dansk
 2) Deutsch (Deutschland/Österreich)
 3) Deutsch (Schweiz)
 4) English (United Kingdom)
 5) English (United States/Canada)
 6) Español
 7) Français (Canada)
 8) Français (France/Belgique)
 9) Français (Suisse)
10) Italiano
11) Nederlands
12) Norsk
13) Português
14) Suomi
15) Svenska
 (1..15):
```

If no response is received within 30 seconds, the language defaults to English (United States/Canada).

NOTE

This action is only taken if the console device supports DEC MCS. Any console device that does not support DEC MCS, such as a VT100, defaults to English (United States/Canada). Language selection is only useful for the CVAX console.

4.3.4.4 LED Codes

In addition to the console diagnostic countdown, a hexadecimal value is displayed on the diagnostic LEDs on the module and the H3602-AC panel. The purpose of the LED display is to improve fault isolation when there is no console terminal or when the hardware is incapable of communicating with the console terminal. Table 4–3 lists all LED codes and the associated actions that are performed at power up. The LED code is changed before the corresponding test or action is performed.

Table 4–3 LED Codes

LED Value	Actions
F	Initial state on power up, no code has executed
E	Entered ROM, some instructions have executed
D	Waiting for power to stabilize (POK)
C	SSC and ROM tests
B	CVAX tests
A	R3000 tests

Table 4–3 (Cont.) LED Codes

LED Value	Actions
9	Memory controller and memory tests
8	CQBIC (Q22-bus) tests
7	Console loopback tests
6	DSSI and SCSI subsystem tests
5	Ethernet subsystem tests
4	CVAX console I/O mode
3	R3000 console I/O mode
2	CVAX primary/secondary bootstrap
1	R3000 primary/secondary bootstrap
0	Operating system running

4.3.5 R3000 Initial Power-Up Test

Upon power up, the R3000 performs the Initial R3000 Power-up Test (IPT). Refer to Section 4.6. It is important to understand that only the R3000 chip set and memory diagnostics are run. The KN220-AA I/O module is not diagnosed.

NOTE
When the processor select jumpers are set for the R3000 to start executing at power up, there is no guarantee that the KN220-AA I/O module is fully operational.

After the R3000 IPT has completed successfully, the R3000 determines if a console device is present and attempts to determine the type of device attached to the console serial line. The firmware sends out a device attributes escape sequence to the console serial line to determine the type of terminal attached and the functions it supports. Terminals that do not respond to the device attributes request correctly are assumed to be hardcopy devices.

NOTE
All IPT failures are considered fatal and the KN220-AA will hang with a value on the LEDs indicating the point of failure. Refer to Table 4–3 for the meaning of the LEDs.

4.3.6 R3000 Operation and Function Switches

The operation and function switches tell the R3000 to automatically bootstrap or to enter the console I/O mode.

4.3.6.1 Operation Switch Set to Normal/Secure

When the operation switch is set to normal/secure and the environment variable bootmode is set to A, the R3000 processor attempts to find a booting device specified by the bootpath environment variable stored in non-volatile memory. The bootmode environment variable is set using the setenv command. Refer to section Section 4.4.2 for information about boot devices.

When the operation switch is set to normal and the environment variable bootmode is not initialized or it is set to D, the R3000 enters the console I/O mode and prompts the user for commands. If the console was made secure, the console prompt is as follows:

s>
If the console was not made secure, the console prompt is as follows:

>>
Refer to Section 4.5.1 for a description of the R3000 console security features. Refer to Section 4.5 for the console command descriptions.

4.3.6.2 Operation Switch Set to Maintenance

When the operation switch is set to the maintenance mode, the R3000 behavior is the same as when in the normal mode.

4.3.6.3 Operation Switch Set to Action

When the operation switch is set to the action mode, the R3000 behavior is the same as when in the normal mode.

4.3.7 Interprocessor Interaction

The R3000 and the CVAX processors can each select the other processor to start or continue execution. This is accomplished by writing to the Select Processor Register (SPR) as described in Section 4.3.7.1.

4.3.7.1 Select Processor Register (SPR)

The SPR determines which processor is selected. This is a write-only register.

If CVAX writes a 0x40000000 to the SPR (CVAX executes the command EXIT), then:

1. The CVAX hangs on a DMA grant.
2. The R3000 begins execution.

If R3000 writes a 0x80000000 to the SPR (R3000 executes the command maint), then:

1. The R3000 hangs on a RDBUSY or WRBUSY stall.
2. The CVAX begins execution.

4.3.8 Power-Up Sequence

The KN220-AA power-up sequence depends on whether the processor is in normal mode or in the maintenance mode.

NOTE
If the R3000 is selected to be the power-up processor, the KN220-AA power-up sequence does not initialize the KN220-AA I/O board components, leaving the KN220-AA processor in an unknown state.

4.3.8.1 Normal Power-Up Operation

When the KN220-AA is set to operate in normal mode, the power-up sequence is as follows:

1. CVAX powers up (begins execution at a location pointed to by 2004 0000).
2. CVAX runs self-test diagnostics.
3. CVAX executes the EXIT command (SPR ← 0x40000000).
4. CVAX hangs on a DMA grant.
5. R3000 begins execution at the address 1FC0 0000.

If the bootmode environment variable is set to "a", the R3000 attempts to auto-boot. If the bootpath environment variable is valid, the auto-boot succeeds, otherwise, the R3000 waits for a command at its prompt (>>).

If the bootmode environment variable is not initialized ("*") or if it is set to "d", the R3000 waits for a command at its prompt (>>) when the R3000 waits for a command at its prompt (>>) console commands can be received.

If the command is maint (SPR ← 0x80000000), then the following occurs:

- The R3000 hangs on a RDBUSY stall.
- The CVAX resumes execution.

4.3.8.2 Maintenance Power-Up Operation

When the KN220-AA is set to operate in the maintenance mode, the power-up sequence is as follows:

1. The CVAX is powered up (begins execution at a location pointed to by 2004 0000).
2. CVAX runs self-test diagnostics.
3. CVAX enters the console mode (CVAX waits for a command at its prompt (>>>).
   - console commands can be received
   - if the command is EXIT (SPR ← 0x40000000), then the following occurs:
     - The CVAX hangs on a DMA grant.
     - The R3000 begins execution.

4.3.9 Processor Identification

The KN220-AA provides the Sys_Type read only register at physical location 20040004 in the CVAX firmware EPROM and through the R3000 environment variable Sys_Type.

4.3.9.1 R3000 Sys_Type Environment Variable

The Sys_Type environment variable format is described in Section 4.5.4.3.

4.3.9.2 CVAX Sys_Type Register Layout

The layout of the CVAX Sys_Type register is shown in Figure 4–1. The CVAX Sys_Type register address is hex 20040004.

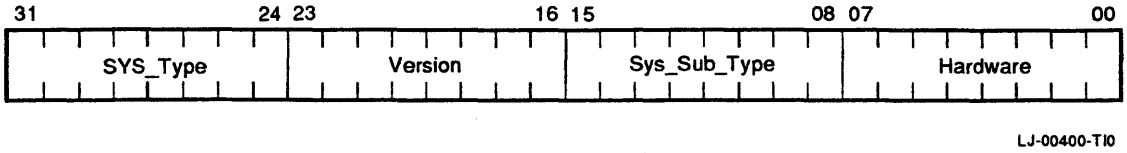


Figure 4–1 System Type Register

Field	Name	RW	Description
31:24	SYS_TYPE	ro	This field identifies the type of system for a specific processor. 11 : DECsystem 5500.
23:16	VERSION	ro	This field identifies the resident version of the firmware EPROM encoded as two hexadecimal digits. For example, if the banner displays V5.0, then this field is 50 (hex).
15:8	SYS_SUB_ TYPE	ro	This field identifies the particular system sub-type. *01 : KN220-AA*
7:0	HARDWARE	ro	This field identifies the hardware revision level.

4.4 Operating System Bootstrap

Bootstrapping is the process of loading and transferring control to an operating system. The KN220-AA supports bootstrap of ULTRIX-32. Additionally, the KN220-AA will boot MDM diagnostics and any user application image which conforms to the boot formats described herein.

On the KN220-AA a bootstrap occurs whenever a **boot** command is issued at the console in maintenance or normal mode or whenever the processor halts and the conditions specified in Section 4.4.7 for automatic bootstrap are satisfied.

The rest of this chapter assumes that the CVAX processor always starts executing the Firmware at power up.

4.4.1 MDM Bootstrap

MDM is booted by the CVAX processor when in maintenance mode and the **boot** command is typed at the console prompt.

4.4.2 Operating System Bootstrap

The operating system is booted by the R3000 when it is in one of the following states:

- The OPERATION switch is in the normal position and the environment variable bootmode is set to a
- The OPERATION switch is in the normal position and the environment variable bootmode is undefined or set to **d** and the **boot** command is typed
- The Operating System initiates a reboot operation

One of four ports may be used for bootstrapping:

- KN220-AA I/O module Ethernet controller
- KN220-AA I/O moduleDSSI controller
- KN220-AA I/O module SCSI controller
- KN220-AA Q22-bus MSCP or TMSCP controller

Refer to Section 4.4.6 and to Section E.1 for a list of the supported devices.

The disk and tape bootstrap code has no knowledge of any file structure on disks or tapes. Because of this, the operating system must provide an image capable of being loaded via the boot block that in turn loads the remainder of the operating system.

The Ethernet bootstrap code is capable of booting using the MOP Ethernet bootstrap protocol. Support for BOOTP is not planned.

The boot code allows the operator to set the default bootstrap device and file name by storing them in a non-volatile environment variable (see Section 4.5.4.3). Once set, the system normally uses this default for bootstraps. However, the operator can override the default manually.

The boot code interface also provides for command extension via the bootstrap mechanism. If, after parsing a command line, the boot code interface fails to recognize a keyword that begins a command, it looks for the environment variable path and if it finds the path, it inserts the value of path prior to the keyword and attempts to boot the resulting filename. Any remaining arguments following the initial keywords are passed as arguments to the program after it is loaded.

4.4.3 Boot Process

The boot process is as follows:

1. Determine the device. This is done by either parsing parsing the **boot** command or from the bootpath environment variable. The bootpath must be set prior to executing the **boot** command (using the **setenv** command) if the environment variable is to be used to specify the boot device.
2. Open the boot device and read block zero (see Figure 4–2).
3. Read the sequence of blocks described in the bootblock into main memory.
4. Transfer control to the loaded code at the start address found on the bootblock. Arguments supplied through the **boot** command are passed using the C-language (argc,argv) conventions. A pointer to the current environment table is also passed.

Any exception in this process forces the boot code to display the console prompt.

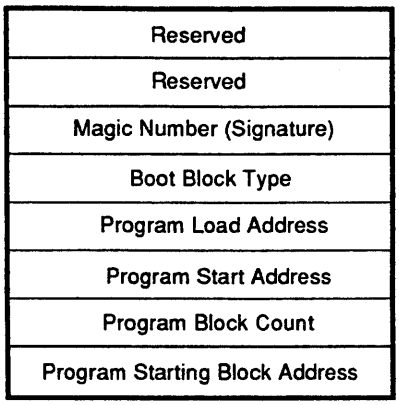


Figure 4–2 Bootblock Layout

4.4.4 Bootstrap Support Routines in the Console

The ULTRIX-32 bootstrap loader contains no code for performing I/O or machine dependent operations. Instead, the loader calls a set of routines provided by the console. The address of these routines are contained in a transfer vector located in console ROM. All routines are called as normal C routines using the R3000 calling conventions.

The entry point routines can be classified into the following groups:

- console - Invoke console program restarts, reboots, etc.,
- saio - I/O support for standalone programs. Raw device I/O
- machine - Machine specific functions (cache flushes, interlock memory references, etc.,)
- libc - Subset of the C library (strings, getenv, etc.,)
- parser - Access to console command parser
- commands - Access to subset of console commands

Section D.1 describes the console entry points.

4.4.5 Console Use of Memory Space

The KN220-AA console reserves the bottom 128 Kbytes of physical address space visible from kseg1. The 128 Kbytes are partitioned as follows:

Table 4–4 Console Memory Space

Address	O/S Interface	Console Program
Starting-Ending	0x00000000-0x000004FF	0x00000500-0x0001FFFF
Starting-Ending		and 0x00020000-0x0002FFFF

The Operating System interface region is used to hold the boot block and the current exception handling code.

The Console Program region is used to hold the console program stack and static data structures.

4.4.6 Boot Devices

Three types of boot media are supported, disk, tape and Ethernet. When specifying a boot device and a boot file the following syntax is used:

dev(controller, unit, lbn)filename

The **dev** field is a device mnemonic. The controller and unit fields specify the hardware controller and unit number of the device. The logical block number (lbn) field specifies absolute block number from the beginning of the disk. Logical block numbers are only meaningful for disk devices. If not specified, controller, unit, and lbn default to zero. The parenthesis are always required. The filename field identifies the name of the file to be loaded.

In the case of SCSI devices, the unit consists of two parts, the bus identification number of the boot device (busid), and a logical unit number (lun). The SCSI unit is specified busid.lun. At this time only one lun is supported, and the default specification is to lun 0. Consequently, the SCSI unit will typically be the busid alone.

4.4.6.1 Disk

rf( ), rz( ) and ra( )

DSSI, SCSI and MSCP disks are supported. A simple "bootblock" bootstrap is supported from disk. If the disk is partitioned, only a bootblock per partition is possible.

4.4.6.2 Tape

tz( ) and tm( )

Only SCSI and TMSCP tapes are supported. A simple "bootblock" bootstrap is supported from tape.

4.4.6.3 Ethernet

Bootstrapping over Ethernet is a very important function and the MOP protocol is used for network bootstraps.

mop( )

This is a standard DIGITAL bootstrap protocol. With MOP, either the client or the server may specify the file to be downloaded. If the client supplies a file specification, then that file is provided by the server. If the client does not supply a file specification, the server determines the file from a database of registered clients. In the latter case, the client must be known to the server and be in its database. The former case has one restriction: The file has to be known to the server.

4.4.7 Halts

The R3000 processor does not provide a halt input to unconditionally call the console program. The KN220-AA has a hardware Halt interrupt at hardware IRQ 5. The Operating System interrupt handler must call the console at the PROM_HALT entry point. Interrupts must be left disabled. The console program will save the machine state and restore in response to the **continue** command.

Table 4–5 describes the R3000 registers that are saved when the PROM_HALT entry point is called. The **continue** command first restores the R3000 registers to the saved values and then restarts the Operating System at the saved return address in register **ra**.

Table 4–5 PROM_HALT Saved Registers

Register	Name	Register	Name
$2..$3	v0..v1	$4..$7	a0..a3
$16..$23	s0..s7	$26..$27	k0-k1
$28	gp	$29	sp
$30	fp	$31	ra
	c0_sr		cause
	c0_sr		cause
	badaddr		context
	tlbhi		tlblow
	index		random
	mdhi		mdlow
$f0..$f31	fgr1..fgr31	fcr31	c1_sr

The use of the **maint** command while the R3000 is in console I/O mode destroys the saved machine state. If the Operating System is halted and maintenance mode is invoked, a subsequent attempt to restart the Operating System will cause unpredictable results.

NOTE
Entering maintenance mode after an R3000 Operating System halt destroys the saved machine state.

4.5 KN220-AA Console Language and Security Features

4.5.1 Console Security Features

The system can either be secure or unsecure. Correspondingly, the operator is either privileged or unprivileged. An encrypted password is stored in NVR (non-volatile ram). If the encrypted value is zero, the system is unsecure. If the encrypted value is non-zero, the system is secure. The operator becomes privileged by entering the proper password.

Table 4–6 Security Operation

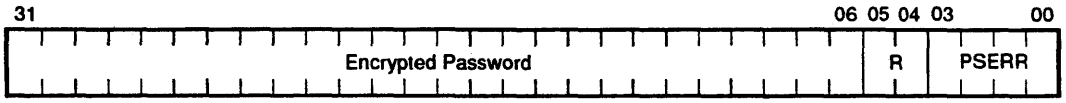
	Secure	Unsecure
Priv	All console commands allowed.	All console commands allowed

Table 4–6 (Cont.) Security Operation

	Secure	Unsecure
UnPriv	Only boot command with no arguments and passwd command with no arguments allowed.	All console commands allowed

If the system is secure, on entry to the console the operator always begins in unprivileged mode. The password command described in Section 4.5.4.4.12 is used to change between privileged and unprivileged operation. In secure/unprivileged mode, only the boot command with no arguments and the "passwd" command with no arguments are allowed.

A 32-bit field in non-volatile RAM (NVR) is reserved for security use. 26 bits are for an "encrypted" password. Four bits are used as a counter for logging failed "password" attempts. Two bits are unused. The non-volatile RAM is shown in Figure 4–3.



PSERR: 4 bit count of password failures
R:2bit field reserved for future use
Encrypted Password: 26 bit encryption of user password

LJ-00401-TI0

Figure 4–3 Non-Volatile RAM Format

If the encrypted input does not exactly match the encrypted password field the PSERR field will be checked for its maximum value (15). If PSERR is not at its maximum value it will be incremented. If PSERR is at its maximum value the console delays 10 seconds. PSERR will remain at its maximum value until a valid password is entered. Once the password is accepted, and if the value in PSERR is not zero, the PSERR value is displayed on the console terminal.

When in the secure/unprivileged mode, the R3000 console prompt is "S>", in all other cases is it ">>"

If the password is forgotten or if it somehow becomes corrupted, there is a mechanism that causes the password in NVR to be ignored or to be zeroed. When the MODE switch is in Maintenance Mode, the operator can enter the "unpriv" command to the CVAX console causing the password field in NVR to be zeroed.

4.5.2 Console Security Commands

The security feature allows a privileged user access to all of the console commands. An unprivileged user is limited to using only the boot command.

4.5.2.1 Password Command

The four variants of this command are used to control the console security feature. The variants are listed as follows:

- **passwd** - This command enables the console user to enter the security password to access the privileged console prompt.

  NOTE
  The use of the passwd command with flags (-s, -c, or -u) is restricted to privileged mode, while the use of the passwd command without flags is restricted to the unprivileged mode.

- **passwd-s** - This command is used to set a new security password. The password can be from 8 to 32 characters long.
- **passwd-c** - This command removes security restrictions by clearing the security password.
- **passwd-u** - This command causes the console user to be unprivileged. The unprivileged prompt is displayed.

For more information about console security feature, refer to the *KN220 CPU System Maintenance Manual* (EK-375AA-SM).

4.5.3 Maintenance Mode Console Command Language

When in maintenance mode, the KN220-AA console command language interface is similar to the KN210 maintenance console language. Three additional commands are implemented: the **EXIT** command, the **SHOW SCSI** command, and the **UNPRIV** command, The **EXIT** command has no arguments and has the effect of switching the KN220-AA from maintenance mode to normal mode.

4.5.4 Normal Mode Console Command Language

When the KN220-AA is in secure mode and in console I/O mode, it will read and interpret commands received on the console terminal. The commands are based on the DECstation 3100 command language.

4.5.4.1 Control Characters

Certain ASCII control characters have special meaning when typed on the console terminal. These characters are described in Table 4–7. The symbols <CR>, <LF>, <SP>, and <BS> are used to represent the ASCII carriage-return, line-feed, space, and backspace characters respectively.

Table 4–7 Control Characters

Character	Function
<CR>	Ends a command line. Command characters are buffered until a carriage-return is received.

Table 4–7 (Cont.) Control Characters

Character	Function
Delete	Deletes the previously typed character. If the console terminal is defined as hardcopy (environment variable **term** set to "hardcopy") the deleted text is echoed surrounded by backslashes. If the console terminal is a CRT (environment variable **term** set to "crt") each delete is echoed with the sequence "<BS><SP><BS>". Deletes received when there are no characters to be deleted are ignored.
Control-C	Causes the console to abort processing of a command.
Control-O	Causes console output to be discarded until the next Control-O is entered, or until the next console prompt or error message is issued. Control-O is also canceled when Control-C is typed
Control-Q	Resumes console output that was suspended by Control-S.
Control-R	Causes the current command line to be displayed, omitting any deleted characters.
Control-S	Suspends output on the console terminal until Control-Q is typed.
Control-U	Discards all characters accumulated for the current line.
Control-V	Supresses any special meaning associated with the next character.

4.5.4.2 Lexical Conventions

The console is case sensitive with respect to parsing commands but case is preserved when assigning values to environment variables.

All console commands are specified using US ASCII characters only. Values entered for environment variables however may contain any 8 bit character code.

Numeric values may be entered as decimal, hexadecimal, octal, and binary values.

- Decimal values are represented by a string of decimal digits with no leading zeros (123).
- Octal values are represented by a string of octal digits and a leading zero must be present (0177).
- Hexadecimal values are represented by hexadecimal digits preceded with a "0x" (0x3ff).
- Binary values are represented by binary digits preceded with a "0b" (0b1001).

4.5.4.3 Environment Variables

The KN220-AA console makes use of environment variables to pass information to the operating system. Some of the environment variables are maintained in non-volatile RAM so that their contents are not lost when power is removed.

Additional environment variables may be defined by the operator or are defined automatically by the console program but they are lost when power is lost. Some of the environment variables below (Table 4–8) are maintained in non-volatile memory (nvram) and others are automatically initialized (init) by the console program.

Table 4–8 Environment Variables

Variable	type	Function
baud	init	Baud rate of the console terminal line. Allowed values are 300, 600, 1200, 2400, 4800, 9600, 19200, 38400
bootpath	nvram	A string containing the complete boot path specification. The boot path has two fields; the boot device and the boot file (see Section 4.4.6). An example of a bootpath definition is: ra(0,0,0)moon_lander
bootmode	nvram	A one character code controlling what action the console is to take on powerup or following a reset. Two codes are defined: "a" for autoboot and "d" to halt after performing powerup diagnostics. Allowable console devices are tty(0) for console serial line
bitmap	init	The hexadecimal address of good pages bitmap
bitmaplen	init	The length of the memory bitmap in bytes
osconsole	init	Always tty(0)
console	nvram	Always selects tty(0)
scsiid	nvram	The SCSI port number, defaults to 7
systype	init	Contains information used to identify the processor. Bits 31:24 contain the CPU type. Bits 23:16 contain the system type (11 for KN220-AA). Bits 15:8 contain the Firmware revision level and bits 7:0 contain the hardware version level

4.5.4.4 Commands

The commands documented below are accepted by the console.

4.5.4.4.1 ?

```
? [command-list]
```

Identical to the help command.

4.5.4.4.2 Boot

```
boot [-f file] [-s] [-n] [args]
```

Boot loads the file following the flag -f. If the -f flag is not specified then the file specified by the environment variable bootpath is loaded. If -s is specified, the operating system boots in single user mode. If -n is specified then the file is loaded but control is not passed to the program. If any arguments are present, then they are passed to the booted image using the standard argc/argv mechanism. If any argument begins with a "-" then it must be preceded with an additional "-" character.

4.5.4.4.3 Continue

```
continue
```

The continue command causes the processor to begin execution at the address currently in the saved program counter. The processor state saved at the last console entry is restored before leaving console mode. Note, a state must be saved (using halt) before continue is executed or the results of the command will be unpredictable.

4.5.4.4.4 D

```
d [-bhw] [address] value
```

D deposits a single byte, halfword or word value at the indicated address. If no address is specified, the default is one location above the last, accessed by e or d commands.

4.5.4.4.5 Dump

```
dump [-Bcdoux] [-bhw] range
```

Dump generates a formatted display of the contents of memory. Memory contents may be displayed (simultaneously) in hex (-x), unsigned decimal (-u), octal (-o), decimal (-d), as ASCII characters (-c), or as binary (-B). Memory contents may be dumped as bytes (-b), halfwords (-h), or words (-w). The range of memory to be dumped may be specified as base-address (a single value is dumped) base-address#count (a specified number of values are dumped) or base-address:limit-address (all values between the base address and the limit address are dumped).

4.5.4.4.6 E

```
e [-bhw] [address]
```

E displays the byte, halfword, or word at address. If no address is specified, the default is one location above the last accessed by e or d commands.

4.5.4.4.7 Fill

```
fill [-bhw] [-vvalue]  range
```

Fill sets the range of memory specified to the value specified. If no value is specified then zero is used. Memory contents may be filled as bytes (-b), halfwords (-h), or words (-w). The range of memory to be filled is specified as in the dump command.

4.5.4.4.8 Go

```
go [entry]
```

Go transfers control to the indicated entry point address. If no entry address is supplied, then the entry point of the last program module loaded is used.

4.5.4.4.9 Help

```
help [command-list]
```

Help displays a brief synopsis of the indicated command. If no command list is supplied then console displays a synopsis of each command.

4.5.4.4.10 Init

```
init
```

Init performs a full initialization. The effect is identical to that performed at power-up or reset except that no diagnostics are executed.

4.5.4.4.11 Maint

```
maint
```

Maint causes the console to enter **maintenance** mode. Any saved program state is discarded. See chapter 18 for an implementation description.

4.5.4.4.12 Passwd

```
passwd [-s] [-c] [-u]
```

The password command has four variants. The -s flag sets system password. The operator is prompted for a new password twice, both must agree or the password is not changed. If the password is set successfully, the new encrypted password is installed and the operator is left in privileged mode.

The -c flag clears the system password and turns off security. The -u flag causes operator to be unprivileged when it is executed in privileged mode.

If the command is executed with no flags, the operator is prompted for a password (user input is not echoed to the console). If the password matches the system password, the operator becomes privileged. If the password does not match, the operator remains unprivileged and an error counter is incremented.

4.5.4.4.13 Printenv

```
printenv [variable-list]
```

Printenv displays the indicated environment variable on the console terminal. If no variable is specified, all console environment variables are displayed.

4.5.4.4.14 Setenv

```
setenv variable value
```

Setenv assigns a value to the indicated environment variable.

4.5.4.4.15 Test

```
pst_test test_number
```

Test executes the CPU module ROM diagnostic referenced by test_number. The test number must be preceded by a '0x' (that is specified in hex).

4.5.4.4.16 Unsetenv

```
unsetenv variable
```

Unsetenv removes the indicated variable from the set of console environment variables. The environment variables stored in non-volatile memory are not affected by this command.

4.6 Diagnostics

The ROM based diagnostics constitute approximately half of the firmware on the KN220-AA. These diagnostics run automatically on power up (when the CVAX is selected as the power up processor and when in maintenance mode). They can be executed interactively as a whole, or as individual tests. This section summarizes their operation.

The purpose of the ROM based diagnostics is multifaceted:

1. During power up, they determine if enough of the KN220-AA is working to allow the console to run.
2. During the manufacturing process, they verify that the board was correctly built.
3. In the field, they verify that the board is operational, and report all detected errors.
4. They allow sophisticated users and Customer Services technicians to run individual diagnostics interactively, with the intent of isolating errors to the FRU (Field Replaceable Unit).

To accommodate these requirements, the diagnostics have been designed as a collection of individual parameterized tests. A data structure, called a script, and a program, called the diagnostic executive, orchestrate the running of these tests in the right order with the right parameters.

A script is a data structure that points to various tests. There are several scripts, one for the field, and several for manufacturing, depending on where on the manufacturing line the board is. Sophisticated users may also create their own scripts interactively. Additionally, the script contains other information:

- What parameters need to be passed to the test.
- What is to be displayed, if anything, on the console.
- What is to be displayed, if anything, on the LED.
- What to do on errors (halt, loop, or continue).
- Where the tests may be run from. For example, there are certain tests that can only be run from the EEROM. Other tests are PIC (Position Independent Code), and may be run from EEROM, or main memory in the interests of execution speed.

The diagnostic executive "interprets" scripts to determine what tests are to be run. There are several built-in scripts on the KN220-AA that are used for manufacturing, power up, and Customer Services personnel. The diagnostic executive automatically invokes the correct script based on the current environment of the KN220-AA. Any script can be explicitly run with the TEST command from the console terminal.

The diagnostic executive is also responsible for controlling the tests so that when errors occur they can be caught and reported to the user. The executive also ensures that when the tests are run, the machine is left in a consistent and well defined state.

4.6.1 Error Reporting

Before a console is established, the only error reporting is via the KN220-AA diagnostic LEDs (and any LEDs on other boards). Once a console has been established, all errors detected by the diagnostics are also reported by the console. When possible, the diagnostics issue an error summary on the console.

```
?66 2 03 FE 0000 0002                                                    (1)

P1 =00000000  P2 =00000000  P3 =00000000  P4 =00000000  P5 =00000000 (2)*
P6 =00000000  P7 =00000000  P8 =00000000  P9 =00000000  P10=00000000 (3)*
P11=00000000  P12=00000000  P13=00000000  P14=00000000  P15=00000000 (4)*
P16=00000000  P17=00000000  P18=00000000  P19=00000000  P20=00000000 (5)*
gp =05100000  sp =B8007B48  fp =00000000  sr =B048FF04               (6)*
epc=BFC22934  badvaddr =00000000  cause=00000000                     (7)*
```

* The numbers in parenthesis on the right side refer to lines of the display and are not a part of the diagnostic dump. The information on these lines is summarized below.

1. Test summary containing six hexadecimal fields.
   a. ?66, **test** identifies the diagnostic test.
   b. 2, **severity** is the level of a test failure, as dictated by the script. Failure of a severity level 2 test causes the display of this five-line error printout, and halts an autoboot to console I/O mode. An error of severity level 1 displays the first line of the error printout, but does not interrupt an autoboot. Most tests have a severity level of 2.
   c. 03, **error** is a number, that in conjunction with listing files isolates to within a few instructions where the diagnostic detected the error. This field is also called the subtestlog.
   d. FE, **de_error** is a code with which the diagnostic executive signals the diagnostic's state and any illegal behavior. This field indicates a condition that the diagnostic expects on detecting a failure. The possible codes are:

      FF - Normal error exit from diagnostic
      FE - Unanticipated interrupt or exception in diagnostic IC
      FD - Interrupt in cleanup routine
      FC - Interrupt in interrupt handler
      FB - Script requirements not met
      FA - No such diagnostic
      EF - Unanticipated exception in CVAX executive
      EE - Unanticipated exception in R3000 console

   e. 0000, **vector** is the SCB vector (if non-zero) through which an unexpected exception or interrupt trapped when the **de_error** field indicates an unexpected exception or interrupt (FE or EF). This field does not appear on error dumps on the R3000 console
   f. 0002, **count** is the number of previous errors that have occurred. This field does not appear on error dumps on the R3000 console
2. P1...P5 are the first five longwords of the diagnostic state. This is internal information that is used by repair personnel.
3. P6...P10 are the second five longwords of the diagnostic state.
4. P11...P15 are the third five longwords of the diagnostic state.

5. P16...P20 are the last five longwords of the diagnostic state.
6. gp, sp, fp, and sr are R3000 registers

   gp - global pointer
   sp - stack pointer
   fp - frame pointer
   sr - status register

7. epc, badvaddr, cause are R3000 registers

   epc - exception program counter
   badvaddr - bad virtual address
   cause - cause register

4.6.2 Diagnostic Interdependencies

When running tests interactively on an individual basis, users should be aware that certain tests may be dependent on some state set up from a previous test. In general, tests should not be run out of order.

5
System Specification

5.1 Introduction

This chapter gives the functional, electrical, physical, environmental, and performance characteristics of the 220Qn family of DECsystem 5500 upgrade kits, system building blocks, kernels, and supported options.

5.2 General Description

The 220Qx family of systems combines the CPU/Memory subsystem (KN220-AA/MS220-AA) with the Q22-bus based system packaging. Mass storage subsystems, printers, communications and real time devices are added to create systems.

A complete system (i.e., everything needed to support a typical customer application) based on the 220Qx must be made up of:

- A system kernel (e.g., DU55FT1-A2, DU55FT1-A3, etc.,)
- An operating system license (designated by system kernel)
- A power cord (for 240V system kernels only)
- A Hardware Support Kit (for 240V or non-English system kernels only)
- A console terminal
- A mass storage subsystem with fixed and removable media or a DECNet End Node license with an ULTRIX NFS license.
- Supported printers, terminals, and communications as needed to suit the desired application
- User Installation Manuals.

5.2.1 System Kernels

System kernels have a "2-5-2" part number. The three fields define the contents as shown in Figure 5–1.

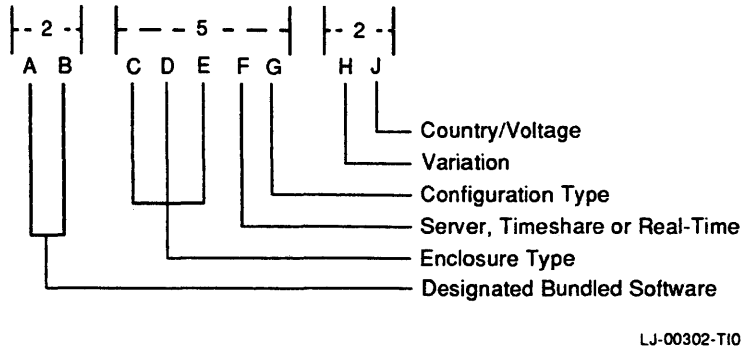


Figure 5–1 System Kernel Part Number

5.2.1.1 Pedestal Systems

The pedestal systems are packaged in 120/240V (70-27437-04) enclosure. The -04 variation includes the plastic panels, access doors and a pedestal base. The BA430 pedestal enclosure measures 27 inches (high) by 21 inches (wide) by 17.8 inches (deep). The BA430 pedestal system is targeted for the office environment and for applications in which 5 ¼ inch RF-type disks, 5 ¼ inch RZ-type disks, and TK-type ½ inch cartridge streaming tape drives provide sufficient capacity, performance and functionality.

5.2.2 220QH Operating System/Network Software Licenses

- Ultrix-32 - V4.1
- Ultrix-32 - V4.1 NFS
- DECNet

5.3 220QH System Base Components

5.3.1 220QH System Building Block

The 220QH system building block is based on the KN220-AA CPU module, the KN220-AA I/O module, and the BA400 series enclosures. The enclosures (70-27437-03,04) feature one 684 watt power supply , and a backplane with twelve Q/CD configured slots. There is mounting space for three fixed and one removable 5 ¼ inch mass storage devices. The SCSI port is assessable via a 50 pin connector on the distributuion panel. The 220QH-Ax is configured for pedestal mounting and the 220QH-Bx is configured for rack mounting.

NOTE
The KN220-AA CPU or I/O modules require two Q-Bus slots, resulting in 10 remaining slots.

There are two variations of the 220QH building block:

1. **220QH-B9 (120-240v/without plastic skins)**
   - 70-27437-03 Enclosure (120-240V - US PowerCord)
     - 54-20181-01 Twelve Slot Backplane
     - One H7874-00 684W Power Supply
   - US Power Cord: 17-00083-43
   - Requires Country Specific Power Cord
   - M7637-AA CPU Module (R3000)
   - M7638-AA I/O Module
   - 17-02700-01,02,03,04 100 pin 2 tap cable x 2
   - H3602-AC Distribution "Cover Panel" for KN220-AA
   - MS220-AA Memory Module
   - H3602-AC Distribution "Cover Panel" for SCSI.

2. **220QH-A9 (120-240v/with plastic skins)**
   - 70-27437-04 Enclosure (120-240V - US Power Cord)
     - 54-20181-01 Twelve Slot Backplane
     - One H7874-00 684W Power Supply
     - Requires Country Specific Power Cord
   - 70-24046-01 Movable Plastic Pedestal
   - 70-24037-01 Plastic Skins
   - 70-24041-01 Plastic Front Panel Assembly
   - M7637-AA CPU Module (R3000)
   - M7638-AA I/O Module
   - 17-02700-01,02,03,04 100 pin 2 tap cable x 2
   - H3602-AC Distribution "Cover Panel" for KN220-AA
   - H3602-AC Distribution "Cover Panel" for SCSI.
   - MS220-AA Memory Module

5.3.1.1 Hardware Support Kits

When ordering a complete system based on the 220QH system building block, the order must include a language specific hardware support kit. This kit includes hardware information, diagnostics on TK50 (TK70 compatible) ½ inch cartridge tape and a license to run the diagnostics. The following table illustrates the hardware support kits for the 220QH system building block and shows Country/Language designations.

220QH HARDWARE SUPPORT KITS

ZNAXX-XX	220QH support kit for US/English; includes 220QH hardware information kit (ZNAXX-XX), diagnostics on TK50 streaming tape (ZNAXX-XX), including a license to use these diagnostics.
ZN*XX-XX	220QH support kit for * (where * is country/ language designation shown below); with 220QH hardware information kit (ZN*XX-XX), diagnostics on TK50 streaming tape (ZN*XX-XX), including a license to use these diagnostics.

* Definition Country/Language

A	US/English	N	Norway/Norwegian
B	Belgium/Flemish	O	Unused
C	Canada/French	P	France/French
D	Denmark/Danish	Q	Canada/English
E	UK, Ireland/English	R	S. America/Spanish
F	Finland/Suomi	S	Spain/Spanish
G	Germany, Austria/German	T	Israel/Hebrew
H	Holland/Dutch	U	S. America/Portuguese
I	Italy/Italian	V	Portugal/Portuguese
J	Japan/Katakana	W	Switzerland/Italian
K	Switzerland/French	X	Reserved
L	Switzerland/German	Y	Japan/Hiragana
M	Sweden/Swedish	Z	Australia/English, New Zealand/English

5.3.1.2 Power Cords

A US power cord BN20A-2E (17-00083-43) is included in 120 volt system building blocks (220QH-B2, or C2).

One of the following line cords must be purchased when ordering a 240 volt system building block (220QH-B3 or C3):

240 VOLT POWER LINES

BN20E-2E	Right angle entry IEC 320 to right angle entry UK/Ireland Fused Plug, 3 wires, 10A @ 250V, 70C, molded, 2.5 meters (17-00209-04).
BN20D-2E	Right angle entry IEC 320 to straight entry Schuko Plug, 3 wires, 10A @ 250V, 70C, molded, 2.5 meters (17-00199-07). For Austria, Belgium, Czech, Finland, France, Germany, Hungary, Netherlands, Norway, Poland, Portugal, Spain and Sweden.
BN20F-2E	Right angle entry IEC 320 to straight entry Swiss (type 12) Plug, 3 wires, 10A @ 250V, 70C, molded, 2.5 meters (17-00210-04).
BN20C-2E	Right angle entry IEC 320 to straight entry Australia Plug, 3 wires, 10A @ 250V, 70C, molded, 2.5 meters (17-00198-04). Also for New Zealand.
BN20H-2E	Right angle entry IEC 320 to straight entry Denmark Plug, 3 wires, 10A @ 250V, 70C, molded, 2.5 meters (17-00310-07).
BN20B-2E	Right angle entry IEC 320 to straight entry Japan/USA Fused Plug, 3 wires, 10A @ 250V, 70C, molded, 2.5 meters (17-00083-44).
BN20K-2E	Right angle entry IEC 320 to right angle entry India, So. Africa Fused Plug, 3 wires, 10A @ 250V, 70C, molded, 2.5 meters (17-00456-05).
BN20L-2E	Right angle entry IEC 320 to right angle entry Isreal Fused Plug, 3 wires, 10A @ 250V, 70C, molded, 2.5 meters (17-00457-05).
BN20J-2E	Right angle entry IEC 320 to straight entry Italy Plug, 3 wires, 10A @ 250V, 65C, molded, 2.5 meters (17-00364-05).

5.3.2 220QH Physical Specifications

The physical specifications for the 220QH-Bx,-Fx pedestal system (with plastic skins) are as follows:

Height	27.0 in.	69 cm
Width	21.0 in.	53 cm
Depth	17.8 in.	45 cm
Weight[1]	129.0 lb.	58.6 kg

[1]Weight given is for typical BA430 with 1 power supply, fully loaded with module options (8 quad height and 4 dual height), 3 hard disks, 1 tape drive, and plastic skins.

The Mechanical Specifications for the 220QH-Cx,-Hx rack mount system are as follows:

Height	24.4 in.	62 cm
Width	17.5 in.	44 cm
Depth	11.5 in.	29 cm
Weight[1]	105.0 lb.	47.7 kg

[1]Weight given is for typical BA430 with 2 power supplies, fully loaded with module options (8 quad height and 4 dual height), 3 hard disks, and 1 tape drive.

5.3.3 220QH Electrical Specifications

For DC power supply specifications refer to Section 5.6.1.4.

For AC input specifications refer to section Section 5.6.1.5.

5.3.4 220QH Environmental Specifications

5.3.4.1 220QH Temperature/Humidity Specifications

Operating Temperature	10C to 40C (50F to 104F)
Operating Humidity	20% to 80%
Maximum Wet Bulb (operating)	25C (77F)
Minimum Dew Point (operating)	2C (36F)
Storage Temperature	-40C to 66C (-40F to 151F)
Storage Humidity	10% to 95%
Mocks Dry Bulb Temp Change (operating)	20C(11F) per hr)

Maximum allowable operating temperatures should be derated by 1.80C per 1000m (1.00F per 1000 ft) for operation at altitudes above sea level.

5.4 220QF System Base Components

5.4.1 220QF System Upgrade

The 220QF cabinet system upgrade is based on the KN220-AA DECsystem 5500 CPU and I/O module, mounted in a 70-26188-01,-02 cabinet-box. The 220QF also provides mounting space for one 10 ½ inch high full rack device or two 10 ½ inch high half rack devices (i.e., 2 RA90's). The cabinet-box features twelve Q/CD configured backplane slots plus mounting space for four 5¼ inch mass storage devices (three fixed media and one removable media). The SCSI port is assessable via a 50 pin connector on the distribution panel. (No SCSI devices are mounted internal.)

NOTE
The KN220-AA CPU or I/O modules require two Q22-bus slots, resulting in 10 remaining slots

There are two variations of the 220QF upgrade:

1. **220QF-A2 (120v)**
   - 70-26188-01 Cabinet
     - 874-D Power Controller (120V)
     - Twelve Slot Backplane
     - One H7874-00 120V/684W Power Supplies
   - M7637-AA-AA CPU Module (R3000)
   - M7638-AA-AA I/O Module
   - 17-02700-01,02,03,04 100 pin 2 tap cable x 2
   - H3602-AC Distribution "Cover Panel" for KN220-AA
   - H3602-AC Distribution "Cover Panel" for SCSI.
   - MS220-AA Memory Module

2. **220QF-A3(240v)**
   - 70-26188-02 Cabinet
     - 874-F Power Controller (240V)
     - Twelve Slot Backplane
     - One H7874-00 240V/684W Power Supplies
     - Requires Country Specific Power Cord
   - M7637-AA CPU Module (R3000)
   - M7638-AA I/O Module
   - 17-02700-01,02,03,04 100 pin 2 tap cable x 2
   - H3602-AC Distribution "Cover Panel" for KN220-AA
   - H3602-AC Distribution "Cover Panel" for SCSI.
   - MS220-AA-BA Memory Module

5.4.1.1 Hardware Support Kits

When ordering a complete system based on the 220QF system building block, the order must include a language specific hardware support kit. This kit includes hardware information, diagnostics on TK50 (TK70 compatible) ½ inch cartridge tape and a license to run the diagnostics. The following table illustrates the hardware support kits for the 220QF system building block and shows Country/Language designations.

220QF HARDWARE SUPPORT KITS

ZNAXX-XX	220QF support kit for US/English; includes 220QF hardware information kit (ZNAXX-XX), diagnostics on TK50 streaming tape (ZNAXX-XX), including a license to use these diagnostics.
ZN*XX-XX	220QF support kit for * (where * is country/ language designation shown below); with 220QF hardware information kit (ZN*XX-XX), diagnostics on TK50 streaming tape (ZN*XX-XX), including a license to use these diagnostics.

* Definition Country/Language

A	US/English	N	Norway/Norwegian
B	Belgium/Flemish	O	Unused
C	Canada/French	P	France/French
D	Denmark/Danish	Q	Canada/English
E	UK, Ireland/English	R	S. America/Spanish
F	Finland/Suomi	S	Spain/Spanish
G	Germany, Austria/German	T	Israel/Hebrew
H	Holland/Dutch	U	S. America/Portuguese
I	Italy/Italian	V	Portugal/Portuguese
J	Japan/Katakana	W	Switzerland/Italian
K	Switzerland/French	X	Reserved
L	Switzerland/German	Y	Japan/Hiragana
M	Sweden/Swedish	Z	Australia/English, New Zealand/English

5.4.1.2 220QF Power Cords

A US power cord is included in a 220QF-A2 120 volt system building block.

When ordering a 220QF-A3 240 volt Cabinet System building block, the customer must also purchase one of the following 200/220/230/240V line cords:

BN18B-4E	UK / Ireland - 240V
BN18C-4E	Central European - 220V
BN18D-4E	Australia / New Zealand - 230/240V
BN18E-4E	Italy - 220V
BN18F-4E	Israel - 230V
BN18H-4E	India - 220V
BN18T-4E	US - 240V
BN18T-4E	Japan - 100/200V

NOTE

These are the detachable power cords for the 874F power controller internal to the cabinet.

5.4.2 220QF Physical Specifications

The physical specifications for the 220QF cabinet system are as follows:

Height	106.8 cm	41.60 in
Width	54.0 cm	21.25 in
Depth	80.0 cm	31.50 in
Weight[1]	204.5 kg	450 lb

[1]System weight of 220QF configured with RA90, fully loaded backplane, 3 RA-type hard disks, and TK70.

5.4.3 220QF Electrical Specifications

5.4.3.1 220QF DC Output Specifications

For the DC power supply specifications on the 220QF refer to Section 5.4.3.3.

5.4.3.2 220QF AC Input Specifications

The AC input specifications for the 220QF cabinet system are listed in the following tables and can vary depending on what disk, tape (or both) subsystem and options are used.

The maximum values in the tables are based on power controller maximums, and typical values are based on measurements from a system configured as follows:

CPU	KN220	Disk	KDA50, 2 RA70s
Memory	2 MS220-AAs	Disk Subsystem	RA90
Async. Comm.	CXA16	Ethernet	KN220-AA
Tape	TQK70,TK70	Load Modules	4 M9060s

PARAMETER	MIN	TYP	MAX	UNITS	SYMBOL
Voltage Nominal	–	101	–	volts	V
Voltage Design Range	90	–	110	volts	V
RMS current @ typical voltage	–	11.5	28.5	amperes	A
Voltage Nominal	–	120	–	volts	V
Voltage Design Range	104	–	132	volts	V
RMS current @ typical voltage	–	11.1	24	amperes	A
Voltage Nominal	–	240	–	volts	V
Voltage Design Range	180	–	264	volts	V
RMS current @ typical voltage	–	5.8	12	amperes	A
Frequency Nominal	–	50/60	–	hertz	Hz
Frequency Range	47	–	63	hertz	Hz
Number of Phases	–	1	–	none	NA
RMS ground current	–	–	–	millamperes	ma
Peak Current (steady-state) 101 V	–	26	–	amperes	A
Peak Current (steady-state) 120 V	–	25	–	amperes	A
Peak Current (steady-state) 220 /240 V	–	15	–	amperes	A
120V AC Cord Type				CSA 12-3 Type	
120V Power Cord Length	–	4.5	–	meters	m
120V Power Cord Length	–	180	–	inches	in
120V AC Plug Type				NEMA L5-30P	

PARAMETER	MIN	TYP	MAX	UNITS	SYMBOL
240V AC Plug Type		Country Specific, refer to Section 5.4.1.2			
Ride-through Time	80	–	–	millisecond	ms
Start-up Current Amplitude 101 V	–	–	114	rms amperes	A
Start-up Current Amplitude 120 V	–	–	96	rms amperes	A
Start-up Current Amplitude 220 /240 V	–	–	48	rms amperes	A
Start-up Current Duration	–	–	1.3	seconds	s
Start-up Current Amplitude 101 V	–	–	57	rms amperes	A
Start-up Current Amplitude 120 V	–	–	48	rms amperes	A
Start-up Current Amplitude 220 /240 V	–	–	24	rms amperes	A
Start-up Current Duration	–	–	10	seconds	s
Start-up Current Amplitude 101 V	–	–	36	rms amperes	A
Start-up Current Amplitude 120 V	–	–	30	rms amperes	A
Start-up Current Amplitude 220 /240 V	–	–	15	rms amperes	A
Start-up Current Duration	–	–	50	seconds	s
120V Power Consumption	–	1064	–	watts	W
240V Power Consumption	–	1052	–	watts	W
120V Apparent Power	–	1346	2880	volt amperes	VA
240V Apparent Power	–	1384	2880	volt amperes	VA
DC Output Watts Available	–	–	670	watts	W
Power Factor	–	.78	–	none	PF
120V Crest Factor	–	2.2	–	none	CF
240V Crest Factor	–	2.6	–	none	CF
120V Current Distortion Factor	–	46	–	per cent	%
240V Current Distortion Factor	–	57	–	per cent	%

5.4.3.3 Power Supplies and Power Controller

The DC Power supplies for the 220QF system are located in the BA430-Bx rack mount box, and any rack mounted mass storage devices in the cabinet.

Power Controller Specifications

Specification	874-D	874-F
Maximum Amperage	24A	12A
Voltamps at Full Load	2880VA	2880VA
Circuit Breaker Rating	30A	15A
Voltage Range	90V-120V	180V-264V
Frequency Range	57-63 Hz	47-53 Hz
No. of Switched Outlets	4	4
No. of Unswitched Outlets	2	2
Amperage Rating per Outlet	16A	12A
Outlet Type	NEMA 5-20R	NEMA 5-15R

The power controller (874-D or 874-F) is mounted at the bottom rear of the H9644 cabinet and is the center point of AC power distribution for all the options installed in the cabinet.

The power controller contains two unswitched power outlets, six switched power outlets on the front, four remote sense input connectors, a three position toggle switch, an indicator lamp, and a circuit breaker on the back. The unswitched power outlets are always energized when the power is on. The switched power outlets are controlled by the three position toggle switch and the devices connected to the remote sense input connectors.

When the toggle switch is in the 'A' (remote) position, the switched power outlets on the power controller are controlled by the devices connected to the remote sense input connectors. When the toggle switch is in the '0' (off) position, the switched power outlets on the power controller are always de-energized. When the toggle switch is in the 'B' (on) position, the switched power outlets on the power controller are always energized. The toggle switch should always be left in the 'A' (remote) position so that the on/off switch on the front of the cabinet controls power for the entire system.

NOTE

In the case of the expanded cabinet system, both power controllers would have the toggle switch in the 'A' (remote) position, and be interconnected by a remote sense cable (pn# 17-08288-06) as shown in Figure 5–2.

The power controller is protected by a circuit breaker. If the circuit breaker trips, all outlets are de-energized, and the entire system loses power. The circuit breaker can be reset by pushing it down, then lifting it up.

The indicator lamp at the back of the power controller remains lit as long as AC power is present at the input of the power controller. It is unaffected by the operation of the circuit breaker.

The interconnection between the 874-x, BA430-Bx system enclosure, and system options is shown in Figure 5–2.

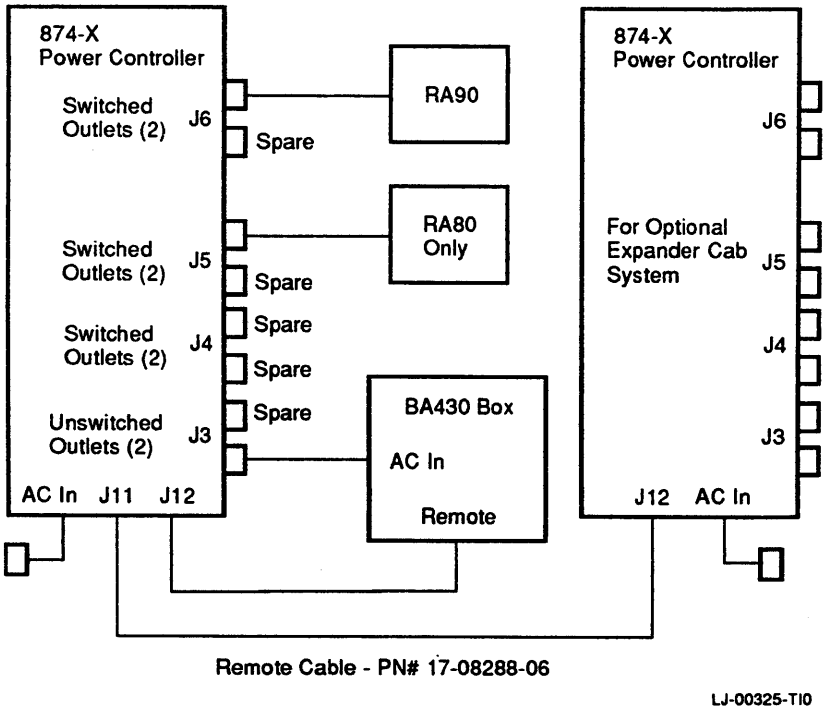


Figure 5–2 BA430-Bx System Enclosure Interconnections

5.4.4 220QF Environmental Specifications

	Operating	Nonoperating
Maximum Altitude	2.4 km (8000 ft)	4.9 km (16000 ft)
Temperature range[1]	5 - 32C (59 - 90F)	-40 - 66C (-40 - 151F)
Temperature change rate	11C/hour (20F/hour)	-
Relative humidity	20% - 80%	10% - 95%
Mocks Dry Bulb Temp Change (operating)	20C(11F) per hr)	

[1]Reduce the maximum temperature specification by 1.8C for each 1000 m. (1C for each 1000 ft.) increase in altitude

5.5 Expansion Cabinet Systems

5.5.1 DSSI Expansion Enclosures

NOTE
This section describes DSSI expansion. The following information is subject to change.

Figure 5–3 shows the BA400x Q-bus and disk expansion enclosure.

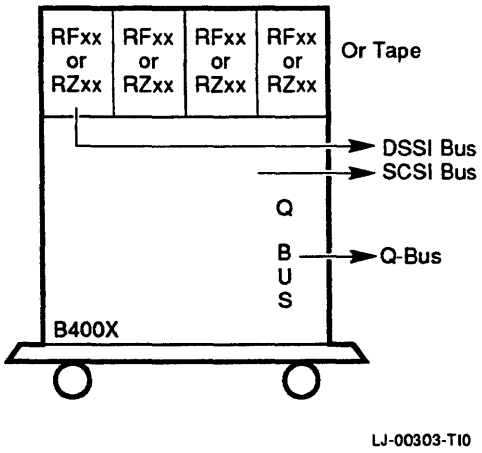


Figure 5–3 BA400x Q-Bus and Disk Expansion Enclosure

- Notes:
  - Support for 10 Q-bus slots and 3 disk drives
  - Tape cavity could be used for a TK70, or disks.
  - Device Unit Numbers are from right to left.
  - Drive types are SCSI or DSSI.
  - Base system loses one Q-bus slot to expand into this enclosure.

5.5.2 DSSI Based DU-55xxx Configurations and Possible Expanded Configurations

Figure 5–4 shows the DU-55xxx pedestal base system.

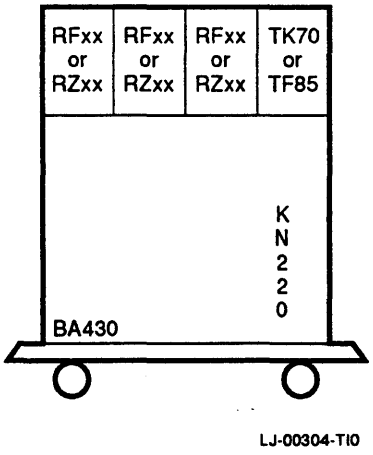


Figure 5–4 DU-55xxx Pedestal Base System

DU-55xxx pedestal (1 to 4 Disks, 400 Mbytes to 1.2 Gbytes) base system, with DSSI bus, SCSI bus and Q22-bus.

- Notes:
  - Support for 12 Q22-bus slots and 3 disk drives
  - Tape cavity could be used for a TK70, or disks.
  - Device Unit Numbers are from right to left.
  - Drive types are SCSI or DSSI.
  - Base system loses one Q22-bus slot to expand into this enclosure.
    - The enclosure is DSSI or SCSI expansion ready for an additional 3-4 RZxx or RFxx drives external to the BA430. (3 if TF85 used, 4 if TK70 used)

Figure 5–5 shows the DU-55xxx system with R215F storage expansion.

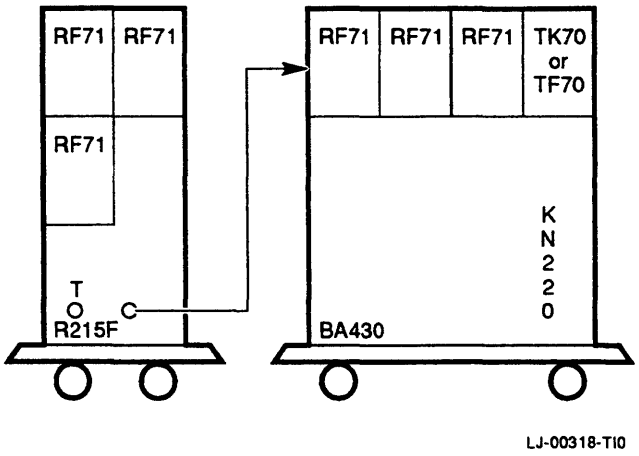


Figure 5–5 DU-55xxx Pedestal with R215F Storage Expansion

DU-55xxx pedestal base system with R215F expansion and a Maximum of 6 RF71 disks, 2.4 Gbytes.

- Notes:
  - Base system is expansion ready no changes needed for this configuration.

Figure 5–6 shows the DU-55xxx system with a R400x storage enclosure.

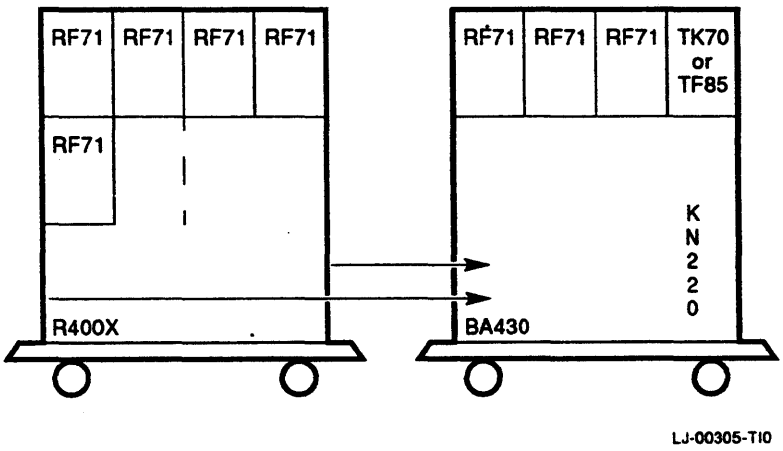


Figure 5–6 DU-55xxx Pedestal with R400X Storage Enclosure

- Notes:
  - This system requires the addition of a KFQSA-SA
  - The SCSI bus can be expanded using the R400X also.

Figure 5–7 shows the DU-55xxx system with B400x Q22-bus and storage expansion.

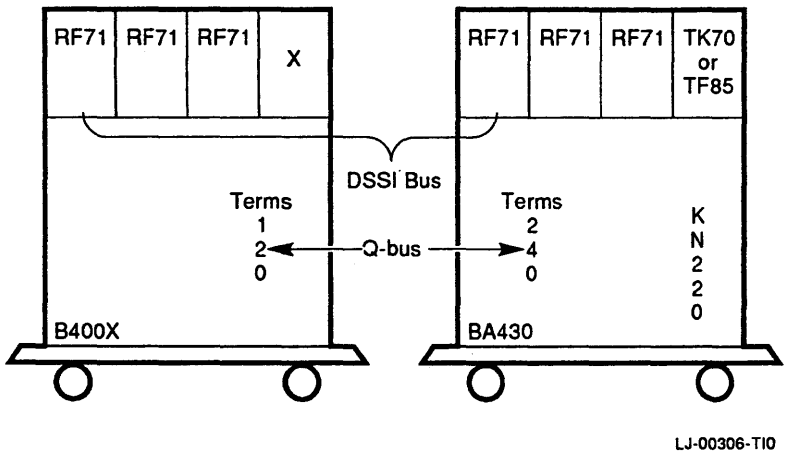


Figure 5–7 DU-55xxx Pedestal with B400x Q22-bus and Storage Expansion

- Notes:
  - The disk drives can be either DSSI or SCSI types.
  - The maximum supported for each bus will be 7 drives.
  - Each BA430 system can have an additional R400X for drives.

DU-55xxx pedestal with R400X expansion and a maximum of 7 RF71 disks with 2.8 Gbytes.

- Notes:
  - 10 Q-bus slots available (2 used by Q-bus expansion cards)
  - R400x tape cavity could hold a TK70 drive if customer wanted to add a TK70 or other Disk drives..

5.5.3 DU-55xxx RA Based Cabinet Systems and Expansion Paths

Figure 5–8 shows the DU55xxx RA based system.

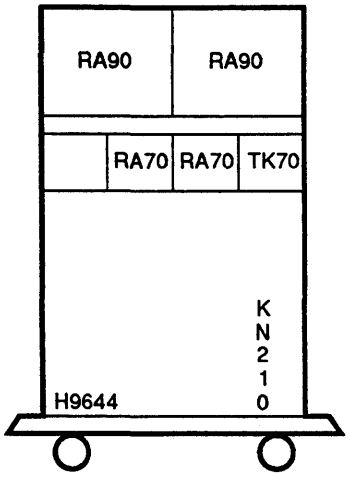


Figure 5–8 DU-55xxx RA Cabinet Base System

DU-55xxx RA cabinet Base system with 1 or 2 RA90 (1.2 Gbytes 2.96 Gbytes).

Figure 5–9 shows the DU-55xxx RA system with a H9644 expansion cabinet.

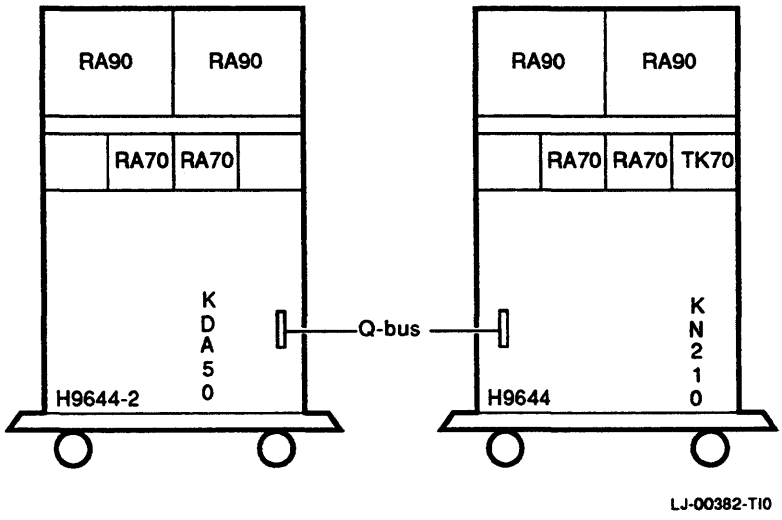


Figure 5–9 DU-55xxx RA Base System with H9644 Expansion Cabinet

DU-55xxx cabinet expanded with H9644-2 expansion RA based cabinet. The system has 22 Q22-bus slots and 5.92 Gbytes storage.

Figure 5–10 shows the DU-55xxx RA system with a SA600 expansion cabinet.

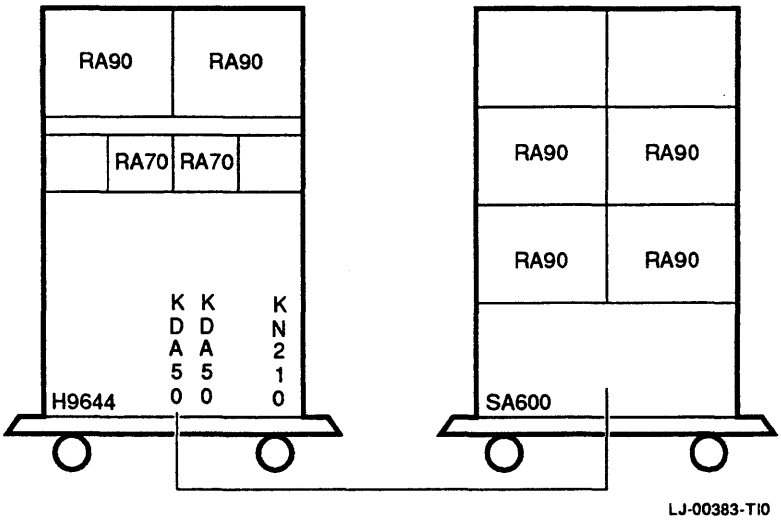


Figure 5–10 DU-55xxx RA Dase System with SA600 Expansion Cabinet

DU-55xxx cabinet expanded with SA600 expansion cabinet. Total storage 7.76 Gbytes.

5.5.4 Expansion Cabinet Specifications

Electrical requirements for expansion cabinet (TU81E with RA82) are shown below.

Nominal AC Voltage	101	120VAC	240 VAC
Voltage range	90 to 110 VAC	90 to 128 VAC	184 to 256 VAC
Power source phase	Single	Single	Single
Nominal frequency	50-60 Hz	50-60 Hz	50-60 Hz
Frequency range	47-63 Hz	47-63 Hz	47-63 Hz
Maximum steady state current at typical voltage	N/A	8.0A	4.1A
Maximum steady state current at maximum voltage	N/A	8.3A	4.6A
Startup current for 150ms	N/A	44A	33A
Power consumption (typical)	N/A	620W	620W
Power consumption (maximum)	N/A	960W	960W

Electrical Requirements for the expansion cabinet only, are shown below.

Nominal AC Voltage	101	120VAC	240 VAC
Voltage range	90 to 110 VAC	90 to 128 Vac	184 to 256 VAC
Power source phase	Single	Single	Single
Nominal frequency	50-60 Hz	50-60 Hz	50-60 Hz
Frequency range	47-63 Hz	47-63 Hz	47-63 Hz
Maximum steady state current at typical voltage	10.2A	8.6A	4.7A
Maximum available current for cabinet	28.5A	24A	12A
Startup current for 10 seconds	57A	48A	24A
Power consumption (typical)	670W	670W	670W
Power consumption (maximum limit of the power controller)	2880W	2880W	2880W

5.6 BA430 Enclosure Specifications

5.6.1 General Description

The BA430 (70-27437-03,-04) enclosure contains a 12 slot backplane, one 684W power supplies, and two 24 Vdc fans.

The BA430 for the DECstation 5500 system comes in two variations. The 70-27437-01 uses the H7874-00 power supply, and is intended for use with 120V/24V circuits.

The 70-27437 requires versions of Q22-bus modules that interface to external devices. These modules have integral I/O distribution inserts which eliminate internal cables and facilitate module removal. Figure 5–11 show the front view of the enclosure.

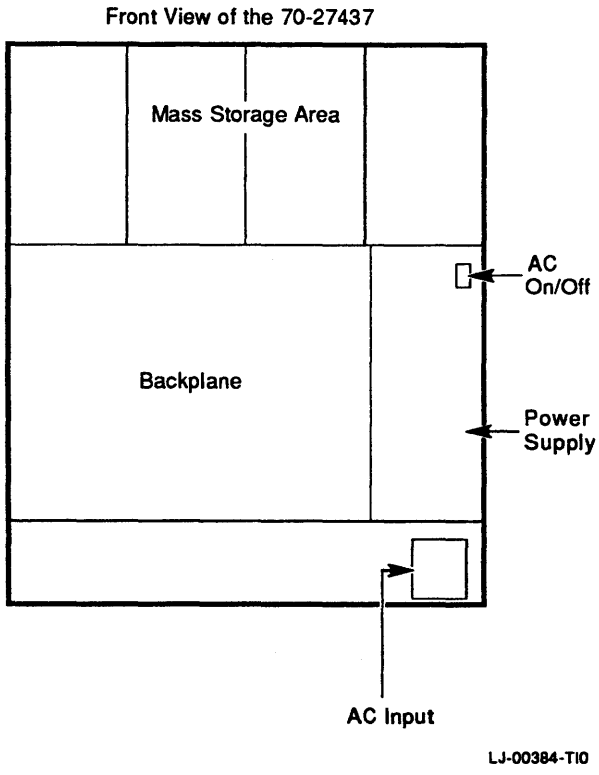


Figure 5–11 BA430 Enclosure Front View

5.6.1.1 Mass Storage

The 70-27437 has a mass storage area with four mounting slots. The enclosure use drive mounting that connects directly to the backplane, eliminating external cabling. Each slot has individual shock mounting so that each mass storage can be removed or installed separately. These slots will be configured according to what disk controller/adapter is used in the system.

A mass storage distribution cover is included with the hardware and can be accessed by removing screws at the top of the plastic cover that holds the cover in place. This configuration is used in conjunction with a TK type tape controller, and will house three RFxx/RZxx disks and one TKxx tape drive.

The I/O module for the KN220 will support DSSI and SCSI type disk/tape drives. The KN220 I/O module contains an adapter that supports the DSSI/SCSI architecture. Each mass storage device (RFxx or TFxx) is "daisy-chained" by means of the backplane. This configuration will support a maximum of six full-height (or half-height) RFxx ISE's, and one TFxx tape drive for a maximum of seven logical devices. The I/O module for the KN220 will also support SCSI (RZxx) type disk/tape drives. The KN220 I/O module contains an adapter that supports the SCSI architecture. Each mass storage device (RZxx or TZxx) is "daisy-chained" by means of the backplane. This configuration will support a maximum of six full-height (or half-height) RZxx disk drives, and one TZxx tape drive for a maximum of seven logical devices.

NOTE
The SCSI drives can be internal to the BA430 enclosure as well as the DSSI drives.

The following are the maximum formatted internal disk capacities:

3 RZ56 drives	1800 Mbytes
3 RF71 drives	1800 Mbytes

RZ, and RF type drives can be mixed in a single enclosure. This is due to the unique connectors for each of the ISEs.

External RAxx drives are supported on a single KDA50 controller, external RFxx drives can be added off the DSSI external port using the R400X or B400X expanders.

Figure 5–12 shows the supported configurations for an TF85, RF71 or TK70 drives.

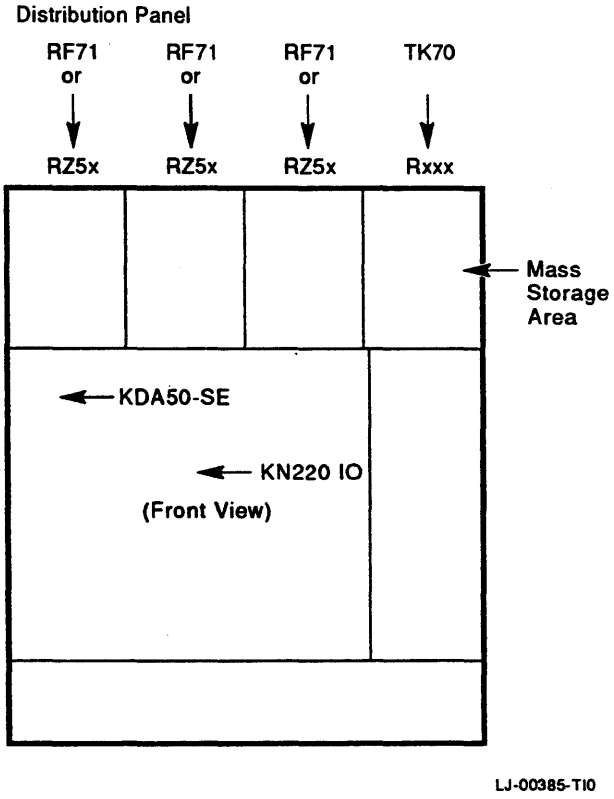


Figure 5–12 BA430 Enclosure Mass Storage Configuration

5.6.1.2 Backplane Specifications

The backplane (54-20181-01) is a 16.0" x 20.5 .125" assembly consisting of a multi layer double sided etch board with 1" center-to-center connections, for up to 12 Q22-bus quad height modules. The backplane is bounded and therefore has no expansion capabilities within the enclosure.

There are two power connectors for the H7874-00 power supply to the backplane. Ground and Vcc are common throughout the backplane.

Number of Q/CD Slots	12 quad
Number of Q/Q Slots	0

Total Slots	12 quad
Available AC Loads[1]	40
Available DC Loads	20
Termination	120 ohms

[1]Available AC loads assume that 240 ohm termination on single KN220 CPU with 120 ohm termination on backplane.

Figure 5–13 shows the general outline, the priorities, and the power supply connectors of the backplane.

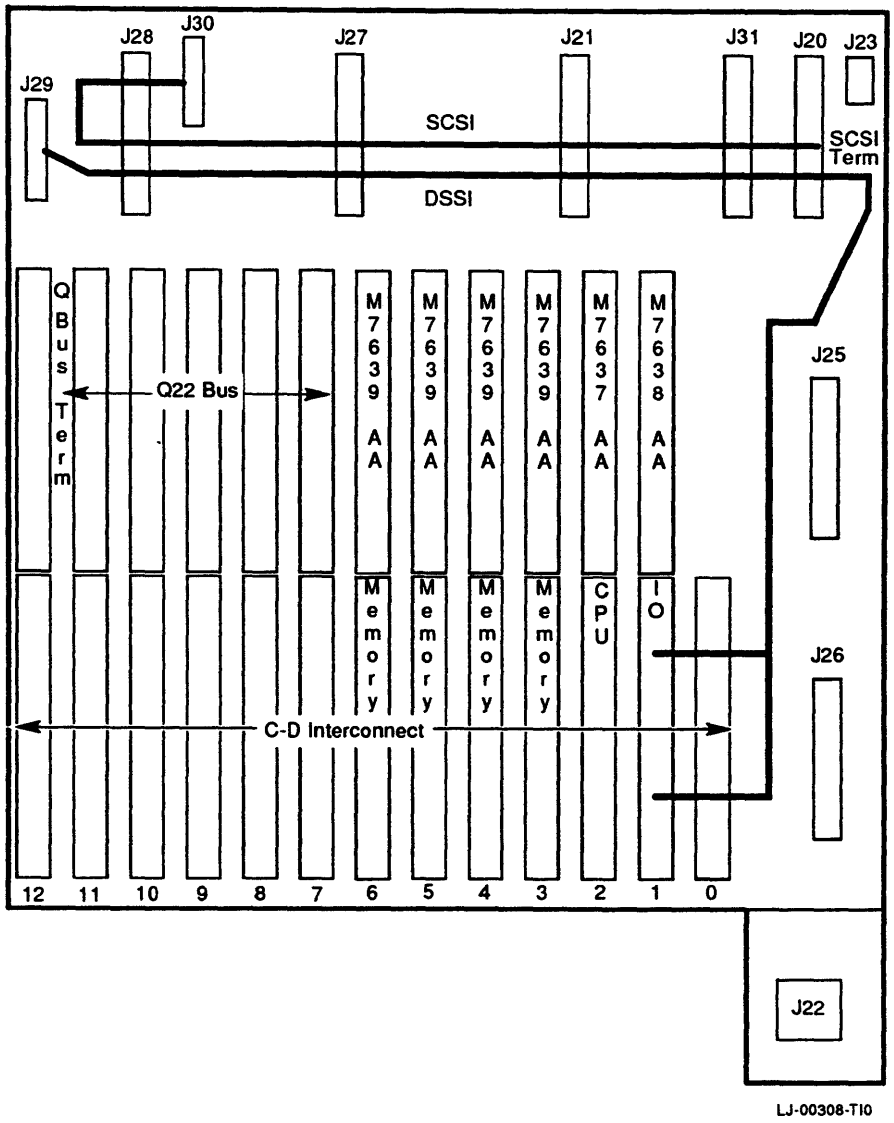


Figure 5–13 BA430 Enclosure Backplane Layout

5.6.1.3 Power Supply Specifications

Both variations of the 70-27437 have two (H7874-00) power supply. The supply is capable of delivering 684 watts of +5 and +12 volts. The power supply will power both of the DC fans. The speed of the fans vary as the ambient temperature varies.

Power for mass storage devices is provided via the backplane connector. The supply also powers the four mass storage slots.

On the front cover of the power supply is a green LED and a over temperature indicator. This LED is lit when the DC voltages are within regulation. Also on the front cover there is also a CPU restart switch and power controller outputs.

NOTE

The H7874-00 power supply is auto voltage selectable, and supports 120V and 240V operation.

The 70-27437-03/04 will use the H7874-00 120V/240V power supply.

5.6.1.4 DC Power Supply Specification

Conditions	5.1V Nominal Output	+12.1V Nominal Output
current - min. load	8.0 amps	
current - max. load	60.0 amps	18.0 amps
ripple & noise	50mv p-p max	75mv p-p max
total regulation	13.0 %	13.0%
min.	+4.95V	+11.74V
typ.	+5.10V	+12.10V
max.	+5.25V	+12.46V
long term stability	0.4%/1000 hrs	0.4%/1000 hrs

5.6.1.5 AC Input Specification

PARAMETER	MIN	TYP	MAX	UNITS	SYMBOL
Voltage Nominal	–	101	–	volts	V
Voltage Design Range	90	–	110	volts	V
RMS Current @Nominal (steady state)	1.3	–	12.0	amperes	A
Voltage Nominal	–	120	–	volts	V
Voltage Design Range	104	–	128	volts	V
RMS Current @Nominal (steady state)	1.07	–	10.0	amperes	A
Voltage Nominal	–	220 /240	–	volts	V
Voltage Design Range	190	–	256	volts	V
RMS Current @Nominal (steady state)	.58	–	5.0	amperes	A
Frequency Nominal	–	50/60	–	hertz	Hz
Frequency Range	47	–	63	hertz	Hz
Number of Phases	–	1	–	none	NA
RMS ground current	–	–	.27	millamperes	ma
Peak Current (Steady State) 101 V	–	xx.0	–	amperes	A
Peak Current (Steady State) 120 V	–	xx.7	–	amperes	A
Peak Current (Steady State) 220 /240 V	–	x.0	–	amperes	A
Power Cord Type		IEC 320 C16 (10A @ 250 V or 15 A @ 125 V)			
Power Cord Length	–	1.9	–	meters	m
Power Cord Length	–	75	–	inches	in
AC Plug Type		country specific, refer to A-SP-EL00002-TB-0			
Ride-through Time	20	–	–	millisecond	ms
Inrush Current	–	–	100	amperes peak	A
Start-up Current Amplitude 101 V	–	–	13.7	rms amperes	A

PARAMETER	MIN	TYP	MAX	UNITS	SYMBOL
Start-up Current Amplitude 120 V	–	–	11.5	rms amperes	A
Start-up Current Amplitude 220 /240 V	–	–	5.8	rms amperes	A
Start-up Current Duration	–	–	30	seconds	s
Power Consumption	60	–	876	watts	W
Apparent Power	86	–	1471	volt amperes	VA
DC Output Watts Available	–	–	600	watts	W
Circuit Breaker Rating per pwr supply	–	15.0	–	amperes	A
Power Factor	–	.7	–	none	PF
Current Crest Factor	–	3.0	–	none	CF
Voltage Crest Factor	1.37	–	–	none	CF

5.6.1.5.1 Airflow Specifications

The 70-27437 is equipped with two 7 $\frac{1}{2}$ inch DC fans. The speed of these fans is determined by a temperature sensor in the power supply. This sensor measures ambient air temperature. The temperature vs airflow transfer characteristics are shown in the table below.

Temperature	Fan Voltage	Airflow Per Slot	Airflow Per Power Supply	Airflow Per Drive
above				
41°C + /-3°C	24.0V	X CFM	XX CFM	XX CFM
		200 LFM		
less than				
>23°C 13°C	12.5V	XX CFM	XX CFM	XX CFM
		135 LFM		

NOTE

These values are approximate design numbers. Actual thermocouple measurements of devices should be used to verify cooling efficiency.

5.6.1.5.2 Physical Specifications

	Rack Mount	Pedestal	Units	Symbol
Height	.62	.69	meters	m
	24.4	27.0	inches	in
Width	.44	.53	meters	m
	17.5	21.0	inches	in
Depth	.29	.45	meters	m
	11.5	17.8	inches	in
Weight	50	68.2	kilograms	kg
	110	150	pounds	lb
	Minimum Service Clearance Required			
front	1	1	meters	m
front	39.4	39.4	inches	in
rear[1]	0	0	meters	m
rear	0	0	inches	in
left side	0	0	meters	m
left side	0	0	inches	in
right side	0	0	meters	m
right side	0	0	inches	in
	Minimum Clearance Required for Airflow			
front	5.1	5.1	centimeters	cm
front	2.0	2.0	inches	in
rear	5.1	5.1	centimeters	cm
rear	2.0	2.0	inches	in
left side	5.1	5.1	centimeters	cm
left side	2.0	2.0	inches	in
right side	5.1	5.1	centimeters	cm
right side	2.0	2.0	inches	in

[1]One Meter (39.4 inches) for backplane service only

5.6.1.5.3 Environmental Specifications

PARAMETER	MIN	TYP	MAX	UNITS	SYMBOL
Operating Temperature	10	–	40	Celsius	C
Operating Temperature	50	–	104	Fahrenheit	F
Nonoperating Temperature	-40	–	66	Celsius	C
Nonoperating Temperature	-40	–	151	Fahrenheit	F
Storage Temperature	-40	–	66	Celsius	C
Storage Temperature	-40	–	151	Fahrenheit	F
Temperature Rate of change	–	–	11	Celsius per hour	C/hr
Temperature Rate of Change	–	–	20	Fahrenheit per hour	F/hr
Operating Relative Humidity	20	–	80	Relative Humidity (non-condensing)	%RH
Nonoperating Relative Humidity	10	–	90	Relative Humidity (Noncondensing)	%RH
Operating Max. Wet Bulb Temperature	–	–	25 77	Celsius Fahrenheit	C F
Heat Dissipation	90	–	876	Watts	W
Airflow Intake and Exhaust Location	Intake Location: Left side, right side, front and rear all located at the upper part of the box.			Exhaust Location: Right side, rear, bottom and front all located at the lower part of the box	
Operating Altitude	–	–	2438	meters	m
Nonoperating Altitude	–	–	8000	feet	ft
Nonoperating Altitude	–	–	4877	meters	m
Operating Mechanical Shock	Duration		16000 /10	feet /milliseconds	ft/ms
Operating Mechanical Shock	Level		10	gravities	g
Operating Vibration Freq. Range	22		500	Hertz	Hz
Operating Vibration Vibra.	0.01"da		.25g		
Floor Mount with Skins Freq. Range	5		22	Hertz	Hz
Class A/B Vibra.	.01"da	–	.25g		
Operating Vibration Freq. Range	30		500	Hertz	Hz
Operating Vibration Vibra.	0.01"da		.50g		

PARAMETER	MIN	TYP	MAX	UNITS	SYMBOL
Class C Vibra.	.01"da		.50g		
Nonoperating Mechanical Shock	Duration		30	milliseconds	ms
Nonoperating Mechanical Shock	Level		40	gravities	g
Nonoperating Vibration Freq. Range					
Nonoperating Vibration Vibra. Level					
Acoustics - Operating LNPE	–	x.x	–	Bels	B
Acoustics - Operating LNA	–	xx	–	Decibels	dBA
Acoustics - Idle LNPE	–	x.x	–	Bels	B
Acoustics - Idle LNA	–	xx	–	Decibels	dBA

5.7 BA213 Enclosure Specifications

NOTE
The BA213 enclosure is field upgradeable only

5.7.1 General Description

The BA213 (70-24227-03,-04) enclosure contains a 12 slot backplane, two 230W power supplies, and two fans.

The BA213 for the 5500 comes in two variations. The 70-24227-03 uses the H7868-A power supply, and is intended for use with 120V circuits. The 70-24227-04 uses the H7868-B power supply, and is intended for use with 240V circuits.

The 70-24227 requires new versions of Q22-bus modules that interface to external devices. These modules have integral I/O distribution inserts which eliminate internal cables and facilitate module removal. Figure 5–14 shows the front view of the enclosure.

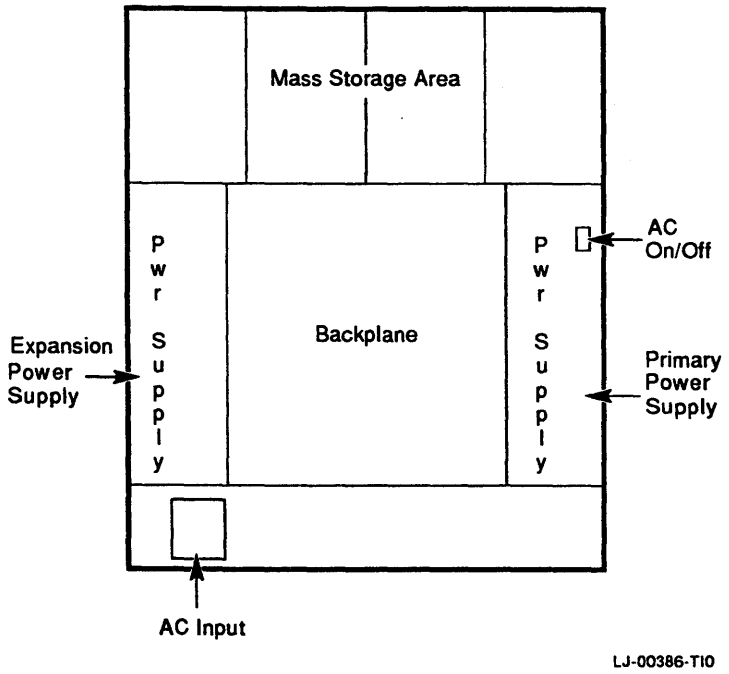


Figure 5–14 BA213 Enclosure Front View

5.7.1.1 Mass Storage

The 70-24227 has a mass storage area with four mounting slots. Each slot has individual shock mounting so that each mass storage can be removed or installed separately. These slots will be configured according to what disk controller is used in the system.

A mass storage distribution board is included with the hardware and can be accessed by removing two finger screws at the top of the bracket that holds the board in place. This configuration is used in conjunction with a TK type tape controller, and will house three RD disks and one TK tape drive.

If the KDA50-SA is used, the mass storage area will be configured for SDI (RAxx) type disk drives. Included with the hardware package are a distribution panel that interfaces the KDA50 with external RA drives. It provides the means for dual-porting the internal RA drives, and a "front panel" module that provides the necessary switches and indicator lights for the internal drives. The distribution panel occupies the left most slot, leaving two slots for the disk drives. This configuration is used in conjunction with a TK type tape controller, and houses one TK tape drive.

The I/O module for the KN220 will support DSSI (RFxx) type disk/tape drives. The KN220 I/O module contains an adapter that supports the DSSI architecture. Each mass storage device (RFxx) is "daisy-chained" by means of a 50 pin cable. This configuration will support a maximum of six full-height (or half-height) RF ISE's, for a maximum of seven logical devices. The I/O module for the KN220 will also support SCSI (RZxx) type disk/tape drives. The KN220 I/O module contains an adapter that supports the SCSI architecture. Each mass storage device (RZxx) is "daisy-chained" by means of a 50 pin cable. This configuration will support a maximum of six full-height (or half-height) RZ disk drives, and one TZ tape drive for a maximum of seven logical devices in an external R400X.

NOTE
The DSSI devices use a five pin power connector. The fifth pin is for the ACOK signal. This signal is used during spin-up for drive sequencing. It is also used to warn the drive's on-board controller that the DC voltages will soon be gone.

The following are the maximum formatted internal disk capacities:

3 RF71 drives	1200 Mbytes

RA, and RF type drives cannot be mixed in a single enclosure. This is due to the unique operator control panels, required by the different disk technologies.

Internal RA70 drives can be dual ported between two systems. However, RF type drives cannot be dual ported.

External RA-type drives are supported on a single KDA50 controller, external RF-type drives can be added off the DSSI external port.

Figure 5–15 shows the supported configurations for an RA70,RF71 or TK70.

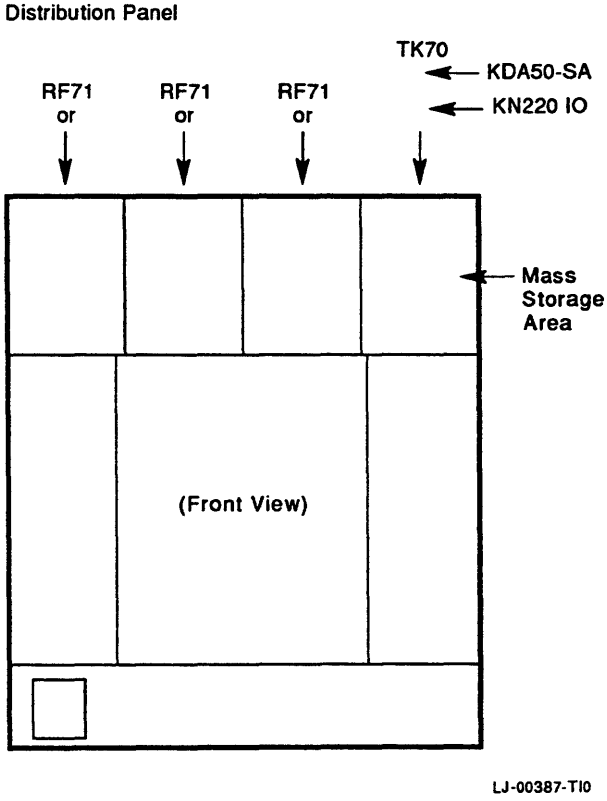


Figure 5–15 BA213 Enclosure Mass Storage Configuration

5.7.1.2 Backplane Specifications

The backplane (70-23712-01) is a 11.9" x 16" assembly consisting of a single layer double sided etch board with 1" center-to-center connections, for up to 12 Q22-bus quad height modules. The backplane is bounded and therefore has no expansion capabilities.

There are two power connectors for the backplane; each is a dual 28 (56 pin total) pin edge board connector. The connector near slot 1 is called the primary power supply connector and distributes +5 volts and +12 volts to slots 1 through 6. The connector near slot 12 is called the expansion power supply connector and distributes +5 volts and +12 volts to slots 7 through 12. Ground is common throughout the backplane.

Number of Q/CD Slots	12 quad
Number of Q/Q Slots	0
Total Slots	12 quad
Available AC Loads[1]	40
Available DC Loads	20
Termination	120 ohms

[1]Available AC loads is based on 240 ohm termination on single KN220 CPU with 120 ohm termination on backplane.

Figure 5–16 shows the general outline, the priorities, and the power supply connectors of the backplane.

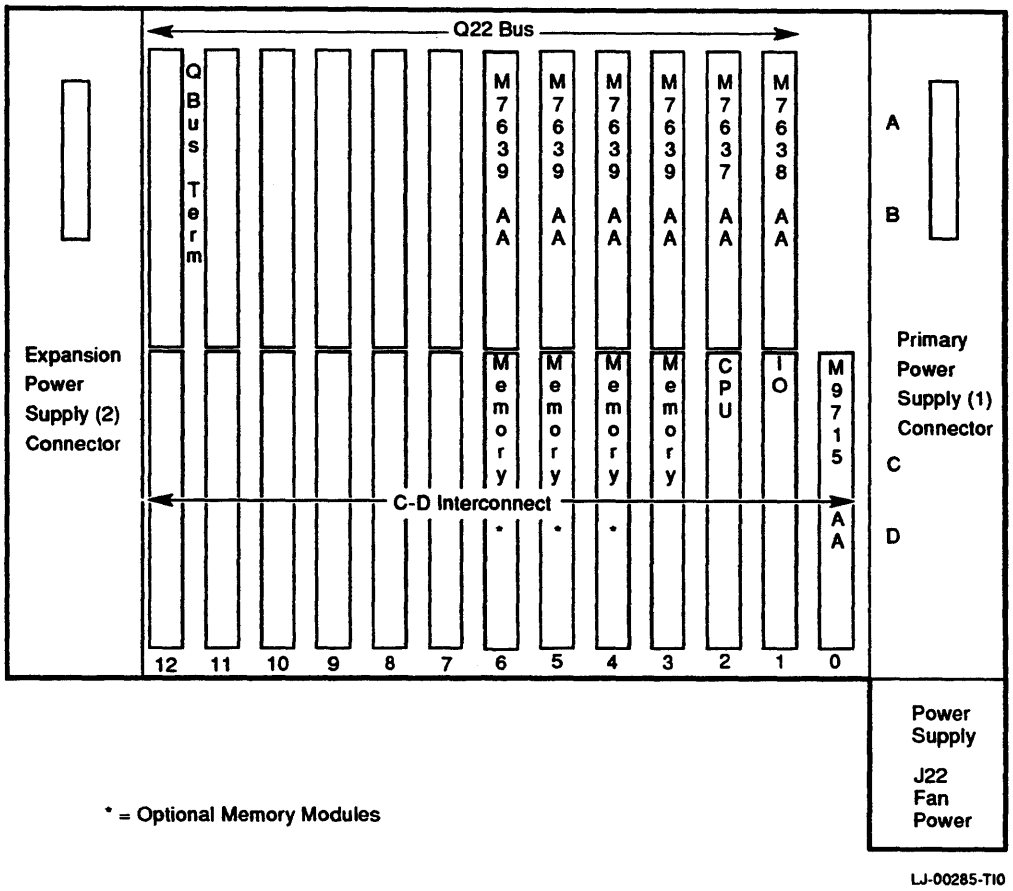


Figure 5–16 BA213 Backplane Layout

5.7.1.3 Power Supply Specifications

Both variations of the 70-24227 have two (H7868-A/B) power supplies. Each supply is capable of delivering 230 watts split between +5 and +12 volts. The primary supply, located on the right hand side when facing the front of the system, will power both the DC fans. The speed of these fans varies as the ambient temperature varies.

Power for mass storage devices is provided via a 9 pin MTA connector on the front of the power supply. Each supply powers two of the four mass storage slots.

On the front cover of the power supply is a green LED. This LED is lit when the DC voltages are within regulation. On the front cover there is also a CPU restart switch and a power controller output.

NOTE

The H7868 power supply is not switchable, and two separate versions are required for 120V and 240V operation.

The 70-24227-03 will use the H7868-A 120V power supply.

The 70-24227-04 will use the H7868-B 240V power supply.

5.7.1.4 DC Power Supply Specification

Conditions	5.1V Nominal Output	+12.1V Nominal Output	Fan B Output
current - min. load	5.0 amps[1]	0.0 amps	0.0A
current - max. load	33.0 amps	7.0 amps	0.70A @ -12.8V, 0.43A @ - 7. 8V
ripple & noise	50mv p-p max	75mv p-p max	150mv p-p max
total regulation	13.0 %	13.0%	15.0%
min.	+4.95V	+11.74V	-7.8V @ < 30C
typ.	+5.10V	+12.10V	
max.	+5.25V	+12.46V	-12.8V @ > 42C nominal voltage varies with temperature
long term stability	0.4%/1000 hrs	0.4%/1000 hrs	n/a
over current	34 amps min	7.2 amps min	1.5 amps min
over current	40 amps max	8.3 amps max	3.0 amps max
start up overcurrent delay		30 sec. min.	
must carry	n/a	11.5A	n/a
must trip		14.0A	

[1]required by each power supply - M9060-YA load module can used to supply 5A minimum.

Conditions	5.1V Nominal Output	+12.1V Nominal Output	Fan B Output
short circuit current	0.1A max into a 0.1 ohm load	0.1A max into a 0.1 ohm load	1.0A max into a 0.1 ohm load
over voltage	6.5v max OVP turn-on limit 5.5V max	14.2v max OVP turn-on limit 13.2V max	no protection
dynamic load regulation	100mv PK 2ms max.	a)240mv PK b)120mv PK 2ms max	600mv PK 2ms max
DI/DT load steps	0.06A/5S 29A to 33A (675S)	a) 0.01A/5S, 3.5A to 7.0A b) 0.04/5S 1.5A to 3.5A	0.01A/5sec 0.0A to 0.8A (8005S)

5.7.1.5 AC Input Specification

PARAMETER	MIN	TYP	MAX	UNITS	SYMBOL
Voltage Nominal	–	101	–	volts	V
Voltage Design Range	88	–	110	volts	V
RMS Current @Nominal (steady state)	1.2	–	10.2	amperes	A
Voltage Nominal	–	120	–	volts	V
Voltage Design Range	104	–	132	volts	V
RMS Current @Nominal (steady state)	1.0	–	8.6	amperes	A
Voltage Nominal	–	220 /240	–	volts	V
Voltage Design Range	176	–	264	volts	V
RMS Current @Nominal (steady state)	.54	–	4.7	amperes	A
Frequency Nominal	–	50/60	–	hertz	Hz
Frequency Range	47	–	63	hertz	Hz
Number of Phases	–	1	–	none	NA
RMS ground current	–	–	.25	millamperes	ma
Peak Current (Steady State) 101 V	–	16.0	–	amperes	A
Peak Current (Steady State) 120 V	–	13.7	–	amperes	A
Peak Current (Steady State) 220 /240 V	–	8.0	–	amperes	A
Power Cord Type		IEC 320 C16 (10A @ 250 V or 15 A @ 125 V)			
Power Cord Length	–	1.9	–	meters	m
Power Cord Length	–	75	–	inches	in
AC Plug Type		country specific, refer to A-SP-EL00002-TB-0			
Ride-through Time	20	–	–	millisecond	ms
Inrush Current	–	–	100	amperes peak	A
Start-up Current Amplitude 101 V	–	–	13.5	rms amperes	A

PARAMETER	MIN	TYP	MAX	UNITS	SYMBOL
Start-up Current Amplitude 120 V	–	–	11.5	rms amperes	A
Start-up Current Amplitude 220 /240 V	–	–	6.3	rms amperes	A
Start-up Current Duration	–	–	30	seconds	s
Power Consumption	150	–	670	watts	W
Apparent Power	250	–	1030	volt amperes	VA
DC Output Watts Available	–	–	460	watts	W
Circuit Breaker Rating per pwr supply	–	7.5	–	amperes	A
Power Factor	–	.65	–	none	PF
Current Crest Factor	–	3	–	none	CF
Voltage Crest Factor	1.37	–	–	none	CF

5.7.1.5.1 Airflow Specifications

The 70-24227 is equipped with two 4 $\frac{1}{2}$ inch DC fans. The speed of these fans is determined by a temperature sensor in the power supply. This sensor measures ambient air temperature. The temperature vs airflow transfer characteristics are shown in the table below.

Temperature	Fan Voltage	Airflow Per Slot	Airflow Per Power Supply	Airflow Per Drive
above				
41°C + /-3°C	12.7V	8.5 CFM	25 CFM	38 CFM
		200 LFM		
less than				
>23°C 13°C	7.8V	5.7 CFM	16.5 CFM	25 CFM
		135 LFM		

NOTE

These values are approximate design numbers. Actual thermocouple measurements of devices should be used to verify cooling efficiency.

5.7.1.5.2 Physical Specifications

	Rack Mount	Pedestal	Units	Symbol
Height	.62	.69	meters	m
	24.4	27.0	inches	in
Width	.44	.53	meters	m
	17.5	21.0	inches	in
Depth	.29	.45	meters	m
	11.5	17.8	inches	in
Weight	26.3	37.2	kilograms	kg
	58	82	pounds	lb
	Minimum Service Clearance Required			
front	1	1	meters	m
front	39.4	39.4	inches	in
rear[1]	0	0	meters	m
rear	0	0	inches	in
left side	0	0	meters	m
left side	0	0	inches	in
right side	0	0	meters	m
right side	0	0	inches	in
	Minimum Clearance Required for Airflow			
front	5.1	5.1	centimeters	cm
front	2.0	2.0	inches	in
rear	5.1	5.1	centimeters	cm
rear	2.0	2.0	inches	in
left side	5.1	5.1	centimeters	cm
left side	2.0	2.0	inches	in
right side	5.1	5.1	centimeters	cm
right side	2.0	2.0	inches	in

[1]One Meter (39.4 inches) for backplane service only

5.7.1.5.3 Environmental Specifications

PARAMETER	MIN	TYP	MAX	UNITS	SYMBOL
Airflow Intake and Exhaust Location	Intake Location: Left side, right side, front and rear all located at the upper part of the box.		Exhaust Location: Right side, rear, bottom and front all located at the lower part of the box.		
Operating Altitude	–	–	2438	meters	m
Nonoperating Altitude	–	–	8000	feet	ft
Nonoperating Altitude	–	–	4877	meters	m
Operating Mechanical Shock	Duration		16000 /10	feet /milliseconds	ft/ms
Operating Mechanical Shock	Level		10	gravities	g
Operating Vibration Freq. Range	22		500	Hertz	Hz
Operating Vibration Vibra.	0.01"da		.25g		
Floor Mount with Skins Freq. Range	5		22	Hertz	Hz
Class A/B Vibra.	.01"da	–	.25g		
Operating Vibration Freq. Range	30		500	Hertz	Hz
Operating Vibration Vibra.	0.01"da		.50g		
Class C Vibra.	.01"da		.50g		
Nonoperating Mechanical Shock	Duration		30	milliseconds	ms
Nonoperating Mechanical Shock	Level		40	gravities	g
Nonoperating Vibration Freq. Range					
Nonoperating Vibration Vibra. Level					
Acoustics - Operating LNPE	–	5.7	–	Bels	B
Acoustics - Operating LNA	–	42	–	Decibels	dBA
Acoustics - Idle LNPE	–	5.7	–	Bels	B
Acoustics - Idle LNA	–	42	–	Decibels	dBA

5.8 KN220 Processor Specifications

The KN220-AA is a Q22-bus compatible R3000 CPU, based on the R3000 (MIPSCO) chip set for use in the BA200 series or BA400 enclosures. The KN220-AA (M7637-AA) is a multi-user CPU.

CENTRAL PROCESSOR		
Clock Rate:	30 megahertz	
Data Path Width:	32 bits	
Number of Instructions:	74	
Number of Data Types:	1	
General Purpose Registers:	32 (32-bit wide)	
Addressing Modes: (3 total)	I-Type (Immediate)	
	J-Type (Jump)	
	R-Type (Register)	
Time Bases: 100Hz		
	Interval Timer:	1 (10ms)
	Programmable Timers:	2
I/O Bus Interface:	One Q22-Bus interface with 8192 entry "scatter-gather" map ,SCSI and DSSI.	
Backplane Termination:	240 ohms	

MEMORY MANAGEMENT AND CONTROL: Demand paged virtual memory with two-level page tables.

Page Size:	512 Kbytes	
Virtual Address Space:	4 Gbytes	
Physical Memory Space:	512 Mbytes	
Number of Memory Modules:	4 maximum	

PERFORMANCE		
Data Cache and Address Cache:		
	Size (each):	64 kbyte
	Speed:	12ns
	Direct Mapped	
Translation Buffer:		
	Size:	64 entry
	Associativity:	Fully associative
Q-22 Bus Address Translation Map Cache:		
	Size:	16 entry
	Associativity:	Fully associative

Q-22 Bus Buffer Size:		
	Input:	32 bytes
	Output:	4 bytes
Maximum Q-22 Bandwidth:		
	DMA Read:	2.4 Mbytes/sec
	DMA Write:	3.3 Mbytes/sec

ETHERNET PORT (SGEC)

Supported Protocols	Ethernet V2.0
Supported media types	ThickWire Ethernet or ThinWire Ethernet
DMA Transfer	
Data Path Width	32 bits

DIGITAL SMALL STORAGE INTERCONNECT (DSSI) PORT

Maximum Number of Supported Devices	7 Devices
Data Path Width	8 bits
Maximum Bandwidth	4 Mbytes/sec
Buffer Size	128 Kbytes

NOTE
The DSSI bus will support 8 devices, however the DSSI port on the KN220-AA counts as one device. Therefore the number of additional mass storage devices which may be connected to this port is 7.

SMALL COMPUTER STORAGE INTERCONNECT (SCSI) PORT

Maximum Number of Supported Devices	7 Devices
Data Path Width	8 bits
Maximum Bandwidth	4 Mbytes/sec
Buffer Size	128 Kbytes

NOTE
The SCSI bus will support 8 devices, however the SCSI port on the KN220-AA counts as one device. Therefore the number of additional mass storage devices which may be connected to this port is 7.

CONSOLE SERIAL LINE

Interface Standards:	EIA RS-423A/CCITT V.10 X.26, EIA RS-232C/CCITT V.28, DEC 423
Data Format:	1 start bit, 8 data bits, 0 parity bits, 1 stop bit
Baud Rates:	300, 600, 1200, 2400, 4800, 9600, 19200, 38400

H3602-AC COVER PANEL

Switches	2-Position Switch for Break Enable/Disable Function
	2-Position Switch for ThinWire/ThickWire Ethernet selection (Ethernet Connector Switch)
	3-Position Switch for CPU Power-Up Mode Selection
	8-Position Switch for Console SLU BAUD Rate Selection
Display	Hexadecimal Display for Console Program Status 2 Green LEDs to indicate which Ethernet connector is selected
Battery for Time of Year Clock	3.75V (nominal) rechargeable NiCad - Pt.#(12-19245-01)
Connections	Modified Modular Jack for Console Terminal
	50 pin cable for console, configuration info, battery backup and hexadecimal display.
	15 pin D-sub Ethernet transceiver cable connector
	ThinWire Ethernet BNC connector

H3605-AA COVER PANEL

	two 50 pin D-sub SCSI cable connectors

NOTE

Placing the Power Up Mode in the Test Position does not wrap the serial line output back to the line input as was done on the distribution panel for the BA123 boxes. When running SLU Test Mode with the H3602-AC, one must insert an external wrap-around plug.

ORDERING INFORMATION

Module Only :	M7637-AA
Module Cover:	H3602-AC
Module Cover:	H3605-AA
Console Cables:	3.0 meters (10 feet):
	7.6 meters (25 feet):
	15.2 meters (50 feet):
Passive Adapters:9-pin:	H8571-B
25-pin:	H8571-A

OPERATING SYSTEM SUPPORT

ULTRIX 32:	4.0 and later

DIAGNOSTIC SUPPORT

MicroVAX Maintenance System:	X.X and later
Loopback capability:	See Note Above
Loopback connector: (MMP)	H3103
Loopback connector: (MMJ)	H3103 + H8572
Self test (KN220-AA/MSXXX):	Yes

CONFIGURATION INFORMATION

Module Form Factor:	Quad Height
Distribution Insert Type:	Cover panel
DC Amps Drawn @ +5V:	TBD
DC Amps Drawn @ +12V:	0.14A
AC Bus Loads:	3.5
DC Bus Loads:	1.0

MODULE CONNECTORS

Number of positions	Gender	Type	Location	Destination
100 pin	male	3M	CPU&I/O A/B	RIO Bus
100 pin	male	3M	CPU C/D	MSXXX memory
50 pin	male	3M	I/O C/D	DSSI Internal
100 pin	male	3M	B	I/O

INTERRUPT CHARACTERISTICS

Number of Q-Bus Interrupt Levels:	6 hardware and 2 software
Supported Q-Bus Interrupt Levels:	BR 4-7

DMA CHARACTERISTICS

Q-Bus Burst Mode DMA Support:	Yes
Q-Bus Block Mode DMA Support:	Up to 16 words

5.9 MS220-AA Memory Module Specifications

5.9.1 MS220-AA

The MS220-AA (M7639-AA) is a 32 Mbytes memory module. It has a 100ns, 74 bit wide array (64-bit data and 14-bit ECC) implemented with 4 Mbytes dynamic RAMS in dual in-line packages (SOIC).

SUPPORTED ACCESS TYPES

Synchronous:	Masked and unmasked longword, quadword
Asynchronous(DMA):	Masked and unmasked longword, quadword, hexword, octaword.

ORDERING INFORMATION

Module only:	M7639-AA
Single-width Blank Cover:	70-23981-01
Dual-width Blank Cover:	70-23982-02
Side Wall Gap Filler:	74-34042-01
2-Tap CPU/Memory Cable:	17-02700-01
3-Tap CPU/Memory Cable:	17-02700-01
4-Tap CPU/Memory Cable:	17-02700-02
5-Tap CPU/Memory Cable:	17-02700-03

PERFORMANCE

	R3000	CVAX
Sync (VAX)Longword Read:	333ns	400-500ns
Synch Unmasked (VAX)Longword Write:	333ns	400ns
Sync Masked (VAX)Longword Write:	600ns	700ns
4 Word Block Read:	500ns	600ns

OPERATING SYSTEM SUPPORT

ULTRIX 32:	4.0 and later
ULTRIX 32 NFS:	4.0

DIAGNOSTIC SUPPORT

MicroVAX Maintenance System:	132 and later
Self test:	Tested by KN220-AA self-test

CONFIGURATION INFORMATION

Module Form Factor:	Quad Height
Distribution Insert Type:	Blank cover
DC Amps Drawn @ +12V:	0.0A
AC Bus Loads:	0
DC Bus Loads:	0

MODULE CONNECTORS

Number of positions	Gender	Type	Location	Destination
100 pin	male	3M	C/D	CPU

Memory address is automatically configured based on backplane placement.

RELIABILITY SPECIFICATIONS

MTBF:(current est.)	306k hours

ENVIRONMENTAL CONSTRAINTS

Storage Temperature:	-40°C to 66°C -40°F to 151°F
Operating Temperature:	5°C to 60°C 41°F to 140°F
Relative Humidity:(Noncondensing)	10% to 95%

5.10 Disk Drive Specifications

5.10.1 RF71-A Fixed Disk Drive Specifications

The RF71-A is a DSSI compatible random access, moving head, fixed media disk drive using 5 ¼ inch disks, with a formatted capacity of 400 Mbytes.

STORAGE CAPACITY	
User capacity:(bytes)	400,097,280
User capacity (blocks)	781,440
Recording density (inner track)	22,775 bpi
DRIVE ORGANIZATION	
Bytes Per Sector	512
Sectors Per Track:(usable)	37
Tracks Per Cylinder	16
Number of Cylinders (usable)	1320
PERFORMANCE	
Average Seek Time to data	35.0ms
Cylinder to Cylinder Seek	6.00ms
Maximum Seek	41ms
Average Rotational Latency	8.33ms
Head Switch Latency	8.00ms
Peak Transfer Rate (to DSSI bus)	1.5 Mbytes/sec. 12.0 Mbits/sec.
Serve Information	Embedded serve
Rotational Speed	3600 RPM
Start Time (availability to user)	45 sec.
Stop Time:	15 sec.
ORDERING INFORMATION	
Drive Only	RF71-A
PHYSICAL SPECIFICATIONS	
Width	14.60 cm(5.75 in)
Depth (incl. bezel)	20.75 cm (8.17 in)
Height (incl. shoe plate)	7.75 cm (3.05 in)
Weight:(approximate)	4.09 kg (9.0 lbs.)

DC ELECTRICAL SPECIFICATIONS (continuous seek mode-1/3 stroke seek with 50% duty cycle)

Current		
	+ 5V Current(idle/avg.)	1.1 A typ./ 1.25 A max.
	+12V Current(idle/avg.)	.75 A typ./.91 A max.
	+ 5V Current(continuous seek)	1.1 A typ./1.25 A max.
	+12V Current(continuous seek/peak)	2.57 A typ./2.96 A max.
	+12V Current(continuous seek/avg.)	1.3 A typ./1.57 A max.
	+ 5V Current(spin-up/peak)	1.1 A typ./1.25 A max.
	+12V Current(spin-up/peak)	4.17 A typ./4.54 A max.
Power		
	Total average power	21.1 W typ./26.5 W max.
	Total peak power	36.3 W typ./44.0 W max.

RELIABILITY	
Hard Error Rate	Not to exceed 1 error in 10E12 bits read.
Soft Error Rate	Not to exceed 1 error in 10E8 bits read.
MTBF	50,000 hr.

ENVIRONMENTAL CONSTRAINTS	
Storage Temperature	-40°C to 66°C -40F to 151F
Operating Temperature	10°C to 50°C 50°F to 122°F
Relative Humidity (noncondensing)	10% to 80%
Average Acoustic Noise (ISO 7779 LNPE) 5.7 Bels (seeking))	4.6 Bels (idle
Maximum Operating Altitude	8,000ft.
Mechanical Shock (operating)	10g PC, @7-13 ms duration
Vibration (operating)	5-22 Hz 0.010 in. DA
	22-500 Hz 0.25 G peak
	500-22 Hz 0.25 G peak
	22-5 Hz 0.010 in. DA
Heat Dissipation:	<21.1 typ /26.6 max.Watts

5.10.2 RZ56-FA Fixed Disk Drive Specifications

The RZ56-FA is a SCSI compatible random access, moving head, fixed media disk drive using 5 ¼ inch disks, with a formatted capacity of 665 Mbytes.

STORAGE CAPACITY	
User capacity:(bytes)	665.17 Mbytes
User capacity (blocks)	1,299,177

Recording density (at ID)	31,846 bpi
DRIVE ORGANIZATION	
Bytes Per Block	512
Blocks Per Track:(usable)	54
Blocks Per Drive	1,299,174
Number of Cylinders (spare)	3
Number of Blocks per Cylinder(spare)	11
PERFORMANCE	
Average Seek Time:(including settle)	less than 16ms
Track to Track Seek	less than 4.0ms
Maximum full stroke Seek	35ms
Average Rotational Latency	8.3ms
Peak Transfer Rate (to/from media)	1.85 Mbytes/sec.
Serve Information	Embedded serve
Rotational Speed	3600 RPM
Start Time (availability to user)	20 sec.
Stop Time:	20 sec.
ORDERING INFORMATION	
Drive Only	RZ56-FA
Expansion Box	RZ5X-DA
50 Pin Terminator	12-30552-01
Connector Cable 50 to 50 pin 72 inches	BCXXX-XX
Connector Cable 50 to 50 pin 96 inches	BCXXX-XX
PHYSICAL SPECIFICATIONS	
Width	14.60 cm(5.75 in)
Depth	20.77 cm(8.18 in)
Height	8.26 cm(3.25 in)
Weight (approximate)	3.1 kg(8.4 lbs.)

Current		
	+ 5V Current(idle/avg.)	1.48 A typ.
	+12V Current(idle/avg.)	1.7 A typ.
	+ 5V Current(random seek)	1.3 A typ./1.43 A max.
	+12V Current(random seek /peak)	2.0 A typ./3.3 A max.
	+ 5V Current(spin-up/peak)	/1.48 A max.
	+12V Current(spin-up/peak)	/4.2 A max.
Power		
	Total average power	30 W max.
	Total peak power	60.7 W max.

RELIABILITY

Hard Error Rate	Not to exceed 10 error in 10E14 bits read.
Soft Error Rate	Not to exceed 10 error in 10E11 bits read.
MTBF	60,000 hr.

ENVIRONMENTAL CONSTRAINTS

Storage Temperature	-40°C to 66°C -40F to 151F
Operating Temperature	10°C to 55°C 50°F to 128°F
Relative Humidity (noncondensing)	10% to 80%
Average Acoustic Noise (ISO 7779 LNPE)	5.0 Bels (avg/ 5.7 Bels (seeking)
Maximum Operating Altitude	8,000ft.
Mechanical Shock (operating)	3g PC, @10=/–3 ms duration
Heat Dissipation:	<32 Watts

5.11 Tape Drive Specifications

5.11.1 TK70-A Tape Drive Specifications

The TK70-A is a complete tape subsystem consisting of a TQK70-SA (M7546-00) controller module and a TK70-A ½ inch cartridge streaming tape drive made to fit in the 5 ¼ inch mass storage footprint. It has a maximum formatted capacity of 296 Mbytes and uses TK50 type½ inch tape cartridges.

STORAGE CAPACITY		
Formatted:(optimum)	296 Mbytes	
FUNCTIONAL SPECIFICATIONS		
Recording Media:	Magnetic Tape	
Tape Length:	182.9 m (600 ft.)	
Tape Width:	1.27 cm (0.5 in)	
Mode of Operation:	Streaming	
Read/Write Method:	Serpentine	
Recording Density:	10,000 bpi	
Number of Tracks:	48	
PERFORMANCE		
Tape Start Time:	325 ms max.	
Tape Stop Time:	200 ms max.	
Tape Speed:	100 in./sec.	
Data Transfer Rate:	125 Kbytes/sec.	
Access Time		
	TK50 Mode:	(read only)35 minutes max.
	TK70 Mode:	60 minutes max.
PHYSICAL SPECIFICATIONS		
Height	8.25 cm (3.25 in.)	
Width	4.60 cm (5.70 in.)	
Depth	1.44 cm (8.44 in.)	
Weight (w/o cartridge)	2.27 Kg (5.0 lbs.)	
DC POWER SPECIFICATIONS		
+ 5V Current:(typical)	1.3 amps	
+12V Current:(typical)	2.0 amps	
+ 5V Current:(maximum)	1.5 amps	
+12V Current:(maximum)	2.4 amps	

+ 5V Tolerance:(incl. ripple)	75 mv p.p.
+12V Tolerance:(incl. ripple)	75 mv p.p.
Power Consumption:(maximum)	36 watts

RELIABILITY SPECIFICATIONS

MTBF Information:(drive and controller)	15,000 hours
Soft Error Rate:	TBD

ENVIRONMENTAL CONSTRAINTS

Storage Temperature:	-40°C to 60°C
	-40°F to 140°F
Operating Temperature:	10°C to 40°C
	50°F to 104°F
Relative Humidity:(noncondensing)	20% to 80%
Average Acoustic Noise:(ISO 7779 LNPE)	2.9 Bels (idle)
	4.7 Bels (streaming)
Average Acoustic Noise:(ISO 7779 LPA)	29 db (idle)
	35 db (streaming)
Maximum Operating Altitude:	8,000ft.
Mechanical Shock (operating):	10g PC, 10ms duration

5.12 Mass Storage Controller Specifications

5.12.1 TQK70-SA Tape Drive Controller

The TQK70-SA Tape Controller (M7546-00) is a Q22-Bus interface used to control the TK70-A tape drive. Note, if TQK70 is used together with a KLESI in same system, the TQK70 must be configured as the first TMSCP device.

FUNCTIONAL SPECIFICATIONS	
Multiporting:	No
TMSCP:	Yes
Start/Stop:	No
Controllers Per System:	1 maximum
Drives per controller:	1 maximum
PERFORMANCE	
Throughput Rate:	80 kilobytes/sec
Buffer Size:	64 kilobytes
ORDERING INFORMATION	
Tape Drive:	TK70-A
Controller Module:	M7546
Cable:	17-01363-01
OPERATING SYSTEM SUPPORT	
ULTRIX 32:	4.0a and later
DIAGNOSTIC SUPPORT	
MicroVAX Maintenance System:	2.11 and later
Self test:	Yes
CONFIGURATION INFORMATION	
Module Form Factor:	Dual Height
Distribution Insert Type:	N/A
DC Amps Drawn @ +5V:	3.5A
DC Amps Drawn @ +12V:	None
AC Bus Loads:	4.3
DC Bus Loads:	0.5

MODULE CONNECTORS

Number of positions	Gender	Type	Location	Destination
26	MALE	3M	B	TK70

INTERRUPT CHARACTERISTICS

Interrupt Level(s): BR 4

DMA CHARACTERISTICS

Type: Block mode

Holdoff Timer: None

Q-BUS ADDRESS

Fixed Address: 1774500

Floating Addresses: 17760404

17760444

Q-BUS VECTOR

Configuration: Programmable

Fixed Vector: 260

RELIABILITY SPECIFICATIONS

MTBF Information: 37,000 hours

5.13 Communication Device Specifications

5.13.1 Asynchronous Communication Devices

5.13.1.1 CXA16-AA

The CXA16 (M3118-YA) is an asynchronous multiplexer, which provides sixteen full duplex serial data channels for use in the BA430 box. The device can operate in either DHV11 or DHU11 mode depending on the setting of an on-board switch. However, the CXA16 is a data lines only device and as such does not support any devices using modem control signals.

FUNCTIONAL INFORMATION

Supported Line Interfaces:	EIA 232-D /CCITT V.24, V.28
	EIA RS423A /CCITT V.10 X.26
	DEC423
Split Speed Operation:	All lines
Flow Control:(XON/XOFF)	All lines
Supported Data Formats:	16 programmable formats
	each with 1 start bit

- 5,6,7 or 8 data bits, 0 or 1 parity bits, and 1 stop bit.
- 5 data bits, 0 or 1 parity bits and 1.5 stop bits.
- 6,7 or 8 data bits, 0 or 1 parity bits and 2 stop bits.
- Parity, if enabled, can be either odd or even.

Modem Control:	None

ORDERING INFORMATION

Module only:	CXA16-AA
Module cables, 7.6 meters (25 feet):	BC16D-25
Module cables, 15.2 meters (50 feet):	BC16D-50
Cable Concentrator:	H3104
Office cables, 3.0 meters (10 feet):	BC16E-10
Office cables, 7.6 meters (25 feet):	BC16E-25
Office cables, 15.2 meters (50 feet):	BC16E-50
Passive adapters, 9-pin:	H8571-B
Passive adapters, 25-pin:	H8571-A
Active Adaptor:	H3105

Cable Extender:	H8572
OPERATING SYSTEM SUPPORT	
ULTRIX 32:	4.0 and later
DIAGNOSTIC SUPPORT	
MicroVAX Maintenance System:	132 and later
Loopback connector:(MMP)	H3103
Loopback connector:(MMJ)	H3103 + H8572
Self test go/no-go:	1 green LED

PERFORMANCE

- Transmit Data Transfers:
  - — Single character programmed transfers or up to 16- character block mode DMA transfers in DHV11 mode
  - — Single character or two-character programmed transfers or up to 16-character block mode DMA transfer in DHU11 mode
- Receive Data Transfers:
  - — Single character programmed transfers in both DHV11 and DHU11 modes
- Transmit Buffer Size:
  - — One character for programmed transfers in DHV11 mode
  - — 64-character FIFO for programmed transfers in DHU11 mode
  - — 64-character FIFO for DMA Transfers in DHV11 and DHU11 modes
- Receive Buffer Size:
  - — 256-character FIFO in DHV11 and DHU11 modes
- Supported Baud Rates:
  - — 16 programmable baud rates: 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800, 7200, 9600, 19200, 38400
- Throughput At Maximum Baud Rate:
  - — 5 data bits, 0 parity, 1 stop bit: 140,000 char/sec
  - — 7 data bits, 1 parity, 1 stop bit: 110,000 char/sec

CONFIGURATION INFORMATION

Module Form Factor:	Quad Height
Distribution Insert Type:	N/A

DC Amps Drawn @ +5V:	1.40
DC Amps Drawn @ +12V:	0.14
AC Bus Loads:	3.0
DC Bus Loads:	1.5

MODULE CONNECTORS

Number of positions	Gender	Type	Location	Destination
36	Female	Amphenol	A/B section	H3104 Cable Concentrator
36	Female	Amphenol	C/D section	H3104 Cable Concentrator

INTERRUPT CHARACTERISTICS	
Interrupt Level(s):	BR 4
DMA CHARACTERISTICS	
Type:	Block mode
Holdoff Timer:	None
Maximum Transfer Size:	16 Characters
Q-BUS ADDRESS	
Configuration:	Via switches
Fixed Address:	None
Floating Addresses, Rank:(decimal)	32
Floating Addresses, Size:(decimal)	8
Floating Addresses, Modulus:(octal)	20
Q-BUS VECTOR	
Configuration:	Switches
Floating Addresses, Rank:(decimal)	57
Floating Addresses, Size:(decimal)	4
Floating Addresses, Modulus:(octal)	10
RELIABILITY SPECIFICATIONS	
MTBF Information:	60,000 hours

5.13.1.2 CXB16-AA

The CXB16 (M3118-YB) is an asynchronous multiplexer, which provides sixteen full duplex serial data channels for use in the BA430 enclosure. The device can operate in either DHV11 or DHU11 mode depending on the setting of an on-board switch. However, the CXB16 is a data lines only device and as such does not support any devices using modem control signals.

FUNCTIONAL INFORMATION

Supported Line Interfaces:	EIA RS422-A/CCITT V.11X.27
Split Speed Operation:	All lines
Flow Control:(XON/XOFF)	All lines
Supported Data Formats:	16 programmable formats
	each with 1 start bit

- 5,6,7 or 8 data bits, 0 or 1 parity bits, and 1 stop bit.
- 5 data bits, 0 or 1 parity bits and 1.5 stop bits.
- 6,7 or 8 data bits, 0 or 1 parity bits and 2 stop bits.
- Parity, if enabled, can be either odd or even.

Modem Control: None

ORDERING INFORMATION

Module only:	CXB16-AA
Module cables, 7.6 meters (25 feet):	BC16D-25
Module cables, 15.2 meters (50 feet):	BC16D-50
Cable Concentrator:	H3104
Office cables, 3.0 meters (10 feet):	BC16E-10
Office cables, 7.6 meters (25 feet):	BC16E-25
Office cables, 15.2 meters (50 feet):	BC16E-50
Passive Adapters, 9-pin:	H8571-B
Passive Adapters, 25-pin:	H8571-A
Active Adaptor:	H3105
Cable Extender:	H8572

OPERATING SYSTEM SUPPORT

ULTRIX 32:	4.0 and later

DIAGNOSTIC SUPPORT

MicroVAX Maintenance System:	131 and later
Loopback connector:(MMP)	H3103
Loopback connector:(MMJ)	H3103 + H8572
Self test go/no-go:	1 green LED

PERFORMANCE

- Transmit Data Transfers:
  - — Single character programmed transfers or up to 16- character block mode DMA transfers in DHV11 mode
  - — Single character or two-character programmed transfers or up to 16-character block mode DMA transfer in DHU11 mode
- Receive Data Transfers:
  - — Single character programmed transfers in both DHV11 and DHU11 modes
- Transmit Buffer Size:
  - — One character for programmed transfers in DHV11 mode
  - — 64-character FIFO for programmed transfers in DHU11 mode
  - — 64-character FIFO for DMA Transfers in DHV11 and DHU11 modes
- Receive Buffer Size:
  - — 256-character FIFO in DHV11 and DHU11 modes
- Supported Baud Rates:
  - — 16 programmable baud rates: 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800, 7200, 9600, 19200, 38400
- Throughput At Maximum Baud Rate:
  - — 5 data bits, 0 parity, 1 stop bit: 140,000 char/sec
  - — 7 data bits, 1 parity, 1 stop bit: 110,000 char/sec

CONFIGURATION INFORMATION

Module Form Factor:	Quad Height
Distribution Insert Type:	N/A
DC Amps Drawn @ +5V:	1.40
DC Amps Drawn @ +12V:	0.14
AC Bus Loads:	3.0
DC Bus Loads:	1.5

MODULE CONNECTORS

Number of positions	Gender	Type	Location	Destination
36	Female	Amphenol	A/B section	H3104 Cable Concentrator
36	Female	Amphenol	C/D section	H3104 Cable Concentrator

INTERRUPT CHARACTERISTICS	
Interrupt Level(s):	BR 4
DMA CHARACTERISTICS	
Type:	Block mode
Holdoff Timer:	None
Maximum Transfer Size:	16 Characters
Q-BUS ADDRESS	
Configuration:	Via switches
Fixed Address:	None
Floating Address, Rank:(decimal)	32
Floating Address, Size:(decimal)	8
Floating Address, Modulus:(octal)	20
Q-BUS VECTOR	
Configuration:	Switches
Fixed Vector:	None
Floating Addresses, Rank:(decimal)	57
Floating Addresses, Size:(decimal)	4
Floating Addresses, Modulus:(octal)	10

5.13.1.3 CXY08-AA

The CXY08 (M3119-YA) is an asynchronous multiplexer, which provides eight full duplex serial data channels for use in the BA430 enclosure. The device can operate in either DHV11 or DHU11 mode depending on the setting of an on-board switch. The CXY08 supports limited modem control that is implemented by software in the host.

FUNCTIONAL INFORMATION

Supported Line Interfaces:	EIA 232-D /CCITT V.24, V.28
	EIA RS423A /CCITT V.10 X.26
	DEC423
Split Speed Operation:	All lines
Flow Control:(XON/XOFF)	All lines
Supported Data Formats:	16 programmable formats
	each with 1 start bit

- 5,6,7 or 8 data bits, 0 or 1 parity bits, and 1 stop bit.
- 5 data bits, 0 or 1 parity bits and 1.5 stop bits.
- 6,7 or 8 data bits, 0 or 1 parity bits and 2 stop bits.
- Parity, if enabled, can be either odd or even.

Modem Control:	Full

ORDERING INFORMATION

Module only:	CXY08-AA
Module cable, 3.6 meters (12 feet):	BC19N-12

NOTE

The BC19N-12 is a 44 conductor cable-assembly . The cable-assembly consists of an 8 foot, 44 conductor cable spliced via an overmolded junction to four 11-conductor cables that are 4 feet in length. The 44 conductor cable is terminated with a 50 pin male CHAMP connector and each 11 conductor cable is terminated with a 25 pin male D-sub connector.

OPERATING SYSTEM SUPPORT

VMS:	4.0 and later
ULTRIX 32:	2.2 and later

DIAGNOSTIC SUPPORT

MicroVAX Maintenance System:	132 and later
Loopback connector:(50 pin male CHAMP)	H3046
Loopback connector:(25 pin female D-sub)	H3197
Self test go/no-go:	1 green LED

PERFORMANCE

- Transmit Data Transfers:
  - — Single character programmed transfers or up to 16- character block mode DMA transfers in DHV11 mode
  - — Single character or two-character programmed transfers or up to 16-character block mode DMA transfer in DHU11 mode
- Receive Data Transfers:
  - — Single character programmed transfers in both DHV11 and DHU11 modes
- Transmit Buffer Size:
  - — One character for programmed transfers in DHV11 mode
  - — 64-character FIFO for programmed transfers in DHU11 mode
  - — 64-character FIFO for DMA Transfers in DHV11 and DHU11 modes
- Receive Buffer Size:
  - — 256-character FIFO in DHV11 and DHU11 modes
- Supported Baud Rates:
  - — 16 programmable baud rates: 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800, 7200, 9600, 19200, 38400
- Throughput At Maximum Baud Rate:
  - — 5 data bits, 0 parity, 1 stop bit: 87,771 char/sec
  - — 7 data bits, 1 parity, 1 stop bit: 61,440 char/sec

CONFIGURATION INFORMATION

Module Form Factor:	Quad Height w/integral recessed cover
Distribution Insert Type:	N/A
DC Amps Drawn @ +5V:	1.30
DC Amps Drawn @ +12V:	.5
AC Bus Loads:	1.5

DC Bus Loads: 1.0

MODULE CONNECTORS

Number of positions	Gender	Type	Location	Destination
50	Female	CHAMP	A/B section	BC19N-12 Cable Assy.
50	Female	CHAMP	C/D section	BC19N-12 Cable Assy.

INTERRUPT CHARACTERISTICS

Interrupt Level(s): BR 4

DMA CHARACTERISTICS

Type: Block mode

Holdoff Timer: None

Maximum Transfer Size: 16 Characters

Q-BUS ADDRESS

Configuration: Via switches

Fixed Address: None

Floating Address, Rank:(decimal) 32

Floating Address, Size:(decimal) 8

Floating Address, Modulus:(octal) 20

Q-BUS VECTOR

Configuration: Switches

Fixed Vector: None

Floating Addresses, Rank:(decimal) 57

Floating Addresses, Size:(decimal) 4

Floating Addresses, Modulus:(octal) 10

RELIABILITY SPECIFICATIONS

MTBF Information: 60,000 hours

NOTE
The cable concentrator assembly (BC19N-12) will operate in a class B environment.

5.14 Line Printer Interface Specifications

The LPV11-SA (M8086-YA) is a dual parallel line printer controller capable of interfacing two LP or LG type line printers to the Q22-bus.

FUNCTIONAL INFORMATION

Supported Character Sets:	Digital Multinational
	OCR-A
	OCR-B
Supported Data Formats:	7 or 8 bit character sets

ORDERING INFORMATION

Module only:	M8086-YA
Interconnect Cable 9.1m(30ft):	BC27L-30

OPERATING SYSTEM SUPPORT

ULTRIX 32:	4.0 and later

DIAGNOSTIC SUPPORT

MicroVAX Maintenance System:	132 and later

PERFORMANCE(with LG01/02)

Upper Case Text Throughput:	600 LPM
Upper/Lower Case Text Throughput:	480 LPM
Width of Printable Area:	13.2in.

CONFIGURATION INFORMATION

Module Form Factor:	Quad Height
Distribution Insert Type:	N/A
Maximum DC Amps Drawn @ +5V:	2.20
Typical DC Amps Drawn @ +5V:	1.60
AC Bus Loads:	1.8
DC Bus Loads:	0.5

MODULE CONNECTORS

Number of positions	Gender	Type	Location	Destination
37	Female	"D"	A/B section	LG or LP type printer
37	Female	"D"	C/D section	LG or LP type printer

INTERRUPT CHARACTERISTICS

Interrupt Level(s): BR 4

Q-BUS ADDRESS

Configuration: Via switches

Q-BUS VECTOR

Configuration: Switches

RELIABILITY SPECIFICATIONS

MTBF Information: 30,000 hours

5.14.1 SCSI Bus Interface

FUNCTIONAL INFORMATION

Small Computer Storage Interconnect Bus.

ORDERING INFORMATION

Module only: Integral to I/O

OPERATING SYSTEM SUPPORT

ULTRIX 32: 4.0

DIAGNOSTIC SUPPORT

MicroVAX Maintenance System: 132 and later

Self test: Yes

MODULE CONNECTORS

Number of positions	Gender	Type	Location	Destination
50	male	3M	C/D	H3605-AA

INTERRUPT CHARACTERISTICS

Interrupt Level(s): BR 4

A
Q22-bus Specification

A.1 Introduction

The Q22-bus, also known as the extended LSI-11 bus, is the low-end member of Digital's bus family. All of Digital's microcomputers, such as the MicroVAX I, MicroVAX II, MicroVAX 3500, MicroVAX 3600, and MicroPDP–11 use the Q22-bus.

The Q22-bus consists of 42 bidirectional and 2 unidirectional signal lines. These form the lines along which the processor, memory, and I/O devices communicate with each other.

Addresses, data, and control information are sent along these signal lines, some of which contain time-multiplexed information. The lines are divided as follows:

- Sixteen multiplexed data/address lines — BDAL<15:00>
- Two multiplexed address/parity lines — BDAL<17:16>
- Four extended address lines — BDAL<21:18>
- Six data transfer control lines — BBS7, BDIN, BDOUT, BRPLY, BSYNC, BWTBT
- Six system control lines — BHALT, BREF, BEVNT, BINIT, BDCOK, BPOK
- Ten interrupt control and direct memory access control lines — BIAKO, BIAKI, BIRQ4, BIRQ5, BIRQ6, BIRQ7, BDMGO, BDMR, BSACK, BDMGI

In addition, a number of power, ground, and space lines are defined for the bus. Refer to Table A–1 for a detailed description of these lines.

The discussion in this appendix applies to the general 22-bit physical address capability. All modules used with the KN220-AA CPU module must use 22-bit addressing.

Most Q22-bus signals are bidirectional and use terminations for a negated (high) signal level. Devices connect to these lines by way of high-impedance bus receivers and open collector drivers. The asserted state is produced when a bus driver asserts the line low.

Although bidirectional lines are electrically bidirectional (any point along the line can be driven or received), certain lines are functionally unidirectional. These lines communicate to or from a bus master (or signal source), but not both. Interrupt acknowledge (BIAK) and direct memory access grant (BDMG) signals are physically unidirectional in a daisy-chain fashion. These signals originate at the processor output signal pins. Each is received on device input pins (BIAKI or BDMGI) and is conditionally retransmitted through device output pins (BIAKO or BDMGO). These signals are received from higher priority devices and are retransmitted to lower priority devices along the bus, establishing the position-dependent priority scheme.

A.1.1 Master/Slave Relationship

Communication between devices on the bus is asynchronous. A master/slave relationship exists throughout each bus transaction. Only one device has control of the bus at any one time. This controlling device is termed the bus master, or arbiter. The master device controls the bus when communicating with another device on the bus, termed the slave.

The bus master (typically the processor or a DMA device) initiates a bus transaction. The slave device responds by acknowledging the transaction in progress and by receiving data from, or transmitting data to, the bus master. Q22-bus control signals transmitted or received by the bus master or bus slave device must complete the sequence according to bus protocol.

The processor controls bus arbitration, that is, which device becomes bus master at any given time. A typical example of this relationship is a disk drive, as master, transferring data to memory as slave. Communication on the Q22-bus is interlocked so that, for certain control signals issued by the master device, there must be a response from the slave in order to complete the transfer. It is the master/slave signal protocol that makes the Q22-bus asynchronous. The asynchronous operation precludes the need for synchronizing with, and waiting for, clock pulses.

Since bus cycle completion by the bus master requires response from the slave device, each bus master must include a timeout error circuit that aborts the bus cycle if the slave does not respond to the bus transaction within 10 μs. The actual time before a timeout error occurs must be longer than the reply time of the slowest peripheral or memory device on the bus.

A.2 Q22-bus Signal Assignments

Table A–1 lists the data and address signal assignments. Table A–2 lists the control signal assignments. Table A–3 lists the power and ground signal assignments. Table A–4 lists the spare signal assignments.

Table A–1 Data and Address Signal Assignments

Data and Address Signal	Pin Assignment
BDAL0	AU2
BDAL1	AV2
BDAL2	BE2
BDAL3	BF2
BDAL4	BH2
BDAL5	BJ2
BDAL6	BK2
BDAL7	BL2
BDAL8	BM2
BDAL9	BN2
BDAL10	BP2
BDAL11	BR2
BDAL12	BS2

Table A–1 (Cont.) Data and Address Signal Assignments

Data and Address Signal	Pin Assignment
BDAL13	BT2
BDAL14	BU2
BDAL15	BV2
BDAL16	AC1
BDAL17	AD1
BDAL18	BC1
BDAL19	BD1
BDAL20	BE1
BDAL21	BF1

Table A–2 Control Signal Assignments

Control Signal	Pin Assignment
Data Control	
BDOUT	AE2
BRPLY	AF2
BDIN	AH2
BSYNC	AJ2
BWTBT	AK2
BBS7	AP2
Interrupt Control	
BIRQ7	BP1
BIRQ6	AB1
BIRQ5	AA1
BIRQ4	AL2
BIAKO	AN2
BIAKI	AM2
DMA Control	
BDMR	AN1
BSACK	BN1
BDMGO	AS2

Table A–2 (Cont.) Control Signal Assignments

Control Signal	Pin Assignment
BDMGI	AR2
System Control	
BHALT	AP1
BREF	AR1
BEVNT	BR1
BINIT	AT2
BDCOK	BA1
BPOK	BB1

Table A–3 Power and Ground Signal Assignments

Power and Ground	Pin Assignment
+5 B (battery) or +12 B (battery)	AS1
+12 B	BS1
+5 B	AV1
+5	AA2
+5	BA2
+5	BV1
+12	AD2
+12	BD2
+12	AB2
-12	AB2
-12	BB2
GND	AC2
GND	AJ1
GND	AM1
GND	AT1
GND	BC2
GND	BJ1
GND	BM1
GND	BT1

Table A–4 Spare Signal Assignments

Spare	Pin Assignment
SSpare1	AE1
SSpare3	AH1
SSpare8	BH1
SSpare2	AF1
MSpareA	AK1
MSpareB	AL1
MSpareB	BK1
MSpareB	BL1
PSpare1	AU1
ASpare2	BU1

A.3 Data Transfer Bus Cycles

Data transfer bus cycles, executed by bus master devices, transfer 32-bit words or 8-bit bytes to or from slave devices. In block mode, multiple words can be transferred to sequential word addresses, starting from a single bus address. Data transfer bus cycles are listed and defined in Table A–5.

Table A–5 Data Transfer Operations

Bus Cycle	Definition	Function (with respect to the bus master)
DATI	Data word input	Read
DATO	Data word output	Write
DATOB	Data byte output	Write-byte
DATIO	Data word input/output	Read-modify-write
DATIOB	Data word input/byte output	Read-modify-write byte
DATBI	Data block input	Read block
DATBO	Data block output	Write block

The bus signals listed in Table A–6 are used in the data transfer operations described in Table A–5.

Table A–6 Bus Signals for Data Transfers

Signal	Definition	Function
BDAL<21:00> L	22 data/address lines	BDAL<15:00> L are used for word and byte transfers. BDAL<17:16> L are used for extended addressing, memory parity error (16), and memory parity error enable (17) functions. BDAL<21:18> L are used for extended addressing beyond 256 Kbytes.
BSYNC L	Bus cycle control	Indicates bus transaction in progress.
BDIN L	Data input indicator	Strobe signals
BDOUT L	Data output indicator	Strobe signals
BRPLY L	Slave's acknowledge of bus cycle	Strobe signals
BWTBT L	Write/byte control	Control signals
BBS7	I/O device select	Indicates address is in the I/O page.

Data transfer bus cycles can be reduced to five basic types: DATI, DATO(B), DATIO(B), DATBI, and DATBO. These transactions occur between the bus master and one slave device selected during the addressing part of the bus cycle.

A.3.1 Bus Cycle Protocol

Before initiating a bus cycle, the previous bus transaction must have been completed (BSYNC L negated) and the device must become bus master. The bus cycle can be divided into two parts: addressing and data transfer. During addressing, the bus master outputs the address for the desired slave device, memory location, or device register. The selected slave device responds by latching the address bits and holding this condition for the duration of the bus cycle until BSYNC L becomes negated. During data transfer the actual data transfer occurs.

A.3.2 Device Addressing

Device addressing of a data transfer bus cycle comprises an address setup and deskew time, and an address hold and deskew time. During address setup and deskew time, the bus master does the following operations:

- Asserts BDAL<21:00> L with the desired slave device address bits.
- Asserts BBS7 L if a device in the I/O page is being addressed.
- Asserts BWTBT L if the cycle is a DATO(B) or DATBO bus cycle.

During this time, the address, BBS7 L, and BWTBT L signals are asserted at the slave bus receiver for at least 75 ns before BSYNC goes active. Devices in the I/O page ignore the nine high-order address bits BDAL<21:13>, and instead, decode BBS7 L along with the 13 low-order address bits. An active BWTBT L signal during address setup time indicates that a DATO(B) or DATBO operation follows, while an inactive BWTBT L indicates a DATI, DATBI, or DATIO(B) operation.

The address hold and deskew time begins after BSYNC L is asserted.

The slave device uses the active BSYNC L bus received output to clock BDAL address bits, BBS7 L, and BWTBT L into its internal logic. BDAL<21:00> L, BBS7 L, and BWTBT L remain active for 25 ns minimum after the BSYNC L bus receiver goes active. BSYNC L remains active for the duration of the bus cycle.

Memory and peripheral devices are addressed similarly, except for the way the slave device responds to BBS7 L. Addressed peripheral devices must not decode address bits on BDAL<21:13> L. Addressed peripheral device can respond to a bus cycle when BBS7 L is asserted (low) during the addressing of the cycle. When asserted, BBS7 L indicates that the device address resides in the I/O page (the upper 4K address space). Memory devices generally do not respond to addresses in the I/O page; however, some system applications may permit memory to reside in the I/O page for use as DMA buffers, read-only memory bootstraps, and diagnostics.

DATI

The DATI bus cycle, shown in Figure A–1, is a read operation. During DATI, data is input to the bus master. Data consists of 16-bit word transfers over the bus. During data transfer of the DATI bus cycle, the bus master asserts BDIN L 100 ns minimum after BSYNC L is asserted. The slave device responds to BDIN L active as follows:

- Asserts BRPLY L 0 ns minimum (8 ns maximum to avoid bus timeout) after receiving BDIN L, and 125 ns maximum before BDAL bus driver data bits are valid.
- Asserts BDAL<21:00> L with the addressed data and error information 0 ns (minimum) after receiving BDIN, and 125 ns (maximum) after assertion of BRPLY.

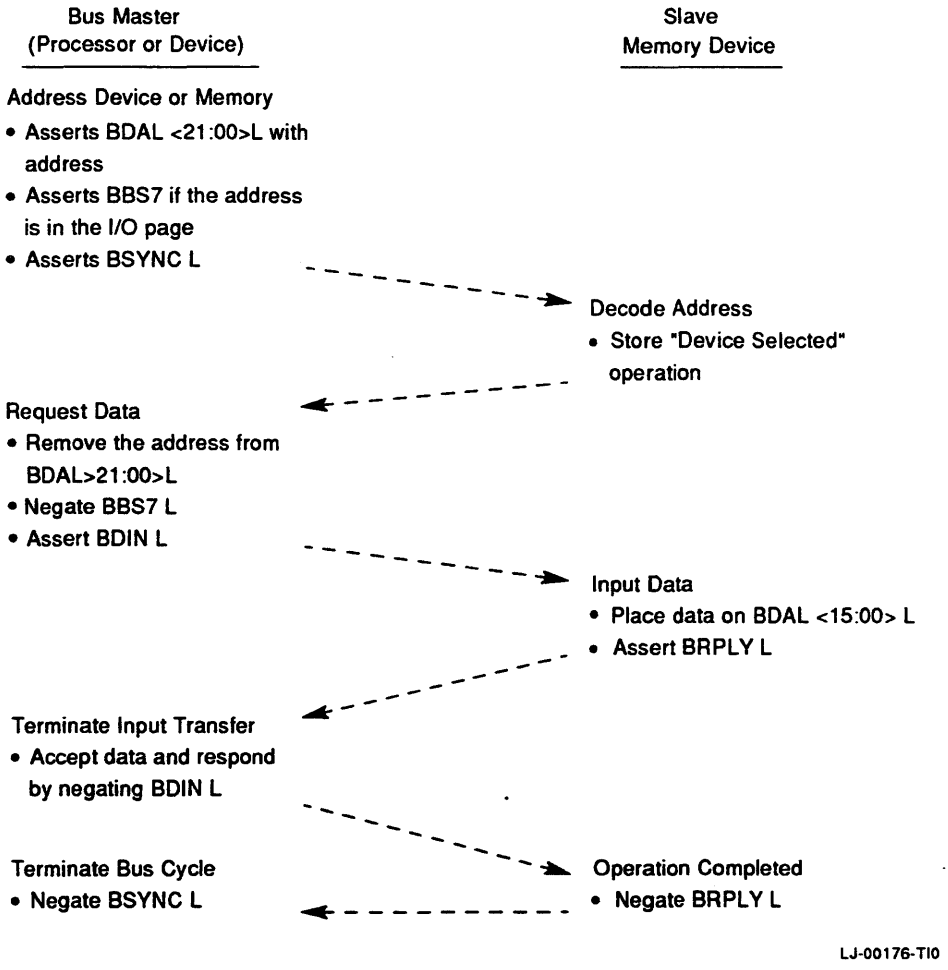


Figure A–1 DATI Bus Cycle

When the bus master receives BRPLY L, it does the following:

- Waits at least 200 ns deskew time and then accepts input data at BDAL<17:00> L bus receivers. BDAL <17:16> L are used for transmitting parity errors to the master.
- Negates BDIN L 200 ns minimum to 2 μs maximum after BRPLY L goes active.

The slave device responds to BDIN L negation by negating BRPLY L and removing read data from BDAL bus drivers. BRPLY L must be negated 100 ns maximum prior to removal of read data. The bus master responds to the negated BRPLY L by negating BSYNC L.

Conditions for the next BSYNC L assertion are as follows:

- BSYNC L must remain negated for 200 ns minimum.
- BSYNC L must not become asserted within 300 ns of previous BRPLY L negation.

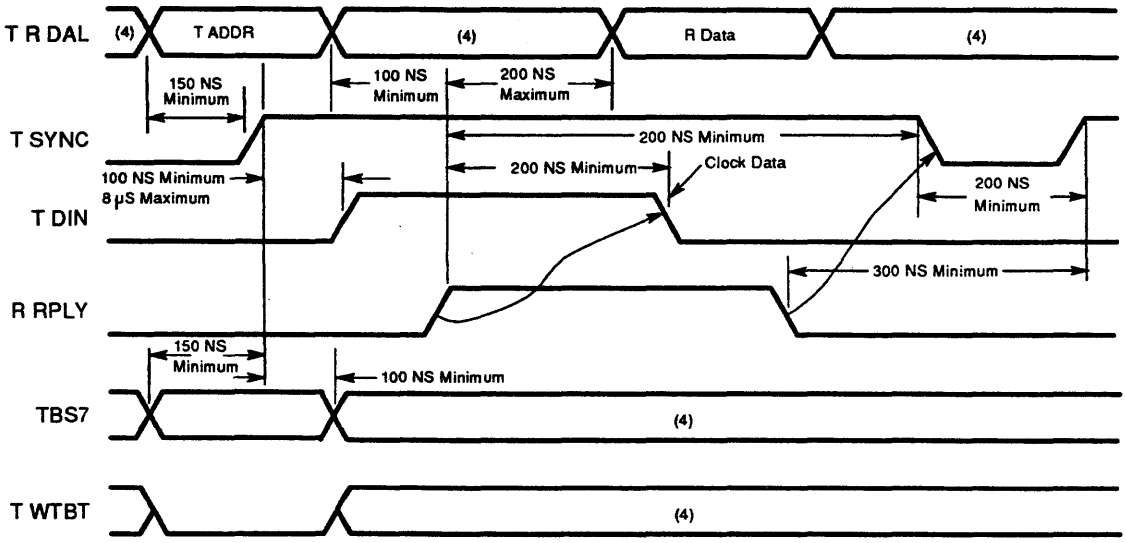
Figure A–2 shows DATI bus cycle timing.

NOTE

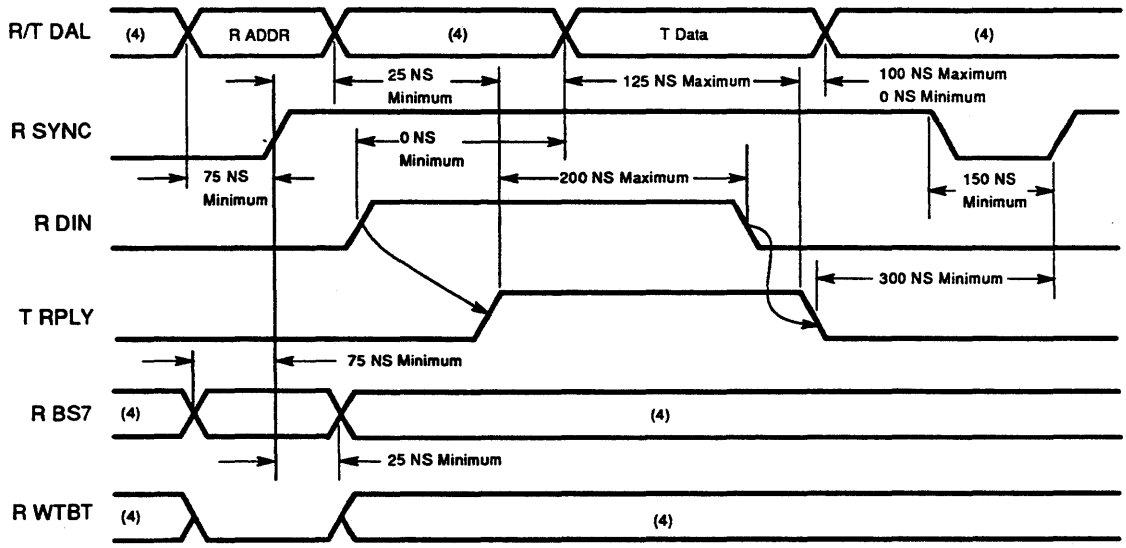
Continuous assertion of BSYNC L retains control of the bus by the bus master, and the previously addressed slave device remains selected. This is done for DATIO(B) bus cycles where DATO or DATOB follows a DATI without BSYNC L negation and a second device addressing operation. Also, a slow slave device can hold off data transfers to itself by keeping BRPLY L asserted, which causes the master to keep BSYNC L asserted.

DATOB

DATOB, shown in Figure A–3, is a write operation. Data is transferred in 32-bit words (DATO) or 8-bit bytes (DATOB) from the bus master to the slave device. The data transfer output can occur after the addressing part of a bus cycle when BWTBT L has been asserted by the bus master, or immediately following an input transfer part of a DATIOB bus cycle.



TIMING AT MASTER DEVICE



TIMING AT SLAVE DEVICE

NOTES:

1. Timing shown at master and slave device bus driver inputs and bus receiver outputs.
2. Signal name prefixes are defined below:
   T=Bus Driver Input
   R=Bus Receiver Output
3. Bus driver output and bus receiver input signal names include a "B" prefix.
4. Don't care condition.

LJ-00177-TI0

Figure A–2 DATI Bus Cycle Timing

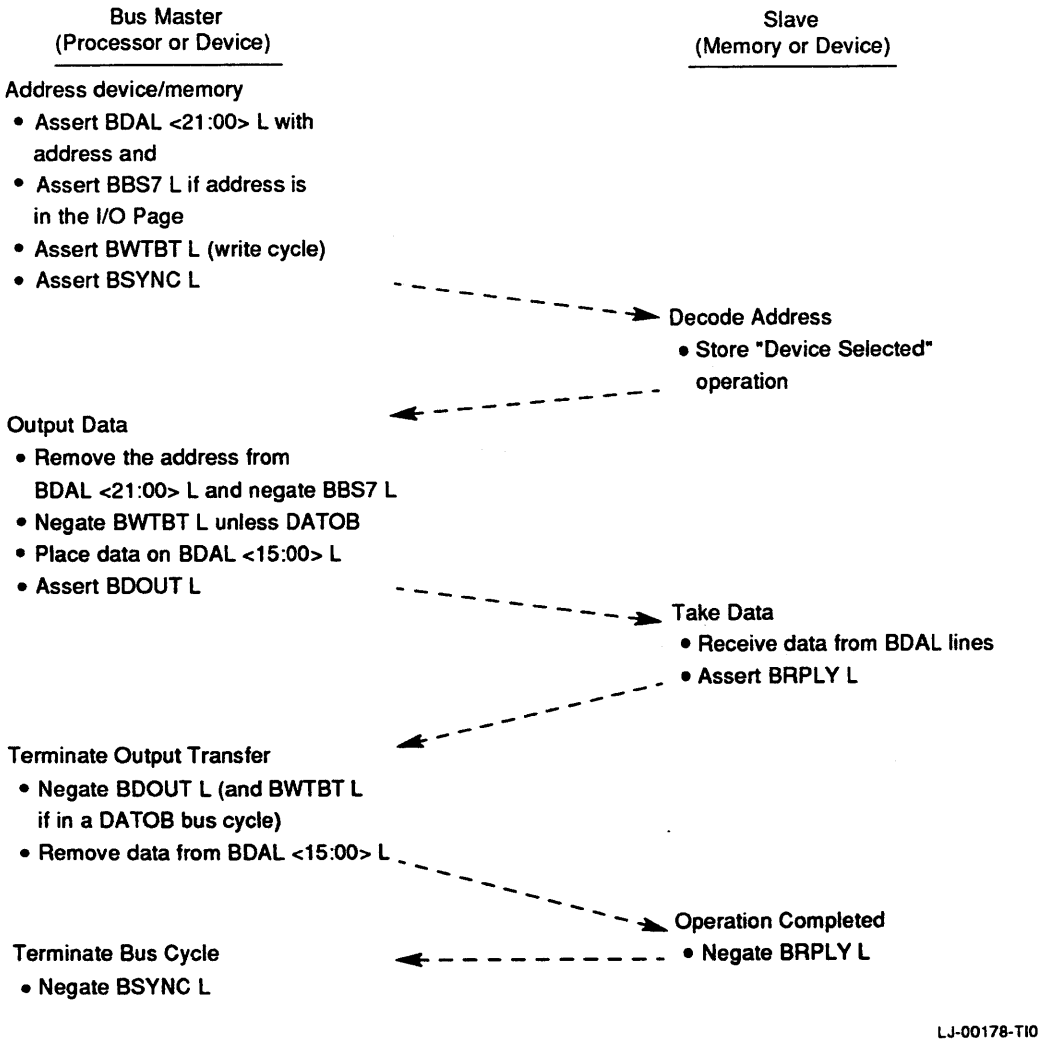


Figure A–3 DATO or DATOB Bus Cycle

The data transfer part of a DATOB bus cycle comprises a data setup and deskew time and a data hold and deskew time.

During the data setup and deskew time, the bus master outputs the data on BDAL<15:00> L at least 100 ns after BSYNC L assertion. BWTBT L remains negated for the length of the bus cycle. If the transfer is a byte transfer, BWTBT L remains asserted. If it is the output of a DATIOB, BWTBT L becomes asserted and lasts the duration of the bus cycle.

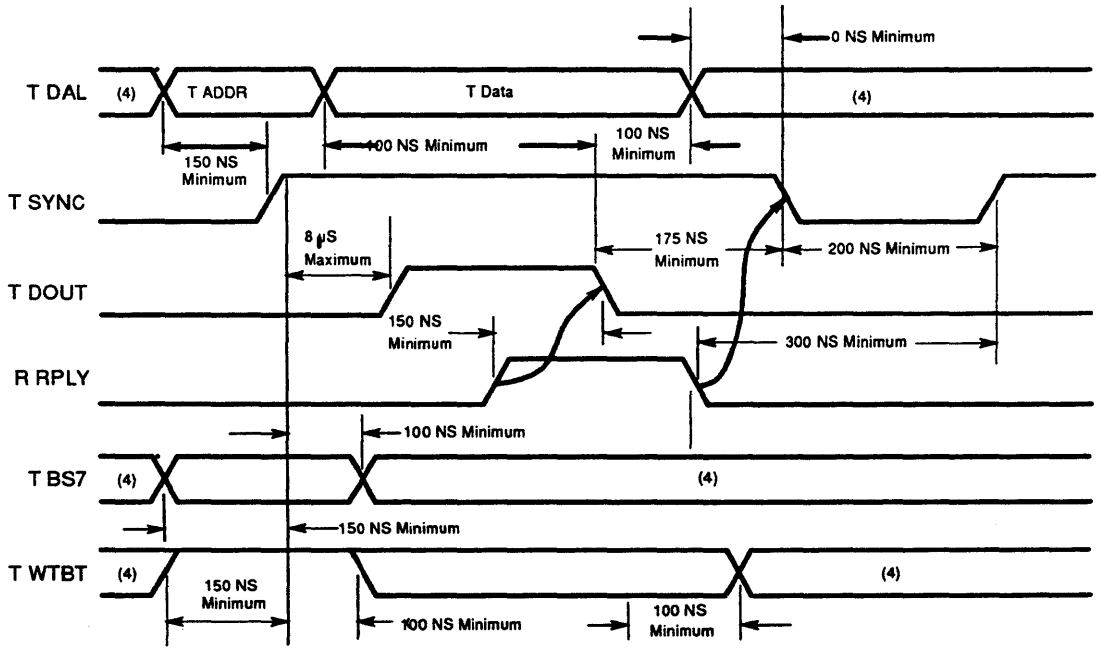
During a byte transfer, BDAL<00> L selects the high or low byte. This occurs in the addressing part of the cycle. If asserted, the high byte (BDAL<15:08> L) is selected; otherwise, the low byte (BDAL<07:00> L) is selected. An asserted BDAL 16 L at this time forces a parity error to be written into memory if the memory is a parity-type memory. BDAL 17 L is not used for write operations. The bus master asserts BDOUT L at least 100 ns after BDAL and BDWTBT L bus drivers are stable. The slave device responds by asserting BRPLY L within 10 μs to avoid bus timeout. This completes the data setup and deskew time.

During the data hold and deskew time, the bus master receives BRPLY L and negates BDOUT L, which must remain asserted for at least 150 ns from the receipt of BRPLY L before being negated by the bus master. BDAL<17:00> L bus drivers remain asserted for at least 100 ns after BDOUT L negation. The bus master then negates BDAL inputs.

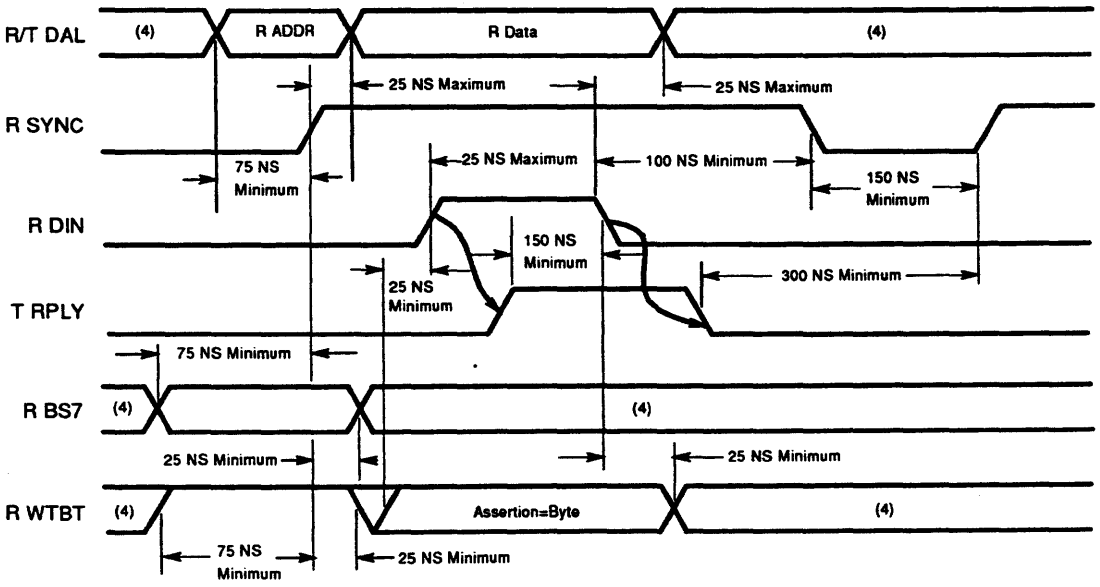
During this time, the slave device senses BDOUT L negation. The data is accepted and the slave device negates BRPLY L. The bus master responds by negating BSYNC L. However, the processor does not negate BSYNC L for at least 175 ns after negating BDOUT L. This completes the DATOB bus cycle. Before the next cycle, BSYNC L must remain unasserted for at least 200 ns. Figure A–4 shows DATOB bus cycle timing.

DATIOB

The protocol for a DATIOB bus cycle is identical to the addressing and data transfer part of the DATI and DATOB bus cycles, and is shown in Figure A–5. After addressing the device, a DATI cycle is performed as explained earlier; however, BSYNC L is not negated. BSYNC L remains active for an output word or byte transfer (DATOB). The bus master maintains at least 200 ns between BRPLY L negation during the DATI cycle and BDOUT L assertion. The cycle is terminated when the bus master negates BSYNC L, as described for DATOB. Figure A–6 illustrates DATIOB bus cycle timing.



TIMING AT MASTER DEVICE



TIMING AT SLAVE DEVICE

NOTES:

1. Timing shown at requesting device bus driver inputs and bus receiver outputs.
2. Signal name prefixes are defined below
   T=Bus Driver Input
   R=Bus Receiver Output
3. Bus driver output and bus receiver input signal names include a "B" prefix.
4. Don't care condition.

LJ-00179-TI0

Figure A–4 DATO or DATOB Bus Cycle Timing

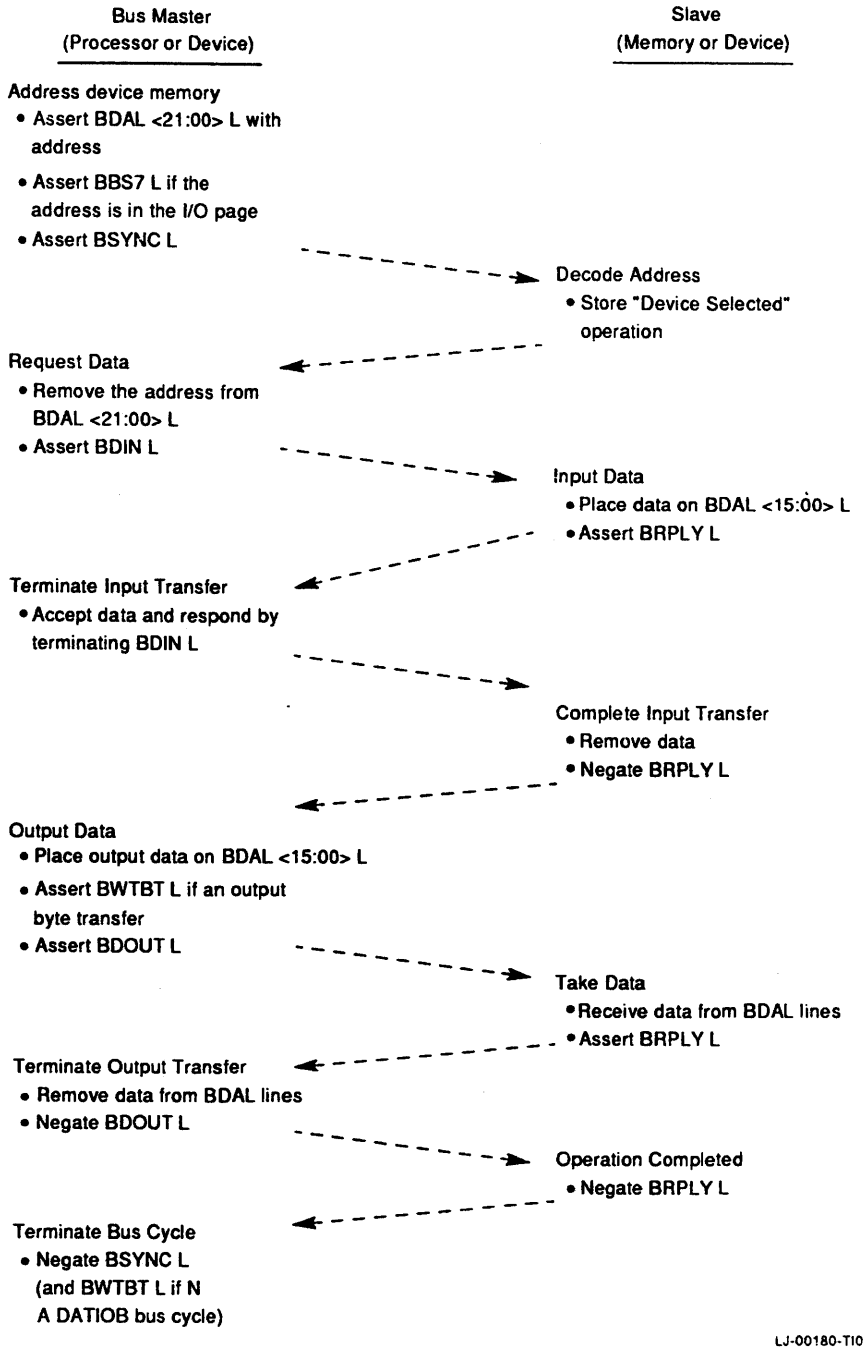
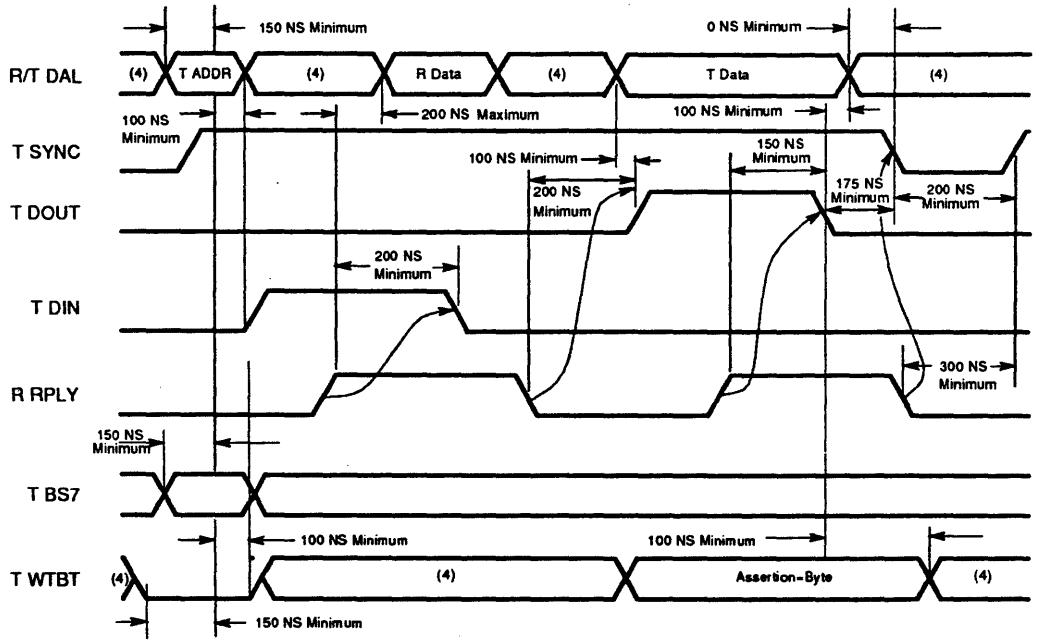
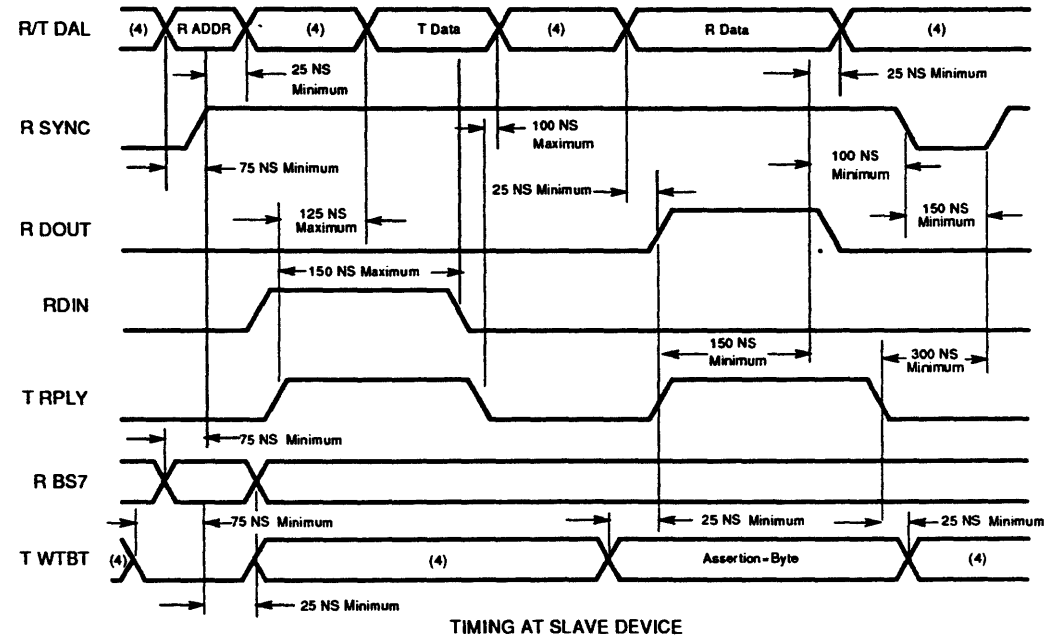


Figure A–5 DATIO or DATIOB Bus Cycle



TIMING AT MASTER DEVICE



TIMING AT SLAVE DEVICE

NOTES:

1. Timing shown at requesting device bus driver inputs and bus receiver outputs.
2. Signal name prefixes are defined below:
   T=Bus Driver Input
   R=Bus Receiver Output
3. Bus driver output and bus receiver input signal names include a "B" prefix.
4. Don't care condition.

LJ-00309-TI0

Figure A–6 DATIO or DATIOB Bus Cycle Timing

A.4 Direct Memory Access

The direct memory access (DMA) capability allows direct data transfer between I/O devices and memory. This is useful when using mass storage devices (for example, disks) that move large blocks of data to and from memory. A DMA device needs to be supplied with only the starting address in memory, the starting address in mass storage, the length of the transfer, and whether the operation is read or write. When this information is available, the DMA device can transfer data directly to or from memory. Since most DMA devices must perform data transfers in rapid succession or lose data, DMA devices are given the highest priority.

DMA is accomplished after the processor (normally bus master) has passed bus mastership to the highest priority DMA device that is requesting the bus. The processor arbitrates all requests and grants the bus to the DMA device electrically closest to it. A DMA device remains bus master until it relinquishes its mastership. The following control signals are used during bus arbitration:

- BDMGI L DMA grant input
- BDMGO L DMA grant output
- BDMR L DMA request line
- BSACK L bus grant acknowledge

A.4.1 DMA Protocol

A DMA transaction can be divided into the following three phases:

- Bus mastership acquisition phase
- Data transfer phase
- Bus mastership relinquishment phase

During the bus mastership acquisition phase, a DMA device requests the bus by asserting BDMR L. The processor arbitrates the request and initiates the transfer of bus mastership by asserting BDMGO L.

The maximum time between BDMR L assertion and BDMGO L assertion is DMA latency. This time is processor-dependent. BDMGO L/BDMGI L is one signal that is daisy-chained through each module in the backplane.

It is driven out of the processor on the BDMGO L pin, enters each module on the BDMGI L pin, and exits on the BDMGO L pin. This signal passes through the modules in descending order of priority until it is stopped by the requesting device. The requesting device blocks the output of BMDGO L and asserts BSACK L. If BDMR L is continuously asserted, the bus hangs.

During the data transfer phase, the DMA device continues asserting BSACK L. The actual data transfer is performed as described earlier.

The DMA device can assert BSYNC L for a data transfer 250 ns minimum after it received BDMGI L and its BSYNC L bus receiver is negated.

During the bus mastership relinquishment phase, the DMA device gives up the bus by negating BSACK L. This occurs after completing (or aborting) the last data transfer cycle (BRPLY L negated). BSACK L can be negated up to a maximum of 300 ns before negating BSYNC L.

NOTE
If multiple data transfers are performed during this phase, consideration must be given to the use of the bus for other system functions, such as memory refresh (if required).

Figure A–7 shows the DMA protocol, and Figure A–8 shows DMA request/grant timing.

A.4.2 Block Mode DMA

For increased throughput, block mode DMA can be implemented on a device for use with memories that support this type of transfer. In a block mode transaction, the starting memory address is asserted, followed by data for that address, and data for consecutive addresses.

By eliminating the assertion of the address for each data word, the transfer rate is almost doubled.

There are two types of block mode transfers, DATBI (input) and DATBO (output). The DATBI bus cycle is described in Section A.4.2.1 and illustrated in Figure A–9.

The DATBO bus cycle is described in Section A.4.2.2 and illustrated in Figure A–10.

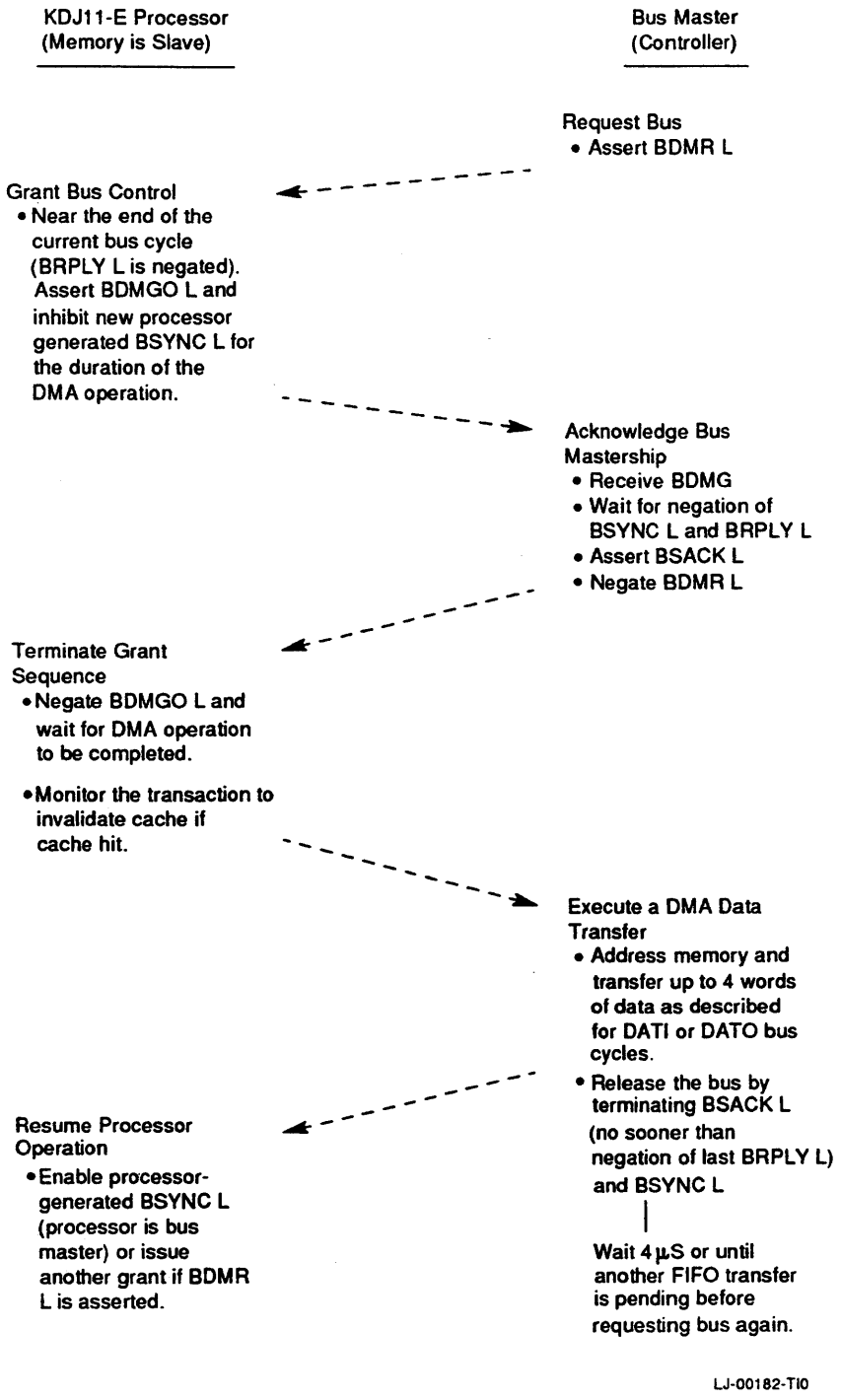
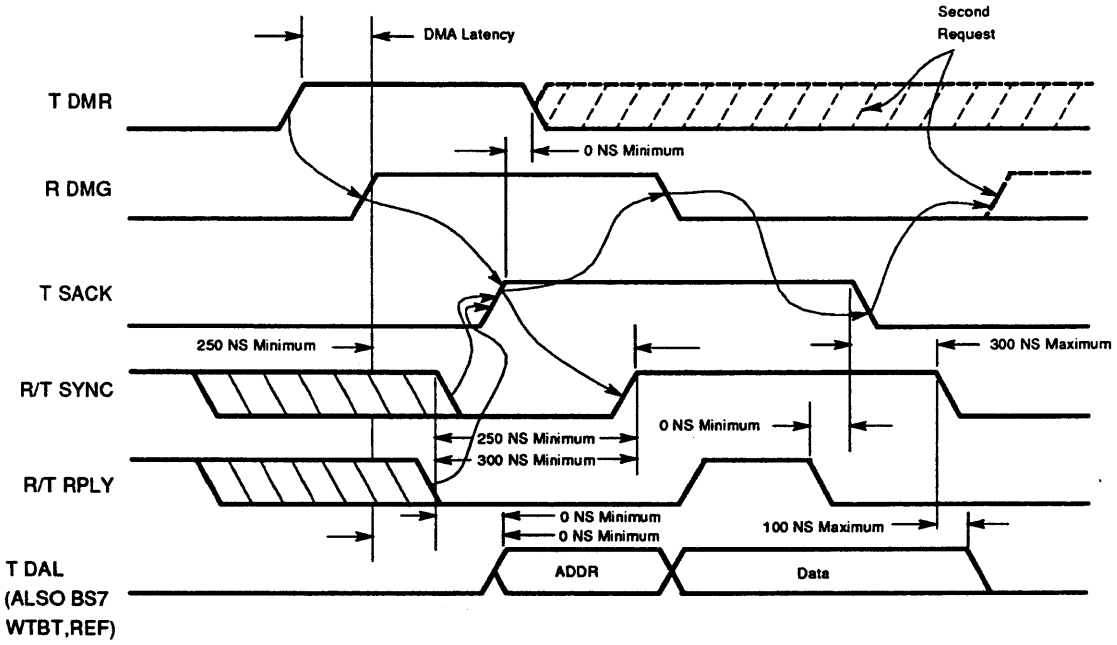


Figure A–7 DMA Protocol



NOTES:

1. Timing shown at requesting device bus driver inputs and bus receiver outputs.
2. Signal name prefixes are defined below
   T=Bus Driver Input
   R=Bus Receiver Output
3. Bus driver output and bus receiver input signal names include a "B" prefix.

LJ-00183-TI0

Figure A–8 DMA Request/Grant Timing

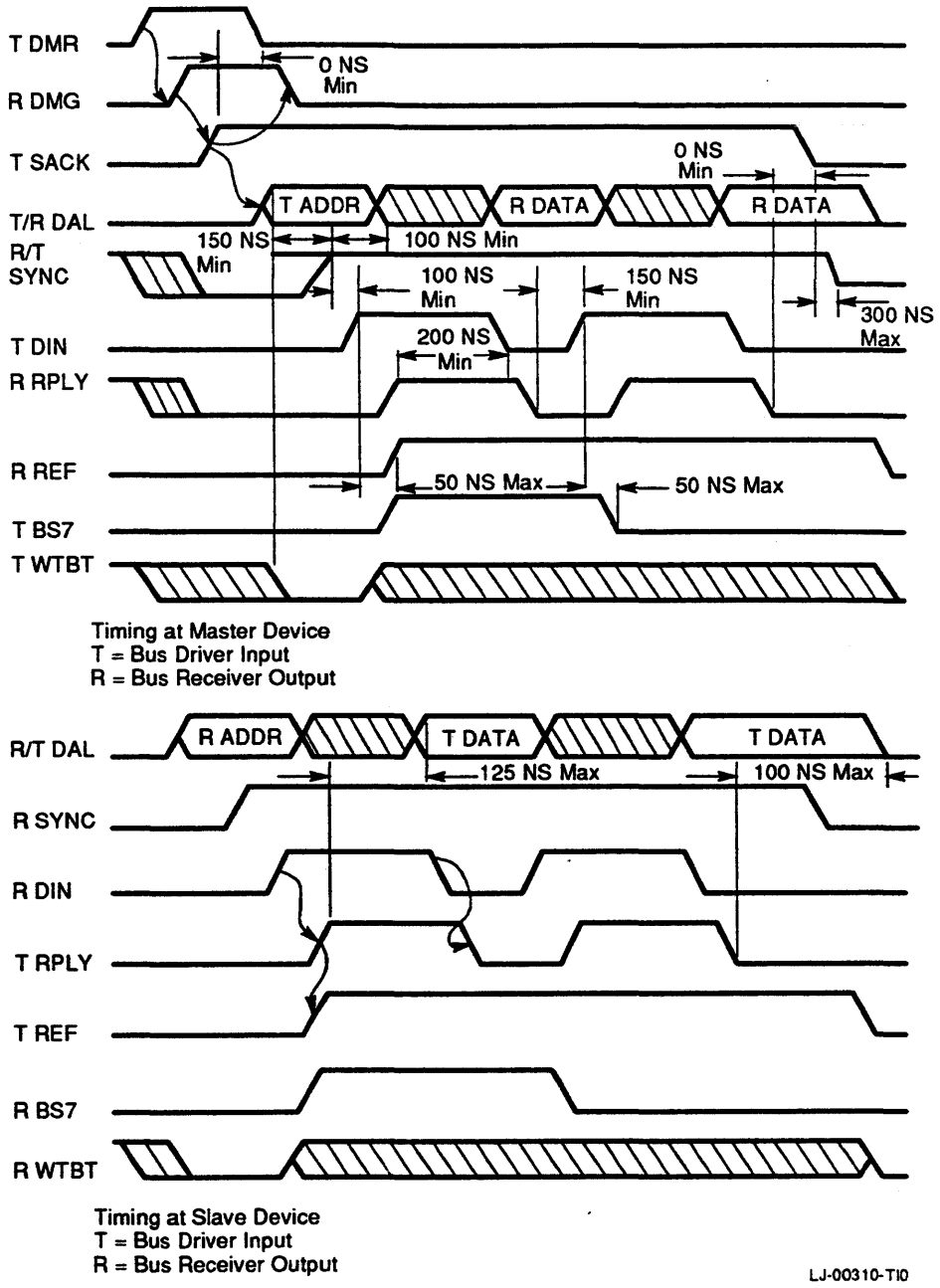


Figure A–9 DATBI Bus Cycle Timing

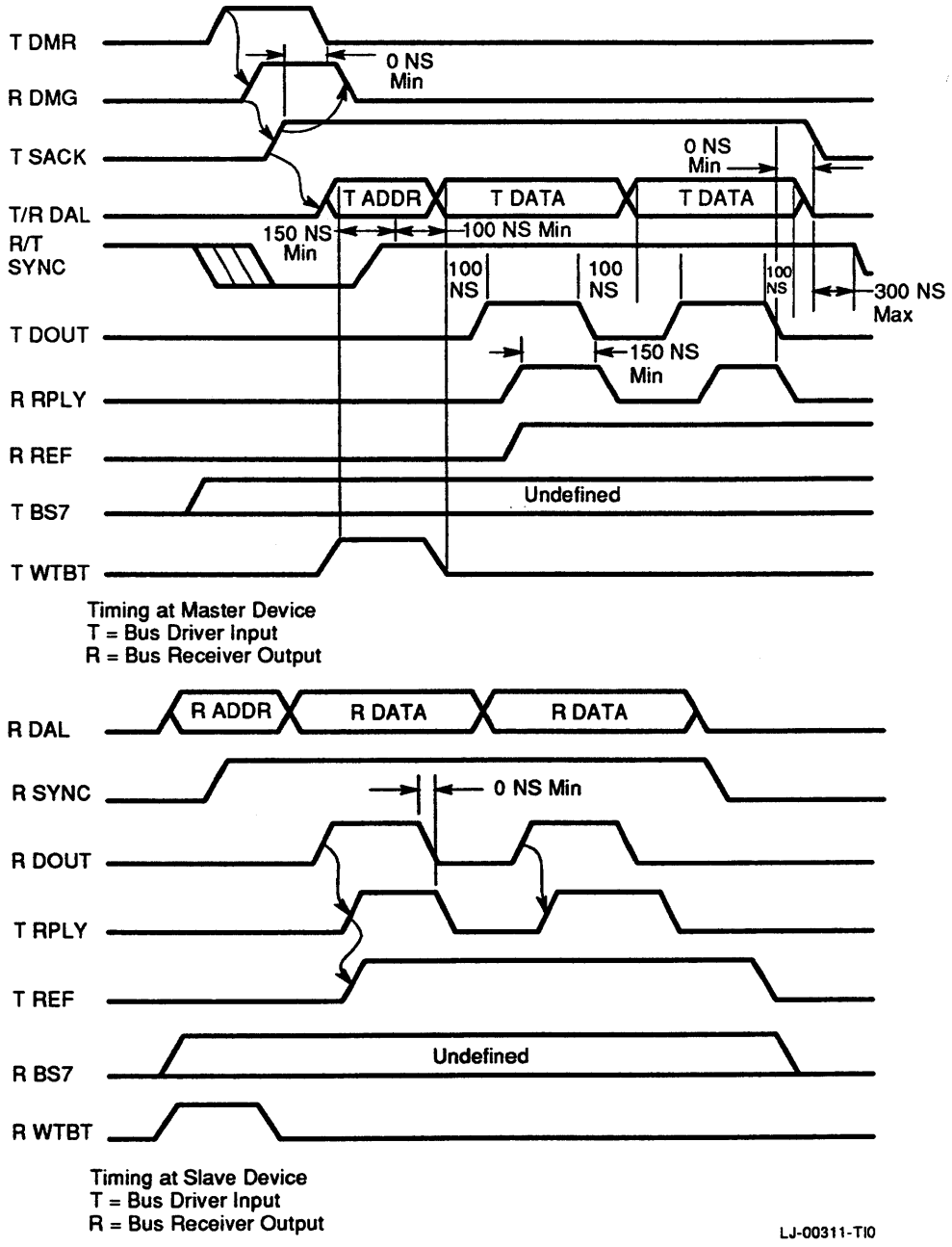


Figure A–10 DATBO Bus Cycle Timing

A.4.2.1 DATBI Bus Cycle

Before a DATBI block mode transfer can occur, the DMA bus master device must request control of the bus. This occurs under conventional Q22-bus protocol.

A block mode DATBI transfer is executed as follows:

- **Address device memory**–the address is asserted by the bus master on TADDR<21:00> along with the negation of TWTBT. The bus master asserts TSYNC 150 ns minimum after gating the address onto the bus.
- **Decode address**–the appropriate memory device recognizes that it must respond to the address on the bus.
- **Request data**–the address is removed by the bus master from TADDR<21:00> 100 ns minimum after the assertion of TSYNC. The bus master asserts the first TDIN 100 ns minimum after asserting TSYNC. The bus master asserts TBS7 50 ns maximum after asserting TDIN for the first time. TBS7 remains asserted until 50 ns maximum after the assertion of TDIN for the last time. In each case, TBS7 can be asserted or negated as soon as the conditions for asserting TDIN are met. The assertion of TBS7 indicates the bus master is requesting another read cycle after the current read cycle.
- **Send data**–the bus slave asserts TRPLY 0 ns minimum (8000 ns maximum to avoid a bus timeout) after receiving RDIN. The bus slave asserts TREF concurrent with TRPLY if, and only if, it is a block mode device which can support another RDIN after the current RDIN. The bus slave gates TDATA<15:00> onto the bus 0 ns minimum after receiving RDIN and 125 ns maximum after the assertion of TRPLY.

  NOTE
  Block mode transfers must not cross 16-word boundaries.

- **Terminate input transfer**–the bus master receives stable RDATA<15:00> from 200 ns maximum after receiving RRPLY until 20 ns minimum after the negation of RDIN. (The 20 ns minimum represents total minimum receiver delays for RDIN at the slave and RDATA<15:00> at the master.) The bus master negates TDIN 200 ns minimum after receiving RRPLY.
- **Operation completed**–the bus slave negates TRPLY 0 ns minimum after receiving the negation of RDIN. If RBS7 and TREF are both asserted when TRPLY negates, the bus slave prepares for another DIN cycle. RBS7 is stable from 125 ns after RDIN is received until 150 ns after TRPLY negates. If TBS7 and RREF were both asserted when TDIN negated, the bus master asserts TDIN 150 ns minimum after receiving the negation of RRPLY and continues with the timing relationship in send data above. RREF is stable from 75 ns after RRPLY asserts until 20 ns minimum after TDIN negates. (The 0 ns minimum represents total minimum receiver delays for RDIN at the slave and RREF at the master.)

NOTE
The bus master must limit itself to not more than eight transfers unless it monitors RDMR. If it monitors RDMR, it may perform up to 16 transfers as long as RDMR is not asserted at the end of the seventh transfer.

- **Terminate bus cycle**–if RBS7 and TREF were not both asserted when TRPLY negated, the bus slave removes TDATA<15:00> from the bus 0 ns minimum and 100 ns maximum after negating TRPLY. If TBS7 and RREF were not both asserted when TDIN negated, the bus master negates TSYNC 250 ns minimum after receiving the last assertion of RRPLY and 0 ns minimum after the negation of that RRPLY.
- **Release the bus**–the DMA bus master negates TSACK 0 ns after negation of the last RRPLY. The DMA bus master negates TSYNC 300 ns maximum after it negates TSACK. The DMA bus master must remove RDATA<15:00>, TBS7, and TWTBT from the bus 100 ns maximum after clearing TSYNC.

At this point the block mode transfer is complete, and the bus arbitration logic in the CPU enables processor-generated TSYNC or issues another bus grant (TDMGO) if RDMR is asserted.

A.4.2.2 DATBO Bus Cycle

Before a block mode transfer can occur, the DMA bus master device must request control of the bus. This occurs under conventional Q22-bus protocol.

A Block mode DATBO transfer is executed as follows:

- **Address device memory**–the address is asserted by the bus master on TADDR<21:00> along with the aasertion of TWTBT. The bus master asserts TSYNC 150 ns minimum after gating the address onto the bus.
- **Decode address**–the appropriate memory device recognizes that it must respond to the address on the bus.
- **Send data**–the bus master gates TDATA<15:00> along with TWTBT 100 ns minimum after the assertion of TSYNC. TWTBT is negated. The bus master asserts the first TDOUT 100 ns minimum after gating TDATA<15:00>.

NOTE
During DATBO cycles, TBS7 is undefined.

- **Receive data**–the bus slave receives stable data on RDATA<15:00> from 25 ns minimum before receiving RDOUT until 25 ns minimum after receiving the negation of RDOUT. The bus slave asserts TRPLY 0 ns minimum after receiving RDOUT. The bus slave asserts TREF concurrent with TRPLY if, and only if, it is a block mode device which can support another RDOUT after the current RDOUT.

NOTE
Block mode transfers must not cross 16-word boundaries.

- **Terminate output transfer**–the bus master negates TDOUT 150 ns minimum after receiving RRPLY.

- **Operation completed**–the bus slave negates TRPLY 0 ns minimum after receiving the negation of RDOUT. If RREF was asserted when TDOUT negated and if the bus master wants to transfer another word, the bus master gates the new data on TDATA<15:00> 100 ns minimum after negating TDOUT. RREF is stable from 75 ns maximum after RRPLY asserts until 20 ns minimum after RDOUT negates. (The 20 ns minimum represents minimum receiver delays for RDOUT at the slave and RREF at the master). The bus master asserts TDOUT 100 ns minimum after gating new data on TDATA<15:00> and 150 ns minimum after receiving the negation of RRPLY. The cycle continues with the timing relationship in receive data above.

  NOTE
  The bus master must limit itself to not more than 8 transfers unless it monitors RDMR. If it monitors RDMR, it may perform up to 16 transfers as long as RDMR is not asserted at the end of the seventh transfer.

- **Terminate bus cycle**–if RREF was not asserted when RRPLY negated or if the bus master has no additional data to transfer, the bus master removes data on TDATA<15:00> from the bus 100 ns minimum after negating TDOUT. If RREF was not asserted when TDOUT negated, the bus master negates TSYNC 275 ns minimum after receiving the last RRPLY and 0 ns minimum after the negation of the last RRPLY.

- **Release the bus**–the DMA bus master negates TSACK 0 ns after negation of the last RRPLY. The DMA bus master negates TSYNC 300 ns maximum after it negates TSACK. The DMA bus master must remove TDATA, TBS7, and TWTBT from the bus 100 ns maximum after clearing TSYNC.

At this point the block mode transfer is complete, and the bus arbitration logic in the CPU enables processor-generated TSYNC or issues another bus grant (TDMGO) if RDMR is asserted.

A.4.3 DMA Guidelines

The following is a list of DMA guidelines:

- Systems with memory refresh over the bus must not include devices that perform more than one transfer per acquisition.
- Bus masters that do not use block mode are limited to four DATI, four DATO, or two DATIO transfers per acquisition.
- Block mode bus masters that do not monitor BDMR are limited to eight transfers per acquisition.
- If BDMR is not asserted after the seventh transfer, block mode bus masters that do monitor BDMR may continue making transfers until the bus slave fails to assert BREF, or until they reach the total maximum of 16 transfers. Otherwise, they stop after eight transfers.

A.5 Interrupts

The interrupt capability of the Q22-bus allows an I/O device to temporarily suspend (interrupt) current program execution and divert processor operation to service the requesting device. The processor inputs a vector from the device to start the service routine (handler). Like the device register address, hardware fixes the device vector at locations within a designated range below location 001000. The vector indicates the first of a pair of addresses. The processor reads the contents of the first address, the starting address of the interrupt handler. The contents of the second address is a new processor status word (PS).

The new PS can raise the interrupt priority level, thereby preventing lower-level interrupts from breaking into the current interrupt service routine. Control is returned to the interrupted program when the interrupt handler is ended. The original interrupted program's address (PC) and its associated PS are stored on a stack. The original PC and PS are restored by a return from interrupt (RTI or RTT) instruction at the end of the handler. The use of the stack and the Q22-bus interrupt scheme can allow interrupts to occur within interrupts (nested interrupts), depending on the PS.

Interrupts can be caused by Q22-bus options or the MicroVAX CPU. Those interrupts that originate from within the processor are called traps. Traps are caused by programming errors, hardware errors, special instructions, and maintenance features.

The following Q22-bus signals are used in interrupt transactions:

Signal	Definition
BIRQ4 L	Interrupt request priority level 4
BIRQ5 L	Interrupt request priority level 5
BIRQ6 L	Interrupt request priority level 6
BIRQ7 L	Interrupt request priority level 7
BIAKI L	Interrupt acknowledge input
BIAKO L	Interrupt acknowledge output
BDAL<21:00>	Data/address lines
BDIN L	Data input strobe
BRPLY L	Reply

A.5.1 Device Priority

The Q22-bus supports the following two methods of device priority:

- Distributed arbitration — priority levels are implemented on the hardware. When devices of equal priority level request an interrupt, priority is given to the device electrically closest to the processor.
- Position-defined arbitration — priority is determined solely by electrical position on the bus. The closer a device is to the processor, the higher its priority is.

A.5.2 Interrupt Protocol

Interrupt protocol on the Q22-bus has three phases:

- Interrupt request
- Interrupt acknowledge and priority arbitration
- Interrupt vector transfer phase

The interrupt request phase begins when a device meets its specific conditions for interrupt requests. For example, the device is ready, done, or an error occurred. The interrupt enable bit in a device status register must be set. The device then initiates the interrupt by asserting the interrupt request line(s). BIRQ4 L is the lowest hardware priority level and is asserted for all interrupt requests for compatibility with previous Q22-bus processors. The level at which a device is configured must also be asserted. A special case exists for level 7 devices that must also assert level 6. The following list gives the interrupt levels and the corresponding Q22-bus interrupt request lines. For an explanation, refer to Section A.5.3.

Interrupt Level	Lines Asserted by Device
4	BIRQ4 L
5	BIRQ4 L, BIRQ5 L
6	BIRQ4 L, BIRQ6 L
7	BIRQ4 L, BIRQ6 L, BIRQ7 L

Figure A–11 shows the interrupt request/acknowledge sequence.

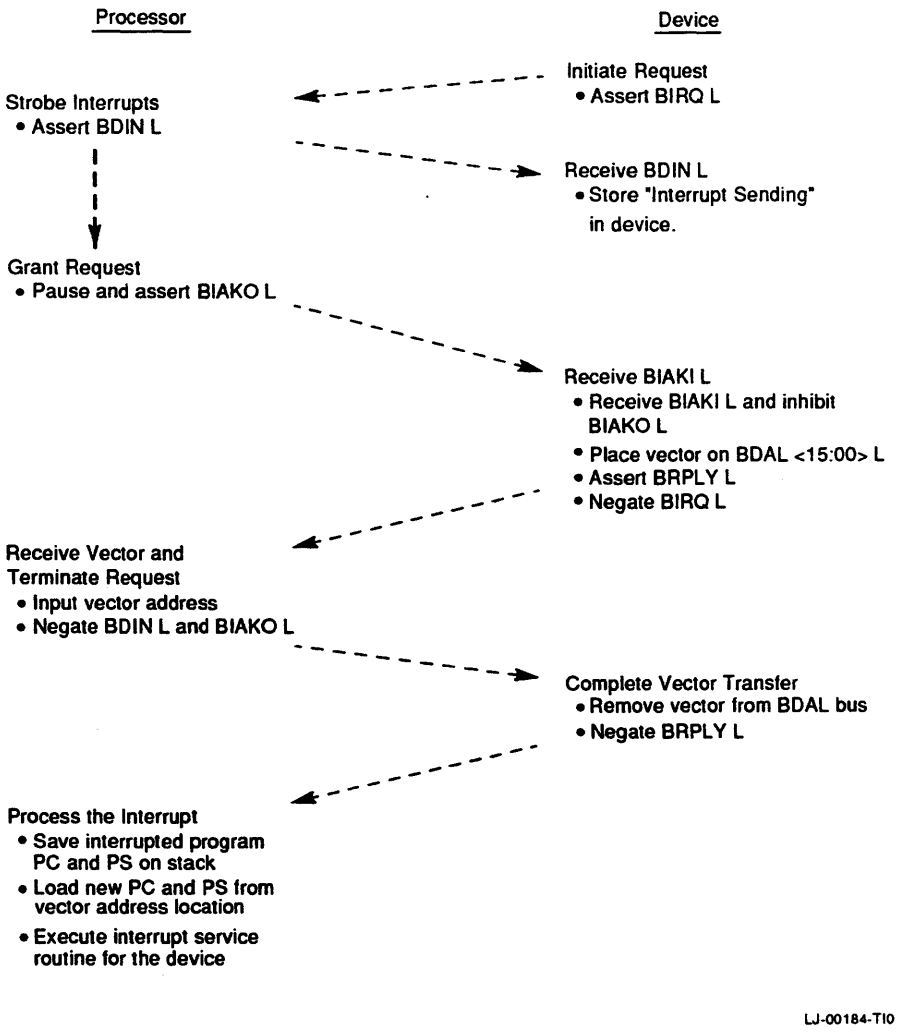


Figure A–11 Interrupt Request/Acknowledge Sequence

The interrupt request line remains asserted until the request is acknowledged.

During the interrupt acknowledge and priority arbitration phase, the processor acknowledges interrupts under the following conditions:

- The device interrupt priority is higher than the current PS<7:5>.
- The processor has completed instruction execution and no additional bus cycles are pending.

The processor acknowledges the interrupt request by asserting BDIN L, and 150 ns minimum later asserting BIAKO L. The device electrically closest to the processor receives the acknowledge on its BIAKI L bus receiver.

At this point, the two types of arbitration must be discussed separately. If the device that receives the acknowledge uses the four-level interrupt scheme, it reacts as follows:

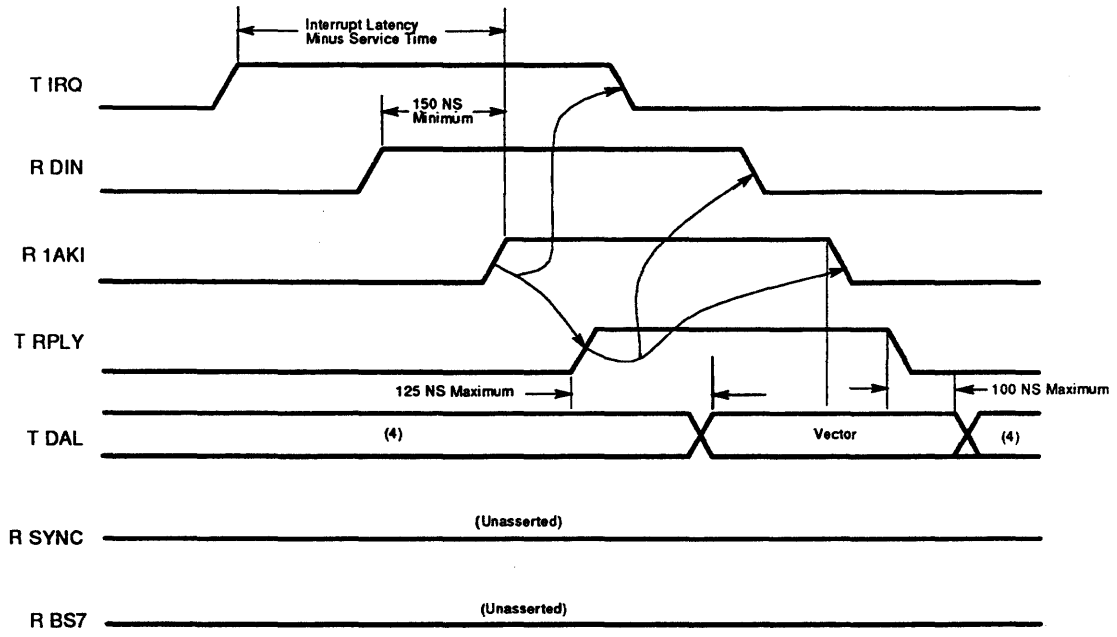
- If not requesting an interrupt, the device asserts BIAKO L and the acknowledge propagates to the next device on the bus.
- If the device is requesting an interrupt, it must check that no higher-level device is currently requesting an interrupt. This is done by monitoring higher-level request lines. The following table lists the lines that need to be monitored by devices at each priority level:

Device Priority Level	Line(s) Monitored
4	BIRQ5, BIRQ6
5	BIRQ6
6	BIRQ7
7	–

  In addition to asserting levels 7 and 4, level 7 devices must drive level 6. This is done to simplify the monitoring and arbitration by level 4 and 5 devices. In this protocol, level 4 and 5 devices need not monitor level 7 because level 7 devices assert level 6. Level 4 and 5 devices become aware of a level 7 request because they monitor the level 6 request. This protocol has been optimized for level 4, 5, and 6 devices, since level 7 devices are very seldom necessary.

- If no higher-level device is requesting an interrupt, the acknowledge is blocked by the device. (BIAKO L is not asserted.) Arbitration logic within the device uses the leading edge of BDIN L to clock a flip-flop that blocks BIAKO L. Arbitration is won and the interrupt vector transfer phase begins.
- If a higher-level request line is active, the device disqualifies itself and asserts BIAKO L to propagate the acknowledge to the next device along the bus.

Signal timing must be considered carefully when implementing four-level interrupts (Figure A–12).



NOTES:

1. Timing shown at requesting device bus driver inputs and bus receiver outputs.
2. Signal name prefixes are defined below
   T=Bus Driver Input
   R=Bus Receiver Output
3. Bus driver output and bus receiver input signal names include a "B" prefix.
4. Don't care condition.

LJ-00185-TI0

Figure A–12 Interrupt Protocol Timing

If a single-level interrupt device receives the acknowledge, it reacts as follows:

- If not requesting an interrupt, the device asserts BIAKO L and the acknowledge propagates to the next device on the bus.
- If the device was requesting an interrupt, the acknowledge is blocked using the leading edge of BDIN L, and arbitration is won. The interrupt vector transfer phase begins.

The interrupt vector transfer phase is enabled by BDIN L and BIAKI L. The device responds by asserting BRPLY L and its BDAL<15:00> L bus driver inputs with the vector address bits. The BDAL bus driver inputs must be stable within 125 ns maximum after BRPLY L is asserted. The processor then inputs the vector address and negates BDIN L and BIAKO L. The device then negates BRPLY L and 100 ns maximum later removes the vector address bits. The processor then enters the device's service routine.

NOTE
Propagation delay from BIAKI L to BIAKO L must not be greater than 500 ns per Q22-bus slot. The device must assert BRPLY L within 10 µs maximum after the processor asserts BIAKI L.

A.5.3 Q22-bus Four-Level Interrupt Configurations

If you have high-speed peripherals and desire better software performance, you can use the four-level interrupt scheme. Both position-independent and position-dependent configurations can be used with the four-level interrupt scheme.

Figure A–13 shows the position-independent configuration. This allows peripheral devices that use the four-level interrupt scheme to be placed in the backplane in any order. These devices must send out interrupt requests and monitor higher-level request lines as described. The level 4 request is always asserted from a requesting device regardless of priority. If two or more devices of equally high priority request an interrupt, the device physically closest to the processor wins arbitration. Devices that use the single-level interrupt scheme must be modified, or placed at the end of the bus, for arbitration to function properly.

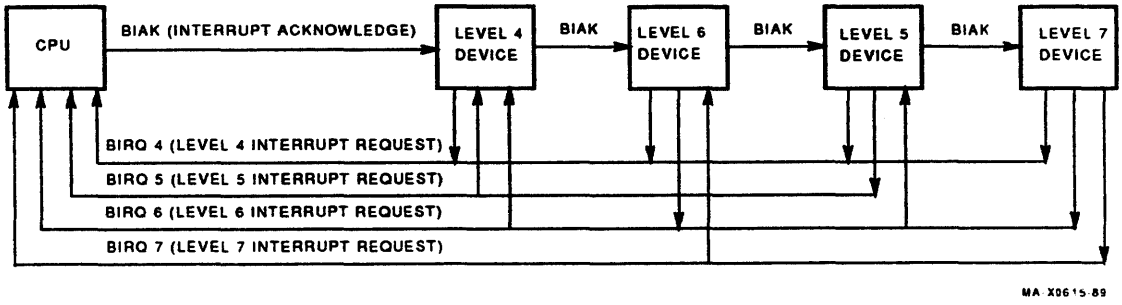


Figure A–13 Position-Independent Configuration

Figure A–14 shows the position-dependent configuration. This configuration is simpler to implement. A constraint is that peripheral devices must be inserted with the highest priority device located closest to the processor, and the remaining devices placed in the backplane in decreasing order of priority (with the lowest priority devices farthest from the processor). With this configuration, each device has to assert only its own level and level 4. Monitoring higher-level request lines is unnecessary. Arbitration is achieved through the physical positioning of each device on the bus. Single-level interrupt devices on level 4 should be positioned last on the bus.

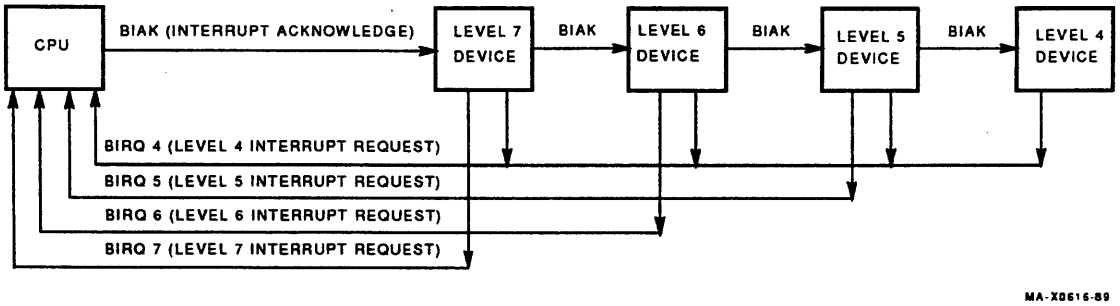


Figure A–14 Position-Dependent Configuration

A.6 Control Functions

The following Q22-bus signals provide control functions:

Signal	Definition
BREF L	Memory refresh (also block mode DMA)
BHALT L	Processor halt
BINIT L	Initialize
BPOK H	Power OK
BDCOK H	DC power OK

A.6.1 Halt

Assertion of BHALT L for at least 25 ns interrupts the processor, which stops program execution and forces the processor unconditionally into console I/O mode.

A.6.2 Initialization

Devices along the bus are initialized when BINIT L is asserted. The processor can assert BINIT L as a result of executing a reset instruction as part of a power-up or power-down sequence. BINIT L is asserted for approximately 10 µs when reset is executed.

A.6.3 Power Status

Power status protocol is controlled by two signals, BPOK H and BDCOK H. These signals are driven by an external device (usually the power supply).

A.7 Q22-bus Electrical Characteristics

The input and output logic levels for Q22-bus signals are given in Section A.7.1.

A.7.1 Signal Level Specifications

The signal level specifications for the Q22-bus are as follows:

Input Logic Level	
TTL logical low	0.8 Vdc maximum
TTL logical high	2.0 Vdc minimum
Output Logic Level	
TTL logical low	
TTL logical high	0.4 Vdc maximum
	2.4 Vdc minimum

A.7.2 Load Definition

AC loads make up the maximum capacitance allowed per signal line to ground. A unit load is defined as 9.35 pF of capacitance. DC loads are defined as maximum current allowed with a signal line driver asserted or unasserted. A unit load is defined as 210 µA in the unasserted state.

A.7.3 120-Ohm Q22-bus

The electrical conductors interconnecting the bus device slots are treated as transmission lines. A uniform transmission line, terminated in its characteristic impedance, propagates an electrical signal without reflections. Since bus drivers, receivers, and wiring connected to the bus have finite resistance and nonzero reactance, the transmission line impedance is not uniform, and introduces distortions into pulses propagated along it. Passive components of the Q22-bus (such as wiring, cabling, and etched signal conductors) are designed to have a nominal characteristic impedance of 120 ohms.

The maximum length of interconnecting cable, excluding wiring within the backplane, is limited to 4.88 m (16 ft).

A.7.4 Bus Drivers

Devices driving the 120-ohm Q22-bus must have open collector outputs and meet the following specifications:

DC Specifications

- Output low voltage when sinking 70 mA of current is 0.7 V maximum.
- Output high leakage current when connected to 3.8 Vdc is 25 μA (even if no power is applied, except for BDCOK H and BPOK H).
- These conditions must be met at worst-case supply temperature, and input signal levels.

AC Specifications

- Bus driver output pin capacitance load should not exceed 10 pF.
- Propagation delay should not exceed 35 ns.
- Skew (difference in propagation time between slowest and fastest gate) should not exceed 25 ns.
- Transition time (from 10% to 90% for positive transition—rise time, from 90% to 10% for negative transition—fall time) must be no faster than 10 ns.

A.7.5 Bus Receivers

Devices that receive signals from the 120-ohm Q22-bus must meet the following requirements:

DC Specifications

- Input low voltage maximum is 1.3 V.
- Input high voltage minimum is 1.7 V.
- Maximum input current when connected to 3.8 Vdc is 80 μA (even if no power is applied).

These specifications must be met at worst-case supply voltage, temperature, and output signal conditions.

AC Specifications

- Bus receiver input pin capacitance load should not exceed 10 pF.
- Propagation delay should not exceed 35 ns.
- Skew (difference in propagation time between slowest and fastest gate) should not exceed 25 ns.

A.7.6 Bus Termination

The 120-ohm Q22-bus must be terminated at each end by an appropriate terminator, as shown in Figure A–15. This is to be done as a voltage divider with its Thevenin equivalent equal to 120 ohms and 3.4 V (nominal). This type of termination is provided by an REV11-A refresh/boot/terminator, BDV11-AA, KPV11-B, TEV11, or by certain backplanes and expansion cards.

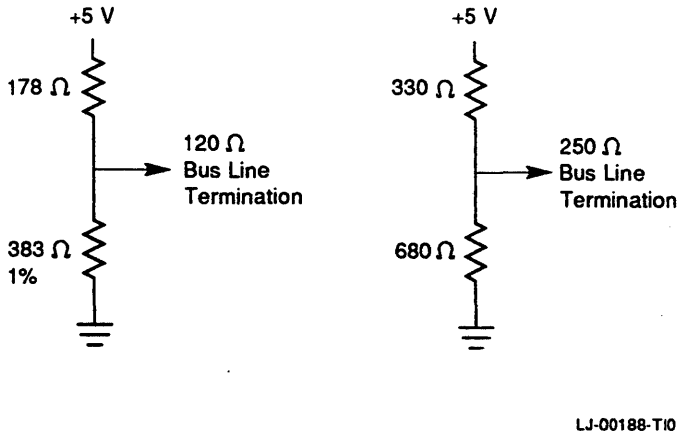


Figure A–15 Bus Line Terminations

Each of the several Q22-bus lines (all signals whose mnemonics start with the letter B) must see an equivalent network with the following characteristics at each end of the bus:

Bus Termination Characteristic	Value
Input impedance (with respect to ground)	120 ohm +5%, -15%
Open circuit voltage	3.4 Vdc +5%
Capacitance load	Not to exceed 30 pF

NOTE
The resistive termination can be provided by the combination of two modules. (The processor module supplies 220 ohms to ground. This, in parallel with another 220-ohm card, provides 120 ohms.) Both terminators must reside physically within the same backplane.

A.7.7 Bus Interconnecting Wiring

The following sections give specific information about bus interconnecting wiring.

A.7.7.1 Backplane Wiring

The wiring that connects all device interface slots on the Q22-bus must meet the following specifications:

- The conductors must be arranged so that each line exhibits a characteristic impedance of 120 ohms (measured with respect to the bus common return).
- Crosstalk between any two lines must be no greater than 5%. Note that worst-case crosstalk is manifested by simultaneously driving all but one signal line and measuring the effect on the undriven line.
- DC resistance of the signal path, as measured between the near-end terminator and the far-end terminator module (including all intervening connectors, cables, backplane wiring, and connector-module etch) must not exceed 20 ohms.
- DC resistance of the common return path, as measured between the near-end terminator and the far-end terminator module (including all intervening connectors, cables, backplane wiring and connector-module etch) must not exceed an equivalent of 2 ohms per signal path. Thus, the composite signal return path dc resistance must not exceed 2 ohms divided by 40 bus lines, or 50 milliohms. Note that although this common return path is nominally at ground potential, the conductance must be part of the bus wiring. The specified low impedance return path must be provided by the bus wiring as distinguished from the common system or power ground path.

A.7.7.2 Intrabackplane Bus Wiring

The wiring that connects the bus connector slots within one contiguous backplane is part of the overall bus transmission line. Owing to implementation constraints, the nominal characteristic impedance of 120 ohms may not be achievable. Distributed wiring capacitance in excess of the amount required to achieve the nominal 120-ohm impedance may not exceed 60 pF per signal line per backplane.

A.7.7.3 Power and Ground

Each bus interface slot has connector pins assigned for the following dc voltages. The maximum allowable current per pin is 1.5 A. +5 Vdc must be regulated to 5% with a maximum ripple of 100 mV pp. +12 Vdc must be regulated to 3% with a maximum ripple of 200 mV pp.

- +5 Vdc — three pins (4.5 A maximum per bus device slot)
- +12 Vdc — two pins (3.0 A maximum per bus device slot)
- Ground — eight pins (shared by power return and signal return)

NOTE

Power is not bused between backplanes on any interconnecting bus cables.

A.8 System Configurations

Q22-bus systems can be divided into two types:

- Systems containing one backplane
- Systems containing multiple backplanes

Before configuring any system, three characteristics for each module in the system must be identified.

- Power consumption — +5 Vdc and +12 Vdc are the current requirements.
- AC bus loading — the amount of capacitance a module presents to a bus signal line. AC loading is expressed in terms of ac loads, where one ac load equals 9.35 pF of capacitance.
- DC bus loading—the amount of dc leakage current a module presents to a bus signal when the line is high (undriven). DC loading is expressed in terms of dc loads, where one dc load equals 210 μA (nominal).

Power consumption, ac loading, and dc loading specifications for each module are included in the *Microcomputer Interfaces Handbook*.

NOTE
The ac and dc loads and the power consumption of the processor module, terminator module, and backplane must be included in determining the total loading of a backplane.

Rules for configuring single-backplane systems are as follows:

- When using a processor with 220-ohm termination, the bus can accommodate modules that have up to 20 ac loads before additional termination is required (Figure A–16). If more than 20 ac loads are included, the other end of the bus must be terminated with 120 ohms, and then up to 35 ac loads may be present.
- With 120-ohm processor termination, up to 35 ac loads can be used without additional termination. If 120-ohm bus termination is added, up to 45 ac loads can be configured in the backplane.
- The bus can accommodate modules up to 20 dc loads (total).
- The bus signal lines on the backplane can be up to 35.6 cm (14 in.) long.

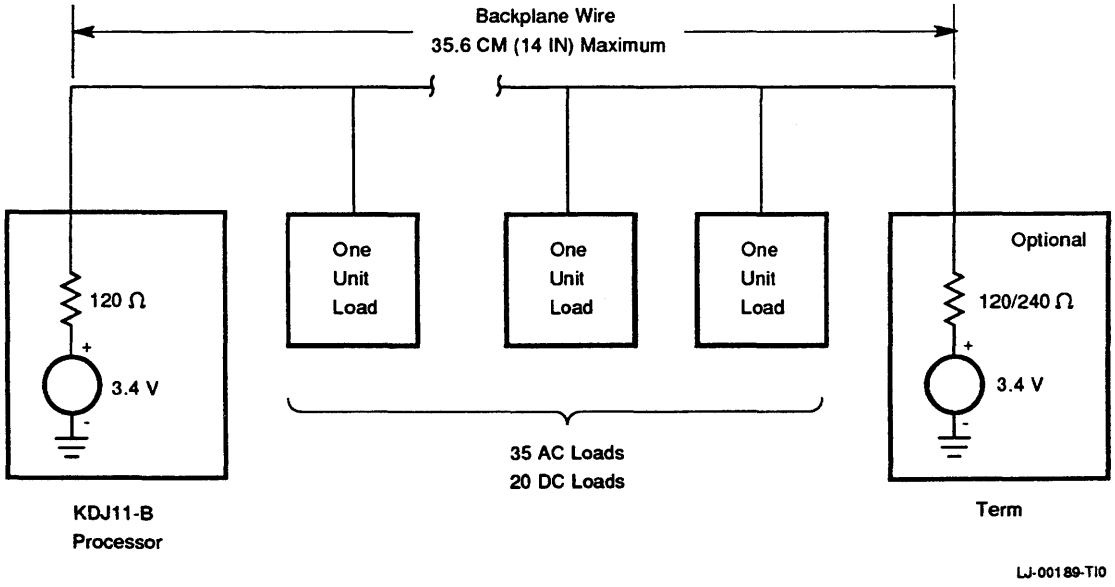
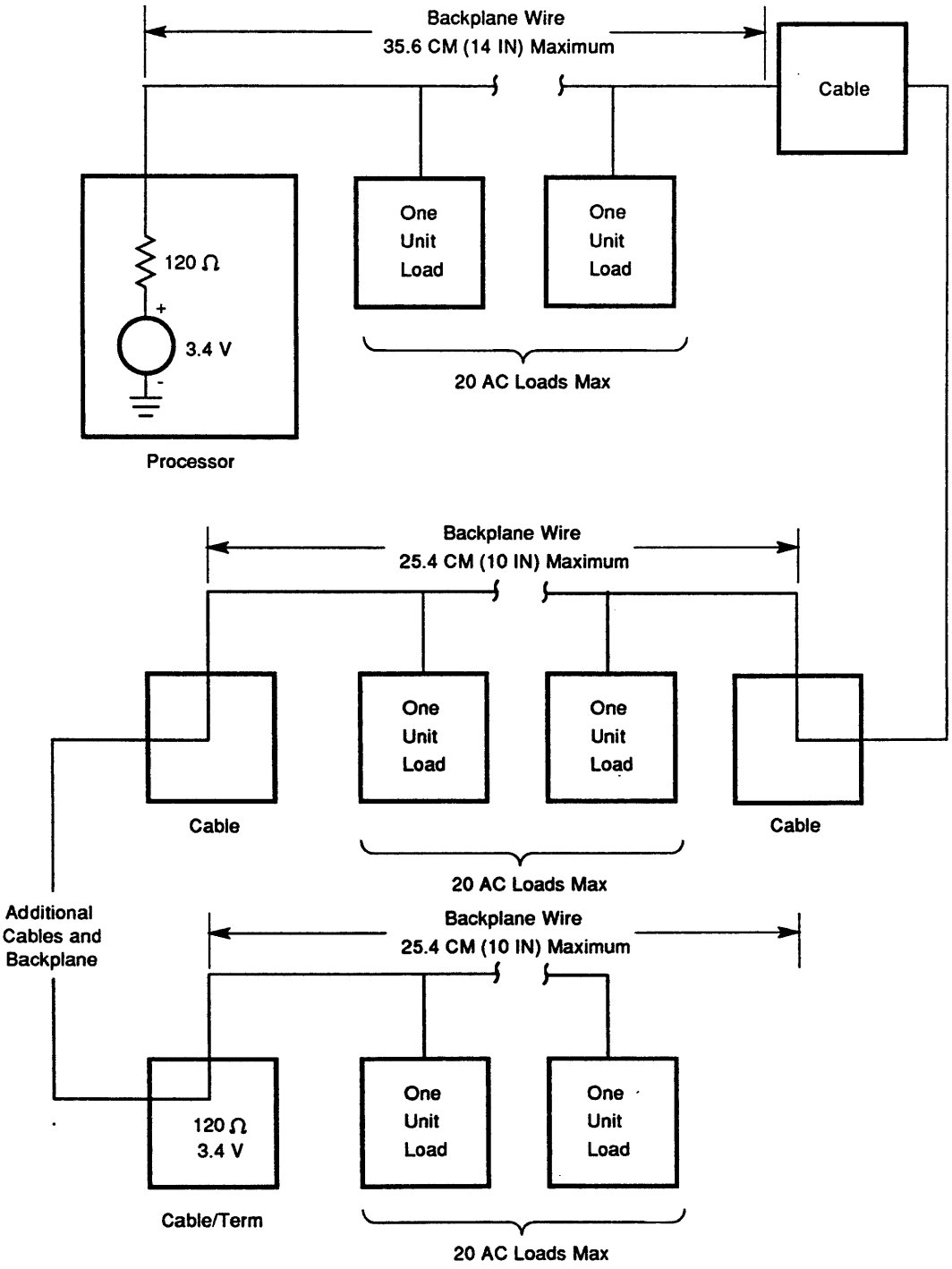


Figure A–16 Single-Backplane Configuration

Rules for configuring multiple backplane systems are as follows:

- Figure A–17 shows that up to three backplanes can make up the system.
- The signal lines on each backplane can be up to 25.4 cm (10 in.) long.
- Each backplane can accommodate modules that have up to 22 ac loads. Unused ac loads from one backplane may not be added to another backplane if the second backplane loading exceeds 22 ac loads. It is desirable to load backplanes equally, or with the highest ac loads in the first and second backplanes.
- DC loading of all modules in all backplanes cannot exceed 20 loads.
- Both ends of the bus must be terminated with 120 ohms. This means the first and last backplanes must have an impedance of 120 ohms. To achieve this, each backplane can be lumped together as a single point. The resistive termination can be provided by a combination of two modules in the backplane – the processor providing 220 ohms to ground in parallel with an expansion paddle card providing 250 ohms to give the needed 120-ohm termination.

  Alternately, a processor with 120-ohm termination would need no additional termination on the paddle card to attain 120 ohms in the first box. The 120-ohm termination in the last box can be provided in two ways: the termination resistors may reside either on the expansion paddle card, or on a bus termination card (such as the BDV11).
- The cable(s) connecting the first two backplanes is 61 cm (2 ft) or more in length.
- The cable(s) connecting the second backplane to the third backplane is 122 cm (4 ft) longer or shorter than the cable(s) connecting the first and second backplanes.
- The combined length of both cables cannot exceed 4.88 m (16 ft).
- The cables used must have a characteristic impedance of 120 ohms.



Notes:
1. Two cables (max) 4.88 M (16 FT) (Max) total length.
2. 20 DC loads total (max).

LJ-00312-TI0

Figure A–17 Multiple Backplane Configuration

A.8.1 Power Supply Loading

Total power requirements for each backplane can be determined by obtaining the total power requirements for each module in the backplane. Obtain separate totals for +5 V and +12 V power. Power requirements for each module are specified in the *Microcomputer Interfaces Handbook*.

When distributing power in multiple backplane systems, do not attempt to distribute power through the Q22-bus cables. Provide separate, appropriate power wiring from each power supply to each backplane. Each power supply should be capable of asserting BPOK H and BDCOK H signals according to bus protocol; this is required if automatic power-fail/restart programs are implemented, or if specific peripherals require an orderly power-down halt sequence. The proper use of BPOK H and BDCOK H signals is strongly recommended.

A.9 Module Contact Finger Identification

Digital's plug-in modules all use the same contact finger (pin) identification system. A typical pin is shown in Figure A–18.

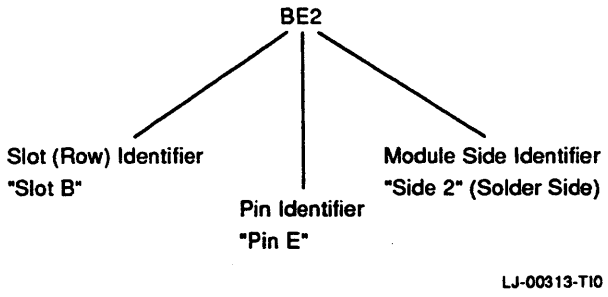


Figure A–18 Typical Pin Identification System

The Q22-bus is based on the use of quad-height modules that plug into a 2-slot bus connector. Each slot contains 36 lines (18 lines on both the component side and the solder side of the circuit board).

Slots, row A, and row B include a numeric identifier for the side of the module. The component side is designated side 1, the solder side is designated side 2, as shown in Figure A–19.

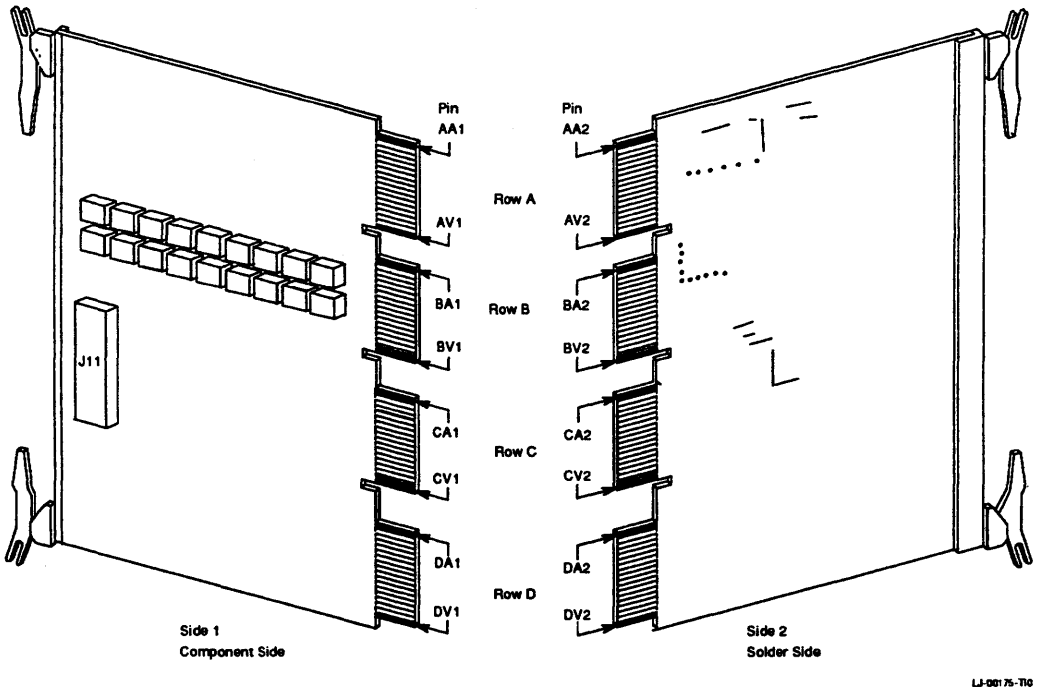


Figure A–19 Quad-Height Module Contact Finger Identification

Letters ranging from A through V (excluding G, I, O, and Q) identify a particular pin on a side of a slot. Table A–7 lists and identifies the bus pins of the quad-height module. A bus pin identifier ending with a 1 is found on the component side of the board, while a bus pin identifier ending with a 2 is found on the solder side of the board.

The positioning notch between the two rows of pins mates with a protrusion on the connector block for correct module positioning.

The dimensions for a typical Q22-bus module are represented in Figure A–20.

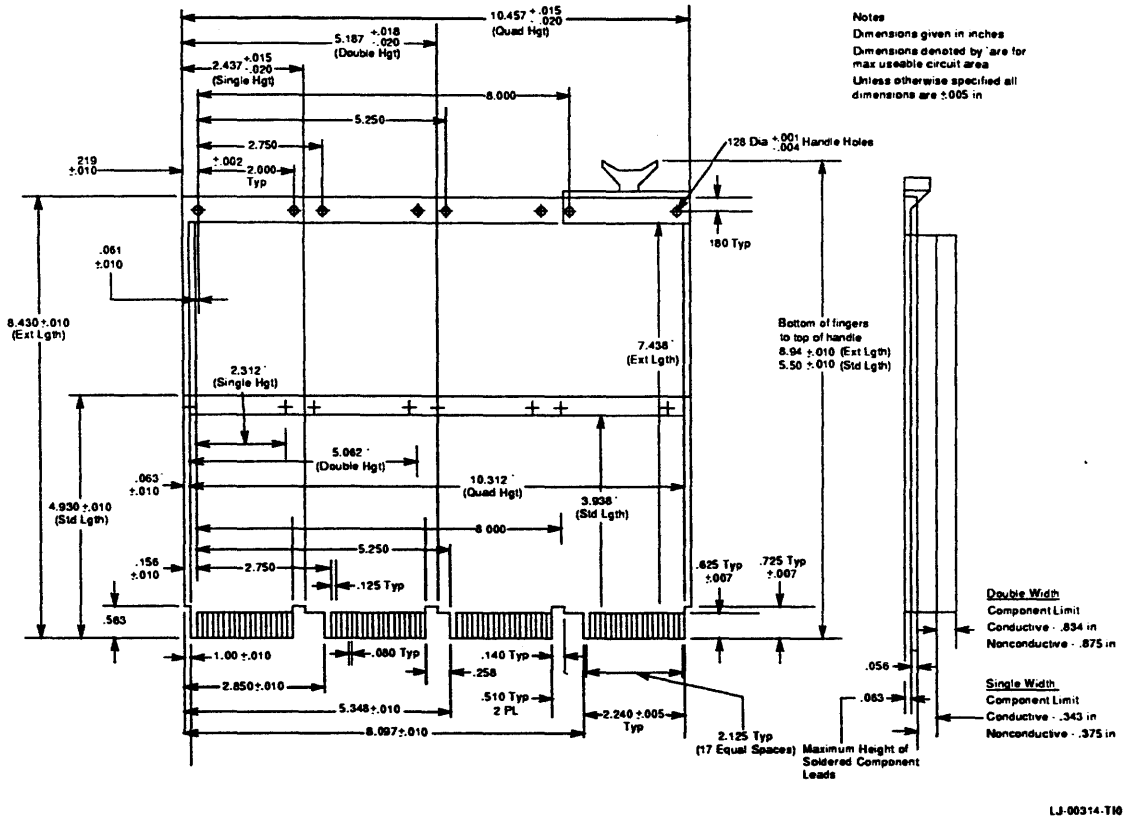


Figure A–20 Typical Q22-bus Module Dimensions

Table A–7 Bus Pin Identifiers

Bus Pin	Signal	Definition
AA1	BIRQ5 L	Interrupt request priority level 5.
AB1	BIRQ6 L	Interrupt request priority level 6.
AC1	BDAL16 L	Extended address bit during addressing protocol; memory error data line during data transfer protocol.
AD1	BDAL17 L	Extended address bit during addressing protocol; memory error logic enable during data transfer protocol.
AE1	SSPARE1 (alternate +5B)	Special spare — not assigned or bused in Digital's cable or backplane assemblies. Available for user connection. Optionally, this pin can be used for +5 V battery (+5 B) back-up power to keep critical circuits alive during power failures. A jumper is required on Q22-bus options to open (disconnect) the +5 B circuit in systems that use this line as SSPARE1.

Table A–7 (Cont.) Bus Pin Identifiers

Bus Pin	Signal	Definition
AF1	SSPARE2	Special spare — not assigned or bused in Digital's cable or backplane assemblies. Available for user interconnection. In the highest priority device slot, the processor can use this pin for a signal to indicate its run state.
AH1	SSPARE3 SRUN	Special spare — not assigned or bused simultaneously in Digital's cable or backplane assemblies; available for user interconnection. An alternate SRUN signal can be connected in the highest priority set.
AJ1	GND	Ground — system signal ground and dc return.
AK1	MSPAREA	Maintenance spare — normally connected together on the backplane at each option location (not a bused connection).
AL1	MSPAREB	Maintenance spare — normally connected together on the backplane at each option location (not a bused connection).
AM1	GND	Ground — system signal ground and dc return.
AN1	BDMR L	DMA request — a device asserts this signal to request bus mastership. The processor arbitrates bus mastership between itself and all DMA devices on the bus. If the processor is not bus master (it has completed a bus cycle and BSYNC L is not being asserted by the processor), it grants bus mastership to the requesting device by asserting BDMGO L. The device responds by negating BDMR L and asserting BSACK L.
AP1	BHALT L	Processor halt — when BHALT L is asserted for at least 25 µs, the processor services the halt interrupt and responds by halting normal program execution. External interrupts are ignored but memory refresh interrupts in Q22-bus operations are enabled if W4 on the M7264 and M7264-YA processor modules is removed and DMA request/grant sequences are enabled. The processor executes the ODT microcode, and the console device operation is invoked.
AR1	BREF L	Memory refresh — asserted by a DMA device. This signal forces all dynamic MOS memory units requiring bus refresh signals to be activated for each BSYNC L/BDIN L bus transaction. It is also used as a control signal for block mode DMA. **CAUTION** **The user must avoid multiple DMA data transfers (burst or hot mode) that could delay refresh operation if using DMA refresh. Complete refresh cycles must occur once every 1.6 ms if required.**
AS1	+12 B or +5 B	+12 Vdc or +5 V battery back-up power to keep critical circuits alive during power failures. This signal is not bused to BS1 in all of Digital's backplanes. A jumper is required on all Q22-bus options to open (disconnect) the backup circuit from the bus in systems that use this line at the alternate voltage.
AT1	GND	Ground — system signal ground and dc return.
AU1	PSPARE 1	Spare — not assigned. Customer usage not recommended. Prevents damage when modules are inserted upside down.

Table A–7 (Cont.) Bus Pin Identifiers

Bus Pin	Signal	Definition
AV1	+5 B	+5 V battery power — secondary +5 V power connection. Battery power can be used with certain devices.
BA1	BDCOK H	DC power OK — a power supply generated signal that is asserted when the available dc voltage is sufficient to sustain reliable system operation.
BB1	BPOK H	Power OK — asserted by the power supply 70 ms after BDCOK is negated when ac power drops below the value required to sustain power (approximately 75% of nominal). When negated during processor operation, a power-fail trap sequence is initiated.
BC1	SSPARE4 BDAL18 L (22-bit only)	Special spare in the Q22-bus — not assigned. Bused in 22-bit cable and backplane assemblies. Available for user interconnection.
BD1	SSPARE5 BDAL19 L (22-bit only)	**CAUTION** **These pins may be used by manufacturing as test points in some options.**
BE1	SSPARE6 BDAL20 L	In the Q22-bus, these bused address lines are address lines <21:18>. Currently not used during data time.
BF1	SSPARE7 BDAL21 L	In the Q22-bus, these bused address lines are address lines <21:18>. Currently not used during data time.
BH1	SSPARE8	Special spare — not assigned or bused in Digital's cable and backplane assemblies. Available for user interconnection.
BJ1	GND	Ground — system signal ground and dc return.
BK1 BL1	MSPAREB MSPAREB	Maintenance spare — normally connected together on the backplane at each option location (not a bused connection).
BM1	GND	Ground — system signal ground and dc return.
BN1	BSACK L	This signal is asserted by a DMA device in response to the processor's BDMGO L signal, indicating that the DMA device is bus master.
BP1	BIRQ7 L	Interrupt request priority level 7.
BR1	BEVNT L	External event interrupt request — when asserted, the processor responds by entering a service routine through vector address 1008. A typical use of this signal is as a line time clock (LTC) interrupt.
BS1	+12 B	+12 Vdc battery back-up power (not bused to AS1 in all of Digital's backplanes).
BT1	GND	Ground — system signal ground and dc return.
BU1	PSPARE2	Power spare 2 — not assigned a function and not recommended for use. If a module is using -12 V (on pin AB2), and, if the module is accidentally inserted upside down in the backplane, -12 Vdc appears on pin BU1.
BV1	+5	+5 V power — normal +5 Vdc system power.
AA2	+5	+5 V power — normal +5 Vdc system power.

Table A–7 (Cont.) Bus Pin Identifiers

Bus Pin	Signal	Definition
AB2	-12	-12 V power — -12 Vdc power for (optional) devices requiring this voltage. Each Q22-bus module that requires negative voltages contains an inverter circuit that generates the required voltage(s). Therefore, -12 V power is not required with Digital's options.
AC2	GND	Ground — system signal ground and dc return.
AD2	+12	+12 V power — +12 Vdc system power.
AE2	BDOUT L	Data output — when asserted, BDOUT implies that valid data is available on BDAL<0:15> L and that an output transfer, with respect to the bus master device, is taking place. BDOUT L is deskewed with respect to data on the bus. The slave device responding to the BDOUT L signal must assert BRPLY L to complete the transfer.
AF2	BRPLY L	Reply — BRPLY L is asserted in response to BDIN L or BDOUT L and during IAK transactions. It is generated by a slave device to indicate that it has placed its data on the BDAL bus or that it has accepted output data from the bus.
AH2	BDIN L	Data input — BDIN L is used for two types of bus operations. • When asserted during BSYNC L time, BDIN L implies an input transfer with respect to the current bus master, and requires a response (BRPLY L). BDIN L is asserted when the master device is ready to accept data from the slave device. • When asserted without BSYNC L, it indicates that an interrupt operation is occurring. The master device must deskew input data from BRPLY L.
AJ2	BSYNC L	Synchronize — BSYNC L is asserted by the bus master device to indicate that it has placed an address on BDAL<0:17> L. The transfer is in process until BSYNC L is negated.
AK2	BWTBT L	Write/byte — BWTBT L is used in two ways to control a bus cycle. • It is asserted at the leading edge of BSYNC L to indicate that an output sequence (DATO or DATOB), rather than an input sequence, is to follow. • It is asserted during BDOUT L, in a DATOB bus cycle, for byte addressing.
AL2	BIRQ4 L	Interrupt request priority level 4 — a level 4 device asserts this signal when its interrupt enable and interrupt request flip-flops are set. If the PS word bit 7 is 0, the processor responds by acknowledging the request by asserting BDIN L and BIAKO L.

Table A–7 (Cont.) Bus Pin Identifiers

Bus Pin	Signal	Definition
AM2 AN2	BIAKI L BIAKO L	Interrupt acknowledge — in accordance with interrupt protocol, the processor asserts BIAKO L to acknowledge receipt of an interrupt. The bus transmits this to BIAKI L of the device electrically closest to the processor. This device accepts the interrupt acknowledge under two conditions. • The device requested the bus by asserting BIRQ**n** L (where **n**= 4, 5, 6 or 7) • The device has the highest priority interrupt request on the bus at that time. If these conditions are not met, the device asserts BIAKO L to the next device on the bus. This process continues in a daisy chain fashion until the device with the highest interrupt priority receives the interrupt acknowledge signal.
AP2	BBS7 L	Bank 7 select — the bus master asserts this signal to reference the I/O page (including that part of the page reserved for nonexistent memory). The address in BDAL<0:12> L when BBS7 L is asserted is the address within the I/O page.
AR2 AS2	BDMGI L BDMGO L	Direct memory access grant — the bus arbitrator asserts this signal to grant bus mastership to a requesting device, according to bus mastership protocol. The signal is passed in a daisy-chain from the arbitrator (as BDMGO L) through the bus to BDMGI L of the next priority device (the device electrically closest on the bus). This device accepts the grant only if it requested to be the bus master (by a BDMR L). If not, the device passes the grant (asserts BDMGO L) to the next device on the bus. This process continues until the requesting device acknowledged the grant. **CAUTION** **DMA device transfers must not interfere with the memory refresh cycle.**
AT2	BINIT L	Initialize — this signal is used for system reset. All devices on the bus are to return to a known, initial state; that is, registers are reset to zero, and logic is reset to state 0. Exceptions should be completely documented in programming and engineering specifications for the device.
AU2 AV2	BDAL0 L BDAL1 L	Data/address lines — these two lines are part of the 16-line data/address bus over which address and data information are communicated. Address information is first placed on the bus by the bus master device. The same device then either receives input data from, or outputs data to, the addressed slave device or memory over the same bus lines.
BA2	+5	+5 V power — normal +5 Vdc system power.
BB2	-12	-12 V power (voltage not supplied) — -12 Vdc power for (optional) devices requiring this voltage.
BC2	GND	Ground — system signal ground and dc return.
BD2	+12	+12 V power — +12 V system power.

Table A–7 (Cont.) Bus Pin Identifiers

Bus Pin	Signal	Definition
BE2	BDAL2 L	Data/address lines — these 14 lines are part of the 16-line data/address bus.
BF2	BDAL3 L	
BH2	BDAL4 L	
BJ2	BDAL5 L	
BK2	BDAL6 L	
BL2	BDAL7 L	
BM2	BDAL8 L	
BN2	BDAL9 L	
BP2	BDAL10 L	
BR2	BDAL11 L	
BS2	BDAL12 L	
BT2	BDAL13 L	
BU2	BDAL14 L	
BV2	BDAL15 L	

B
Specifications

B.1 KN220-AA Physical Specifications (pinouts/connectors)

Specs of fingers - A/B (including Q-bus), C/D, connectors, jumpers.

B.2 Dimensions

The KN220-AA CPU, KN220-AA I/O, M7638-AA, and M7637-AA are quad height modules with the following dimensions:

- Height - 10.457 +.015/ -.020 inches
- Length - 8.430 +.010/ -.010 inches
- Width - .375 inches maximum (non-conductive), .343 inches maximum (conductive)

NOTE
Width, as defined for Digital Equipment modules, is the height of components above the surface of the module.

B.3 KN220-AA Connectors

The KN220-AA I/O Module set has seven connector interfaces: two fingers that plug into rows A and B of the backplane (the A/B row fingers), two fingers that plug into rows C and D of the backplane (the C/D row fingers side 1) to connect to the DSSI bus in the BA430 enclosure, a 50-pin connector (J1) to connect to the DSSI bus in the BA213 enclosure, a 100-pin connector (J2) that connects the KN220-AA I/O and CPU modules (the BRIO connector), two fingers that plug into rows C and D of the backplane (the C/D row fingers side 2) for future use, and a 100-pin connector (J3) that connects the KN220-AA to the SCSI interface connector plate, and to the H3602 CPU cover panel via a 40-pin ribbon cable. The ribbon cable is only 40 pins to be compatible with the existing H3602-AC design.

B.3.1 KN220-AA A/B Row Fingers

The KN220-AA A/B row fingers are compatible with the Q22-bus specification. The SRUN(L) signal appears on pin AF1.

B.3.2 KN220-AA C/D Row Fingers and RIO Connector

The signals carried on the KN220-AA C/D row fingers and RIO Connector are as follows (the pins associated with each signal are listed in parentheses):

- *BRIO_ADDR<29:2> H* - Address lines used to transfer address information between the CPU and I/O Modules.
- *BRIO_DATA<31:0> H* - Data lines used to transfer data information between the CPU and I/O modules. On reads, data is valid when BRIO_RDY is asserted. On writes, data is valid for the entire cycle and is qualified by BRIO_DS being asserted.
- *BRIO_MASK<3:0> L* - Byte Mask signals used to indicate which bytes of BRIO_ DATA contain valid information during the bus cycle.
- *BRIO_DS L* - Data strobe signal used to indicate that a valid RIO bus cycle is occurring.
- *BRIO_WR L* - Write signal used to indicate that the current bus cycle is a write cycle.
- *BRIO_RD L* - Read signal used to indicate that the bus master expects the slave device to provide data on BRIO_DATA.
- *BRIO_RDY L* - Tri-state signal driven by the slave device to indicate that data is available on a read, or that data has been latched on a write. Also used in conjunction with BRIO_ERR L to request a retry of the current bus operation (see below).
- *BRIO_ERR L* - Tri-state signal driven by the slave device to indicate an error on the bus. Also used in conjunction with BRIO_RDY L to request a retry of the current bus operation.

NOTE
The slave device requests a retry by asserting BRIO_ERR and then asserting BRIO_RDY 40-50 ns later. The master must terminate the cycle, the system will arbitrate for DMA to allow the slave to perform the operations it needs to to resolve the retry, and then the master will attempt to perform the cycle over again.

- *BCPSIZE<1:0> H* - Driven by the bus master to specify the length of the current transfer. The bits are defined as follows:
  - 00 - Hexword (3 Longwords = 12 bytes)
  - 01 - Longword (4 bytes)
  - 10 - Quadword (2 Longwords = 8 bytes)
  - 11 - Octaword (4 Longwords = 16 bytes)

  NOTE
  The CPU Module only performs longword accesses to the RIO bus. When one of the other modules has been granted bus mastership, these bits indicate the number of cycles that will be performed before the module releases bus mastership.
- *BINT<3:0> L* - Open collector signals asserted by the I/O module to request an interrupt of the CPU.
- *BSELCVAX L* - Indicates which processor is controlling the system. When asserted, the CVAX processor on the I/O module is selected. When negated, the R3000 processor on the CPU module is selected.
- *BRIO_RESET L* - Asserted by the SSC chip on the I/O module to indicate a system reset.
- *BCVHALT L* - Driven by the SSC on the I/O module to request a Halt of the CPU.
- *BCDALDMR L* - CDAL DMA Request signal asserted by the I/O Module to indicate that a device on the CDAL bus is requesting mastership of the RIO bus.
- *BCDALDMG L* - CDAL DMA Grant signal asserted by the CPU Module to grant mastership of the RIO bus to a device on the I/O Module.
- *BIOIAKO L* - Asserted by the I/O Module to indicate that none of the devices with higher interrupt priority than the CQBIC require service during the current interrupt acknowledge cycle.
- *QIAKI L* - Driven by the CPU Module to indicate to the I/O Module that the Interrupt Acknowledge daisy-chain has propogated through to the CQBIC controller.
- *CPUPRESENT L* - Pulled low by the CPU board when it is present in the system.
- *IOPRESENT L* - Pulled low by the I/O board when it is present in the system.
- *VBAT H* - Battery backup voltage.
- *BCPUNXM H* - Asserted by the CPU module when the address on BRIO_ ADDR<29:02>is not present in main memory.
- *SMEMERR H* - Driven by the I/O module to indicate a memory error interrupt has been generated by the CQBIC.
- *NLMR H* - The NLMR signal, when active, indicates that the most recent memory access that was attempted tried to get to a non-existent memory location.

B.3.3 KN220-AA Configuration and Display Connector

The Configuration and Display Connector is a 50-pin connector which features the following pinouts:

Pin	Mnemonic	Meaning
1-17	Reserved	Reserved for I/O Moduled Connection.
18	GND	Ground.
19	DTR H	Data Terminal Ready Whenever the on-boad initialize signal is negated.
20	GND	Ground.
21	TXD L	Console SLU Output from the KN220-AA module.
22	NC	
23	NC	
24 25	RXD L RXD H	Console SLU Differential Inputs to the KN220-AA module. The received serial data connects to RXD L. The signal return connects to RXD H.
26	NC	
27	+5V	Fused +5 Volts
28 29 30	CONBITRATE2 L CONBITRATE1 L CONBITRATE0 L	Console Bit Rate <2:0>. This 3-bit code selects the console terminal baud rate. In the DECsystem 5500 system, CONBITRATE<2:0> is provided by an 8-position switch on the FCC Cutout or cover panel. In the 5500 system, these signals are not connected at the FCC Cutout, but are negated by pull-up resistors.
31 32 33 34	LEDCODE0 L LEDCODE1 L LEDCODE2 L LEDCODE3 L	Display Register Bits 03:00. When asserted, each of these four output signals lights a corresponding external LED. LEDCODE<3:0> are asserted (low) by power up and by the negation of DCOK. They are updated by the boot and diagnostic programs via the Boot and Diagnostic Register. Writing a ONE asserts the corresponding signal; writing a ZERO negates it.

Pin	Mnemonic	Meaning
35	ENBHALT L	Enable Halt. This input signal controls the response to the halt conditions. If HLT ENB is asserted (low), then the CVAX halts and enters the console program if: 1. Program executes a Halt instruction in Kernel Mode 2. Console detects a break character 3. Q22-bus Halt line is asserted The R3000 has no halt instruction and halts are only encountered because of 2 and 3. If HLT ENB is negated, then the Halt line and break character are ignored and the CVAX ROM program responds to a halt instruction by restarting or rebooting the system. HLT ENB can be read by software via the Boot and Diagnostic Register. In the DECsystem 5500 system, HLT ENB originates from a switch on the FCC Cutout or cover panel.
36 37	BDCODE1 L BDCODE0 L	Boot and Diagnostic Code <1:0>. This 2-bit code can be read by software via the Boot and Diagnostic Register (section tbd). The KN220-AA ROM program uses BDCODE <1:0> to select various Boot Device or Diagnostic test parameters at power up and at system restart.
38	VBAT H	Battery Backup Voltage for TOY Clock.
39 40	CPUCODE1 L CPUCODE0	CPUCode<1:0>. This 2-bit code determines which processor will be in control on power up.

CPUCODE <1:0>	Configuration
00	R3000
01	CVAX
10	Reserved
11	Aux Mode (Unsupported)

CPU Code <1:0> can be read by software via the Boot and Diagnostic Register

NOTE
The KN220-AA module provides 4.7K pull-up resistors for the eight input signals (pins 39-40,35-37 and 28-30).

B.4 KN220-AA Environmental and Reliability Specifications

B.5 Operating Conditions

TEMPERATURE

+5 to +60 degrees Celsius (-40 to +140 degrees Fahrenheit) with a rate of change no greater than 20 +-2 degrees Celsius (36 +-4 degrees Fahrenheit) per hour at sea level. The maximum temperature must be derated by 1.8 degrees Celsius per 1000 meters (1 degree Fahrenheit per 1000 feet) above sea level.

HUMIDITY

10 to 95% non-condensing, with a maximum wet bulb temperature of 32 degrees Celsius (90 degrees Fahrenheit) and a minimum dew point temperature of 2 degrees Celsius (36 degrees Fahrenheit).

ALTITUDE

Up to 2,400 meters (8,000 feet) with a rate of change no greater than 300 meters per minute (1000 feet per minute).

AIRFLOW

The airflow required to meet these specifications is 200 lfm.

B.6 Non-Operating Conditions (Less than 60 days)

TEMPERATURE

-40 to +66 degrees Celsius (-40 to +151 degrees Fahrenheit) with a rate of change no greater than 11 +-2 degrees Celsius (20 +-4 degrees Fahrenheit) per hour at sea level. The maximum temperature must be derated by 1.8 degrees Celsius per 1000 meters (1 degree Fahrenheit per 1000 feet) above sea level.

HUMIDITY

Up to 95% non-condensing.

ALTITUDE

Up to 4,900 meters (16,000 feet) with a rate of change no greater than 600 meters per minute (2000 feet per minute).

B.7 Non-Operating Conditions (Greater than 60 days)

TEMPERATURE

+5 to +60 degrees Celsius (-40 to +140 degrees Fahrenheit) with a rate of change no greater than 20 +-2 degrees Celsius (36 +-4 degrees Fahrenheit) per hour at sea level. The maximum temperature must be derated by 1.8 degrees Celsius per 1000 meters (1 degree Fahrenheit per 1000 feet) above sea level.

HUMIDITY

10 to 95% non-condensing, with a maximum wet bulb temperature of 32 degrees Celsius (90 degrees Fahrenheit) and a minimum dew point temperature of 2 degrees Celsius (36 degrees Fahrenheit).

ALTITUDE

Up to 2,400 meters (8,000 feet) with a rate of change no greater than 300 meters per minute (1000 feet per minute).

C
System Block Diagram

C.1 System Block Diagram

Figure C–1 shows the system block diagram.

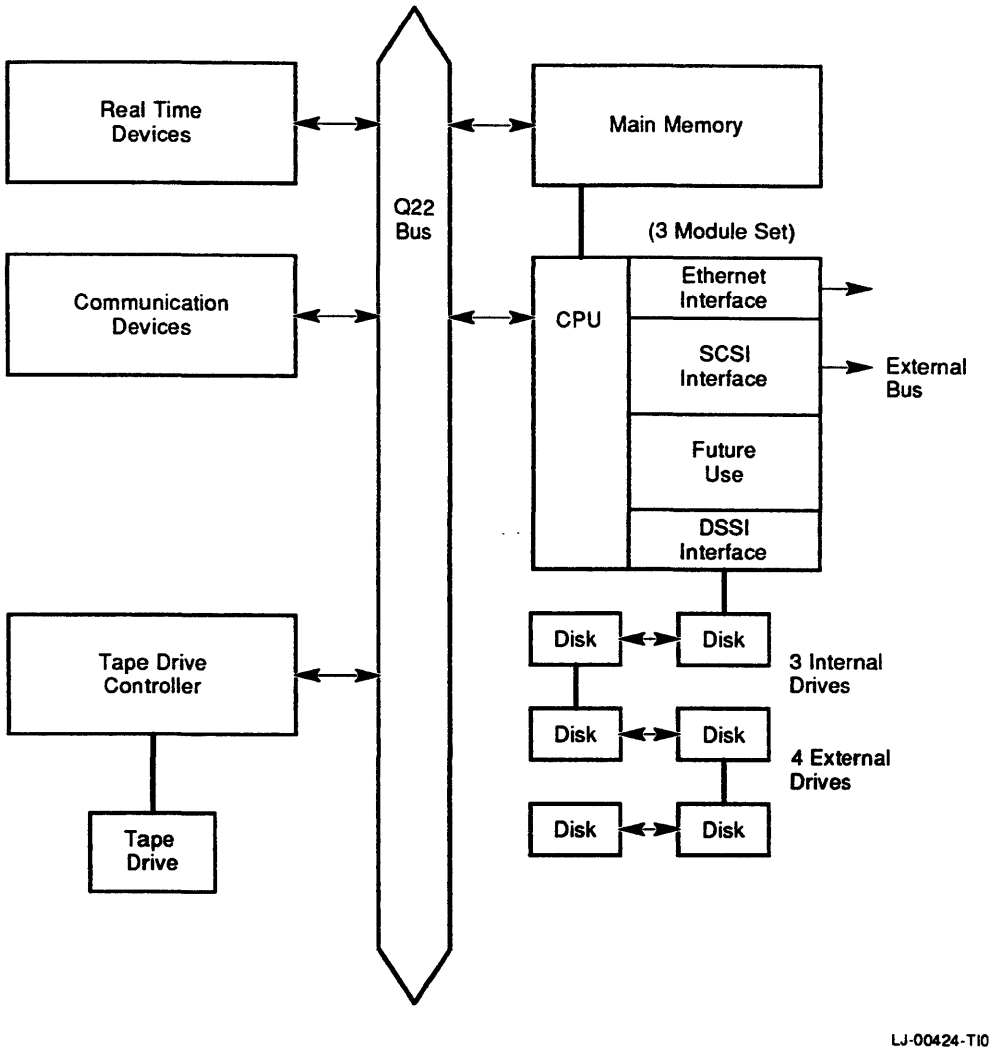


Figure C–1 System Block Diagram

D
Prom Entry Points

D.1 Prom Entry Points

The following entry points are defined in the KN220-AA PROM. This information is provided here only as a general guide as to what functions are available. All routines are called as normal C routines. The normal C conventions are assumed.

D.2 Argvize

```
struct string_list{
char *strptrs[MAXSTRINGS];              /* Vector of string pointers */
char *strbuf[STRINGBYTES];              /* Strings themselves  */
char *strip;                             /* Free ptr in strbuf. */
int strcnt;                             /* Number of strings in strptrs. */ }

int
argvize(str,slp)
char * str;
struct string_list *slp;
```

Breaks a string into tokens and returns the number of tokens. The value returned by *argvize* is *slp.strcnt* which is the argument count (number of tokens), and *slp->strptrs* is the address of vector of strings (argv).

D.3 Atob

```
char *
atob(str,intp)
char *str;
int *intp;
```

Converts ASCII to binary. Accepts all C numeric input formats. Returns pointer to any unconverted substring left after the numeric conversion.

D.4 Autoboot

```
void
autoboot()
```

Performs automatic bootstrap. Unlike **reboot**, the bootstrap is unconditional (although bootpath must be defined). If the autoboot fails, control passes to the console.

D.5 Bevexcept

```
void
bevexcept()
```

Primitive exception handler for exceptions that occur while BEV bit is still set in the system status register. This is not a C routine, and it attempts to enter console IO mode on completion.

D.6 Bevutlb

```
void
bevutlb()
```

Primitive exception handler for exceptions that occur while BEV bit is still set in system status register. This is not a C routine, and it attempts to enter console IO mode on completion.

D.7 Close

```
int
close(fd)
int fd;
```

Closes a file by calling the appropriate fs and device close routines. Returns 0 on a successful close.

D.8 Dumpcmd

```
int
dumpcmd(argc, argv)
int argc;
char **argv;
```

Invokes the console dump command to dump a block of memory. See Section 4.5.4.4.5 for a description of the command. Returns 0 if successful, 1 if not.

D.9 Exec

```
struct promexec_args {
        char    *pa_bootfile;           /* file to boot (only some devices) */
        int     pa_argc;                /* arg count */
        char    **pa_argv;              /* arg vector */
        char    **pa_environ;           /* environment vector */
};

void
exec(pap)
register struct promexec_args *pap;
```

Loads a new program image and transfers control to it.

D.10 Getchar

```
int
getchar()
```

Input a single character from the current console device. Returns a seven-bit ASCII code.

D.11 Getenv

```
>(keep)
char *
getenv(name)
char *name;
```

Returns the value of a console environment variable if it is set, returns NULL otherwise.

D.12 Gets

```
char *
gets(buf)
char *buf;
```

Get a line of input from the console and place in the caller's buffer. Normal line editing functions are performed on input. The address of the buffer is returned.

D.13 Halt

```
void
halt(statebuf)
int *statebuf;
```

Saves the machine state in an area pointed to by the passed argument and enters console I/O mode. This is not a C routine.

D.14 Help

```
struct cmd_table{
  char *name;    /* Command name */
  int (*routine)();  /* Command routine */
  char *usage;    /* Command usage string. */
}

int
help(argc, argv, cmd_table)
int argc;
char **argv;
struct cmd_table *cmd_table;
```

Invokes the console help command. See Section 4.5.4.4.9 for more information. Returns 0 if successful, 1 if not.

D.15 Ioctl

```
int
ioctl(fd, cmd, arg)
int fd;
int cmd;
int arg;
```

Performs a device or file system specific operation on a device or file.

D.16 Longjump

```
typedef int jmp_buf[11];

longjump(jmp_buf, rval)
jmp_buf *jmp_buf;
int rval;
```

Terminates execution in current context and continues from context saved in jmp_buf by a previous setjmp. Execution resumes with rval as the value returned from setjmp.

D.17 Lseek

```
int
lseek(fd, offset, direction);
int fd;
int offset;
int direction;
```

Positions a file into an arbitrary byte position and returns the new position as int.

D.18 Open

```
int
open(filename, flags)
char *filename;
int flags;
```

Opens the indicated filename.

D.19 Parser

```
struct cmd_table{
  char *name;    /* Command name */
  int (*routine)();  /* Command routine */
  char *usage;    /* Command usage string. */
}

int
parser(cmd_table, prompt, search_path)
struct cmd_table *cmd_table;
char *prompt;
char *search_path;
```

Inputs a command line from the console terminal, argvizes it and looks up the first token in the command table. If it is found, invokes the corresponding command with the standard argc/argv argument list. If the command is not found in the command table and search_path is not null, then attempts to perform the command by executing the code in the file specified by the concatenation of search_path and the first token.

D.20 Printenvcmd

```
int
printenvcmd(argc,argv)
int argc;
char *argv;
```

Invoke the console printenv command. Returns 0 if successful, 1 if not.

D.21 Printf

```
void
printf(fmt, va_alist)
char *fmt;
va_dcl
```

Prints formatted values on the current console device.

D.22 Putchar

```
void
putchar(c)
char c;
```

Output a single character to the current console device.

D.23 Puts

```
void
puts(s)
char *s;
```

Output a string to the current console device.

D.24 Range

```
int
range(str, basep, cntp)
char *str;
unsigned *basep;
unsigned *cntp;
```

Parses a console range specification and returns the base address and the count. Returns 0 if the range is of the form address:address, 1 if the range is of the form address#cnt, and minus 1 if the range cannot be parsed.

D.25 Read

```
int
read(fd, buf, cnt)
int fd;
char *buf;
int cnt;
```

Reads data from a file into a buffer. Returns the actual number of bytes read to the caller.

D.26 Reboot

```
void
reboot()
```

Perform automatic test and reboots system if bootmode is "a", otherwise enters console. The effect of this routine is to duplicate the normal console power up/reset boot decision.

D.27 Reinit

```
reinit()
```

Re-initializes the console monitor and enters command mode.

D.28 Reset

```
void
reset()
```

Enters the prom at its entry point, the result is the same as if the reset button had been pressed or the power switch turned on.

D.29 Restart

```
void
restart()
```

Re-enters the console monitor without re-initializing it.

D.30 Setenv

```
int
setenv(name,value)
char *name;
char *value;
```

Sets the value of an environment variable.

D.31 Setenvcmd

```
int
setenvcmd(argc, argv)
int argc;
char **argv;
```

Invokes the console setenv command. Returns 0 if successful, 1 if not.

D.32 Setjmp

```
typedef jmp_buf[11];

int
setjmp(jmp_buf)
jmp_buf jmp_buf;
```

Saves the current register context in jmp_buf and then returns zero. Later, a call to longjump causes the saved context to be restored and execution resumes again, this time with the return value specified by the **longjump** routine. Used to regain control from exceptional conditions.

D.33 Showchar

```
void
showchar(c)
char c;
```

Outputs a single character to the current console device. Displays all printing characters normally. Displays blanks as "\b", form feeds as "\f", newlines as "\n", returns as "\r", tabs as "\t". Displays any other nonprinting character as "\xxx" where xxx is the octal code for the character.

D.34 Strcat

```
char *
strcat(str1, str2)
char *str1;
char *str2;
```

Concatenates string 1 and string 2, returning a pointer to the result.

D.35 Strcmp

```
int
strcmp(str1, str2)
char *str1;
char *str2;
```

Compares string 1 and string 2. Returns 0 if they are the same, a negative value if string 1 is less than string 2, a positive value is string 1 is greater than string 2.

D.36 Strcpy

```
char *
strcpy(str1, str2)
char *str1;
char *str2;
```

Copies string 2 to string 1 and returns a pointer to the first unmodified character in string 1. String 1 must be long enough to contain string 2.

D.37 Strlen

```
int
strlen(str)
char *str;
```

Returns the number of bytes in a string.

D.38 Unsetenvcmd

```
int
unsetenvcmd(argc, argv)
int argc;
char *argv;
```

Deletes an environment variable.

D.39 Write

```
int
write(fd, buf, cnt)
int fd;
char *buf;
int cnt;
```

Writes data from a buffer into a file. Returns the actual number of bytes written to the caller.

E
Supported Devices

E.1 Supported Devices

The KN220-AA firmware supports Operating System bootstrapping from the following devices:

Table E–1 KN220-AA Boot Devices

name	protocol	controllers	units
rf	DSSI [1]	1	0..15
rz	SCSI [2]	1	0..6 (bus id)
ra	MSCP [3]	16	0..31
tz	SCSI [2]	1	0..15
tm	TMSCP	16	0..31
mop	MOP	1	0

[1]One DSSI node with unit numbers from 0 to 15

[2]One SCSI node with bus ids from 0 to 6, only one lun is recognized

[3]Maximum number is for both MSCP and TMSCP devices

The DSSI, SCSI, and NI adapters are in the KN220-AA I/O module. The MSCP devices are attached to the Q22-bus.

NOTE
Only Q22-bus MSCP or TMSCP devices are supported by the console bootstrap procedure. However, the Operating System may support other device types on the Q22-bus.

F

Exit and Maintenance Implementation Guidelines

F.1 Exit and Maintenance Implementation Guidelines

F.2 EXIT command

The CVAX console **EXIT** implementation should abide by the following guidelines:

- Prior to execution of the command all errors must be handled.

- Steps of the EXIT command are:
  1. Q22-Bus Map must be disabled
  2. turn off interrupts to the CVAX
  3. SPR ← 0x40000000

F.3 maint command

The R3000 console command maint should comply with the following guidelines:

- Prior to execution of the command.all errors must be handled.

- Steps of the maint command are:
  1. Q22-Bus Map must be disabled
  2. Turn off interrupts to the R3000
  3. SPR ← 0x80000000

- The CVAX must be able to clean up any outstanding errors after the transition.

- If an interrupt (halt, memerr, pwrfl) occurs while the R3000 was running, the CVAX will be responsible for recovery action.

G
KN220-AA CVAX Diagnostic Processor Instruction Set

G.1 KN220-AA CVAX Diagnostic Processor Instruction Set

The standard notation for operand specifiers is:

<name>.<access type><data type>

where:

1. Name is a suggestive name for the operand in the context of the instruction. It is the capitalized name of a register or block for implied operands.
2. Access type is a letter denoting the operand specifier access type.

   a = address operand
   b = branch displacement
   m = modified operand (both read and written)
   r = read only operand
   v = if not "Rn", same as a, otherwise R[n+1]'R[n]
   w = write only operand
3. Data type is a letter denoting the data type of the operand.

   b = byte
   d = d_floating
   f = f_floating
   g = g_floating
   l = longword
   q = quadword
   v = field (used only in implied operands)
   w = word
   * = multiple longwords (used only in implied operands)
4. Implied operands, that is, locations that are accessed by the instruction, but not specified in an operand, are denoted by curly braces {}.

The abbreviations for condition codes are:

* = conditionally set/cleared
- = not affected
0 = cleared
1 = set

The abbreviations for exceptions are:

rsv = reserved operand fault
iov = integer overflow trap
idvz = integer divide by zero trap
fov = floating overflow fault
fuv = floating underflow fault
fdvz = floating divide by zero fault
dov = decimal overflow trap
ddvz = decimal divide by zero trap
sub = subscript range trap
prv = privileged instruction fault

G.2 Integer Arithmetic And Logical Instructions

Opcode	Instruction	N Z V C	Exceptions
58	ADAWI add.rw, sum.mw	* * * *	iov
80	ADDB2 add.rb, sum.mb	* * * *	iov
C0	ADDL2 add.rl, sum.ml	* * * *	iov
A0	ADDW2 add.rw, sum.mw	* * * *	iov
81	ADDB3 add1.rb, add2.rb, sum.wb	* * * *	iov
C1	ADDL3 add1.rl, add2.rl, sum.wl	* * * *	iov
A1	ADDW3 add1.rw, add2.rw, sum.ww	* * * *	iov
D8	ADWC add.rl, sum.ml	* * * *	iov
78	ASHL cnt.rb, src.rl, dst.wl	* * * 0	iov
79	ASHQ cnt.rb, src.rq, dst.wq	* * * 0	iov
8A	BICB2 mask.rb, dst.mb	* * 0 -	
CA	BICL2 mask.rl, dst.ml	* * 0 -	
AA	BICW2 mask.rw, dst.mw	* * 0 -	
8B	BICB3 mask.rb, src.rb, dst.wb	* * 0 -	
CB	BICL3 mask.rl, src.rl, dst.wl	* * 0 -	
AB	BICW3 mask.rw, src.rw, dst.ww	* * 0 -	
88	BISB2 mask.rb, dst.mb	* * 0 -	
C8	BISL2 mask.rl, dst.ml	* * 0 -	

Opcode	Instruction	N Z V C	Exceptions
A8	BISW2 mask.rw, dst.mw	* * 0 -	
89	BISB3 mask.rb, src.rb, dst.wb	* * 0 -	
C9	BISL3 mask.rl, src.rl, dst.wl	* * 0 -	
A9	BISW3 mask.rw, src.rw, dst.ww	* * 0 -	
93	BITB mask.rb, src.rb	* * 0 -	
D3	BITL mask.rl, src.rl	* * 0 -	
B3	BITW mask.rw, src.rw	* * 0 -	
94	CLRB dst.wb	0 1 0 -	
D4	CLRL{=F} dst.wl	0 1 0 -	
7C	CLRQ{=D=G} dst.wq	0 1 0 -	
B4	CLRW dst.ww	0 1 0 -	
91	CMPB src1.rb, src2.rb	* * 0 *	
D1	CMPL src1.rl, src2.rl	* * 0 *	
B1	CMPW src1.rw, src2.rw	* * 0 *	
98	CVTBL src.rb, dst.wl	* * 0 0	
99	CVTBW src.rb, dst.wl	* * 0 0	
F6	CVTLB src.rl, dst.wb	* * * 0	iov
F7	CVTLW src.rl, dst.ww	* * * 0	iov
33	CVTWB src.rw, dst.wb	* * * 0	iov
32	CVTWL src.rw, dst.wl	* * 0 0	
97	DECB dif.mb	* * * *	iov
D7	DECL dif.ml	* * * *	iov
B7	DECW dif.mw	* * * *	iov
86	DIVB2 divr.rb, quo.mb	* * * 0	iov,idvz
C6	DIVL2 divr.rl, quo.ml	* * * 0	iov,idvz
A6	DIVW2 divr.rw, quo.mw	* * * 0	iov,idvz
87	DIVB3 divr.rb, divd.rb, quo.wb	* * * 0	iov,idvz
C7	DIVL3 divr.rl, divd.rl, quo.wl	* * * 0	iov,idvz
A7	DIVW3 divr.rw, divd.rw, quo.ww	* * * 0	iov,idvz

Opcode	Instruction	N Z V C	Exceptions
7B	EDIV divr.rl, divd.rq, quo.wl, rem.wl	* * * 0	iov,idvz
7A	EMUL mulr.rl, muld.rl, add.rl, prod.wq	* * 0 0	
96	INCB sum.mb	* * * *	iov
D6	INCL sum.ml	* * * *	iov
B6	INCW sum.mw	* * * *	iov
92	MCOMB src.rb, dst.wb	* * 0 -	
D2	MCOML src.rl, dst.wl	* * 0 -	
B2	MCOMW src.rw, dst.ww	* * 0 -	
8E	MNEGB src.rb, dst.wb	* * * *	iov
CE	MNEGL src.rl, dst.wl	* * * *	iov
AE	MNEGW src.rw, dst.ww	* * * *	iov
90	MOVB src.rb, dst.wb	* * 0 -	
D0	MOVL src.rl, dst.wl	* * 0 -	
7D	MOVQ src.rq, dst.wq	* * 0 -	
B0	MOVW src.rw, dst.ww	* * 0 -	
9A	MOVZBW src.rb, dst.wb	0 * 0 -	
9B	MOVZBL src.rb, dst.wl	0 * 0 -	
3C	MOVZWL src.rw, dst.ww	0 * 0 -	
84	MULB2 mulr.rb, prod.mb	* * * 0	iov
C4	MULL2 mulr.rl, prod.ml	* * * 0	iov
A4	MULW2 mulr.rw, prod.mw	* * * 0	iov
85	MULB3 mulr.rb, muld.rb, prod.wb	* * * 0	iov
C5	MULL3 mulr.rl, muld.rl, prod.wl	* * * 0	iov
A5	MULW3 mulr.rw, muld.rw, prod.ww	* * * 0	iov
DD	PUSHL src.rl, {-(SP).wl}	* * 0 -	
9C	ROTL cnt.rb, src.rl, dst.wl	* * 0 -	

Opcode	Instruction	N Z V C	Exceptions
D9	SBWC sub.rl, dif.ml	* * * *	iov
82	SUBB2 sub.rb, dif.mb	* * * *	iov
C2	SUBL2 sub.rl, dif.ml	* * * *	iov
A2	SUBW2 sub.rw, dif.mw	* * * *	iov
83	SUBB3 sub.rb, min.rb, dif.wb	* * * *	iov
C3	SUBL3 sub.rl, min.rl, dif.wl	* * * *	iov
A3	SUBW3 sub.rw, min.rw, dif.ww	* * * *	iov
95	TSTB src.rb	* * 0 0	
D5	TSTL src.rl	* * 0 0	
B5	TSTW src.rw	* * 0 0	
8C	XORB2 mask.rb, dst.mb	* * 0 -	
CC	XORL2 mask.rl, dst.ml	* * 0 -	
AC	XORW2 mask.rw, dst.mw	* * 0 -	
8D	XORB3 mask.rb, src.rb, dst.wb	* * 0 -	
CD	XORL3 mask.rl, src.rl, dst.wl	* * 0 -	
AD	XORW3 mask.rw, src.rw, dst.ww	* * 0 -	

Address Instructions

Opcode	Instruction	N Z V C	Exceptions
9E	MOVAB src.ab, dst.wl	* * 0 -	
DE	MOVAL{=F} src.al, dst.wl	* * 0 -	
7E	MOVAQ{=D=G} src.aq, dst.wl	* * 0 -	
3E	MOVAW src.aw, dst.wl	* * 0 -	
9F	PUSHAB src.ab, {-(SP).wl}	* * 0 -	
DF	PUSHAL{=F} src.al, {-(SP).wl}	* * 0 -	
7F	PUSHAQ{=D=G} src.aq, {-(SP).wl}	* * 0 -	
3F	PUSHAW src.aw, {-(SP).wl}	* * 0 -	

Variable Length Bit Field Instructions

Opcode	Instruction	N Z V C	Exceptions
EC	CMPV pos.rl, size.rb, base.vb, {field.rv}, src.rl	* * 0 *	rsv
ED	CMPZV pos.rl, size.rb, base.vb, {field.rv}, src.rl	* * 0 *	rsv
EE	EXTV pos.rl, size.rb, base.vb, {field.rv}, dst.wl	* * 0 -	rsv
EF	EXTZV pos.rl, size.rb, base.vb, {field.rv}, dst.wl	* * 0 -	rsv
F0	INSV src.rl, pos.rl, size.rb, base.vb, {field.wv}	- - - -	rsv
EB	FFC startpos.rl, size.rb, base.vb, {field.rv}, findpos.wl	0 * 0 0	rsv
EA	FFS startpos.rl, size.rb, base.vb, {field.rv}, findpos.wl	0 * 0 0	rsv

Control Instructions

Opcode	Instruction	N Z V C	Exceptions
9D	ACBB limit.rb, add.rb, index.mb, displ.bw	* * * -	iov
F1	ACBL limit.rl, add.rl, index.ml, displ.bw	* * * -	iov
3D	ACBW limit.rw, add.rw, index.mw, displ.bw	* * * -	iov
F3	AOBLEQ limit.rl, index.ml, displ.bb	* * * -	iov
F2	AOBLSS limit.rl, index.ml, displ.bb	* * * -	iov
1E	BCC{=BGEQU} displ.bb	- - - -	
1F	BCS{=BLSSU} displ.bb	- - - -	
13	BEQL{=BEQLU} displ.bb	- - - -	
18	BGEQ displ.bb	- - - -	
14	BGTR displ.bb	- - - -	
1A	BGTRU displ.bb	- - - -	
15	BLEQ displ.bb	- - - -	
1B	BLEQU displ.bb	- - - -	
19	BLSS displ.bb	- - - -	
12	BNEQ{=BNEQU} displ.bb	- - - -	
1C	BVC displ.bb	- - - -	
1D	BVS displ.bb	- - - -	

Opcode	Instruction	N Z V C	Exceptions
E1	BBC pos.rl, base.vb, displ.bb, {field.rv}	- - - -	rsv
E0	BBS pos.rl, base.vb, displ.bb, {field.rv}	- - - -	rsv
E5	BBCC pos.rl, base.vb, displ.bb, {field.mv}	- - - -	rsv
E3	BBCS pos.rl, base.vb, displ.bb, {field.mv}	- - - -	rsv
E4	BBSC pos.rl, base.vb, displ.bb, {field.mv}	- - - -	rsv
E2	BBSS pos.rl, base.vb, displ.bb, {field.mv}	- - - -	rsv
E7	BBCCI pos.rl, base.vb, displ.bb, {field.mv}	- - - -	rsv
E6	BBSSI pos.rl, base.vb, displ.bb, {field.mv}	- - - -	rsv
E9	BLBC src.rl, displ.bb	- - - -	
E8	BLBS src.rl, displ.bb	- - - -	
11	BRB displ.bb	- - - -	
31	BRW displ.bw	- - - -	
10	BSBB displ.bb, {-(SP).wl}	- - - -	
30	BSBW displ.bw, {-(SP).wl}	- - - -	
8F	CASEB selector.rb, base.rb, limit.rb, displ.bw-list	* * 0 *	
CF	CASEL selector.rl, base.rl, limit.rl, displ.bw-list	* * 0 *	
AF	CASEW selector.rw, base.rw, limit.rw, displ.bw-list	* * 0 *	
17	JMP dst.ab	- - - -	
16	JSB dst.ab, {-(SP).wl}	- - - -	

Opcode	Instruction	N Z V C	Exceptions
05	RSB {(SP)+.rl}	- - - -	
F4	SOBGEQ index.ml, displ.bb	* * * -	iov
F5	SOBGTR index.ml, displ.bb	* * * -	iov

Procedure Call Instructions

Opcode	Instruction	N Z V C	Exceptions
FA	CALLG arglist.ab, dst.ab, {-(SP).w*}	0 0 0 0	rsv
FB	CALLS numarg.rl, dst.ab, {-(SP).w*}	0 0 0 0	rsv
04	RET {(SP)+.r*}	* * * *	rsv

Miscellaneous Instructions

Opcode	Instruction	N Z V C	Exceptions
B9	BICPSW mask.rw	* * * *	rsv
B8	BISPSW mask.rw	* * * *	rsv
03	BPT {-(KSP).w*}	0 0 0 0	
00	HALT {-(KSP).w*}	- - - -	prv
0A	INDEX subscript.rl, low.rl, high.rl, size.rl, indexin.rl, indexout.wl	* * 0 0	sub
DC	MOVPSL dst.wl	- - - -	
01	NOP	- - - -	
BA	POPR mask.rw, {(SP)+.r*}	- - - -	
BB	PUSHR mask.rw, {-(SP).w*}	- - - -	
FC	XFC {unspecified operands}	0 0 0 0	

Queue Instructions

Opcode	Instruction	N Z V C	Exceptions
5C	INSQHI entry.ab, header.aq	0 * 0 *	rsv
5D	INSQTI entry.ab, header.aq	0 * 0 *	rsv
0E	INSQUE entry.ab, pred.ab	* * 0 *	
5E	REMQHI header.aq, addr.wl	0 * * *	rsv
5F	REMQTI header.aq, addr.wl	0 * * *	rsv
0F	REMQUE entry.ab, addr.wl	* * * *	

Character String Instructions

Opcode	Instruction	N Z V C	Exceptions
29	CMPC3 len.rw, src1addr.ab, src2addr.ab,	* * 0 *	
2D	CMPC5 src1len.rw, src1addr.ab, fill.rb, src2len.rw, src2addr.ab	* * 0 *	
3A	LOCC char.rb, len.rw, addr.ab	0 * 0 0	
28	MOVC3 len.rw, srcaddr.ab, dstaddr.ab, {R0-5.wl}	0 1 0 0	
2C	MOVC5 srclen.rw, srcaddr.ab, fill.rb, dstlen.rw, dstaddr.ab, {R0-5.wl}	* * 0 *	
2A	SCANC len.rw, addr.ab, tbladdr.ab, mask.rb	0 * 0 0	
2B	SPANC len.rw, addr.ab, tbladdr.ab, mask.rb	0 * 0 0	
3B	SKPC char.rb, len.rw, addr.ab,	0 * 0 0	

Operating System Support Instructions

Opcode	Instruction	N Z V C	Exceptions
BD	CHME param.rw, {-(ySP).w*}	0 0 0 0	
BC	CHMK param.rw, {-(ySP).w*}	0 0 0 0	
BE	CHMS param.rw, {-(ySP).w*}	0 0 0 0	
BF	CHMU param.rw, {-(ySP).w*} Where y=MINU(x, PSL)	0 0 0 0	
06	LDPCTX {PCB.r*, -(KSP).w*}	- - - -	rsv, prv
DB	MFPR procreg.rl, dst.wl	* * 0 -	rsv, prv
DA	MTPR src.rl, procreg.rl	* * 0 -	rsv, prv
0C	PROBER mode.rb, len.rw, base.ab	0 * 0 -	
0D	PROBEW mode.rb, len.rw, base.ab	0 * 0 -	
02	REI {(SP)+.r*}	* * * *	rsv
07	SVPCTX {(SP)+.r*, PCB.w*}	- - - -	prv

G.3 Microcode-Assisted Emulated Instructions

The KN220-AA CPU provides microcode assistance for the macrocode emulation of these instructions. The CPU processes the operand specifiers, creates a standard argument list, and invokes an emulation routine to perform emulation.

Opcode	Instruction	N Z V C	Exceptions
20	ADDP4 addlen.rw, addaddr.ab, sumlen.rw, sumaddr.ab	* * * 0	rsv, dov
21	ADDP6 add1len.rw, add1addr.ab, add2len.rw, add2addr.ab, sumlen.rw, sumaddr.ab	* * * 0	rsv, dov
F8	ASHP cnt.rb, srclen.rw, srcaddr.ab, round.rb, dstlen.rw, dstaddr.ab	* * * 0	rsv, dov
29	CMPC3 len.rw, src1addr.ab, src2addr.ab	* * 0 *	
2D	CMPC5 src1len.rw, src1addr.ab, fill.rb, src2len.rw, src2addr.ab	* * 0 *	
35	CMPP3 len.rw, src1addr.ab, src2addr.ab	* * 0 0	
37	CMPP4 src1len.rw, src1addr.ab, src2len.rw, src2addr.ab	* * 0 0	
0B	CRC tbl.ab, inicrc.rl, strlen.rw, stream.ab	* * 0 0	
F9	CVTLP src.rl, dstlen.rw, dstaddr.ab	* * * 0	rsv, dov
36	CVTPL srclen.rw, srcaddr.ab, dst.wl	* * * 0	rsv, iov
08	CVTPS srclen.rw, srcaddr.ab, dstlen.rw, dstaddr.ab	* * * 0	rsv, dov
09	CVTSP srclen.rw, srcaddr.ab, dstlen.rw, dstaddr.ab	* * * 0	rsv, dov
24	CVTPT srclen.rw, srcaddr.ab, tbladdr.ab, dstlen.rw, dstaddr.ab	* * * 0	rsv, dov
26	CVTTP srclen.rw, srcaddr.ab, tbladdr.ab, dstlen.rw, dstaddr.ab	* * * 0	rsv, dov
27	DIVP divrlen.rw, divraddr.ab, divdlen.rw, divdaddr.ab, quolen.rw, quoaddr.ab	* * * 0	rsv,dov,ddvz
38	EDITPC srclen.rw, srcaddr.ab, pattern.ab, dstaddr.ab	* * * *	rsv, dov
3A	LOCC char.rb, len.rw, addr.ab	0 * 0 0	
39	MATCHC objlen.rw, objaddr.ab, srclen.rw, srcaddr.ab	0 * 0 0	
34	MOVP len.rw, srcaddr.ab, dstaddr.ab	* * 0 0	
2E	MOVTC srclen.rw, srcaddr.ab, fill.rb, tbladdr.ab, dstlen.rw, dstaddr.ab	* * 0 *	

Opcode	Instruction	N Z V C	Exceptions
2F	MOVTUC srclen.rw, srcaddr.ab, esc.rb, tbladdr.ab, dstlen.rw, dstaddr.ab	* * * *	
25	MULP mulrlen.rw, mulraddr.ab, muldlen.rw, muldaddr.ab, prodlen.rw, prodaddr.ab	* * * 0	rsv, dov
2A	SCANC len.rw, addr.ab, tbladdr.ab, mask.rb	0 * 0 0	
3B	SKPC char.rb, len.rw, addr.ab	0 * 0 0	
2B	SPANC len.rw, addr.ab, tbladdr.ab, mask.rb	0 * 0 0	
22	SUBP4 sublen.rw, subaddr.ab, diflen.rw, difaddr.ab	* * * 0	rsv, dov
23	SUBP6 sublen.rw, subaddr.ab, minlen.rw, minaddr.ab, diflen.rw, difaddr.ab	* * * 0	rsv, dov

H
Address Assignments

H.1 KN220-AA R3000 Address Assignments

H.2 KN220-AA R3000 Physical Address Space Map

Memory Space (up to 256 Mbytes)	0000 0000 - 0FFF FFFF	
Reserved Q22-bus I/O Space	1000 0000 - 1000 0007	\
Q22-bus Floating Address Space	1000 0008 - 1000 07FF	\ Local
User Reserved Q22-bus I/O Space	1000 0800 - 1000 0FFF	\ Q22-bus
Reserved Q22-bus I/O Space	1000 1000 -	
1000 1F3F / I/O Space		
Interprocessor Comm Reg	1000 1F40	/
Reserved Q22-bus I/O Space	1000 1F48 - 1000 1FFF	/
Reserved I/O Module Address Space	1000 2000 - 1000 7FFF	
SGEC Internal Registers	1000 8000 - 1000 803C	
Reserved (copies of SGEC Regs)	1000 8040 - 1001 FFFF	
Reserved I/O Module Address Space	1002 0000 - 1003 FFFF	
Two Copies of CVAX ROM	1004 0000 - 1007 FFFF	
Q22 System Configuration Register	1008 0000	
Q22 System Error Register	1008 0004	
Q22 Master Error Address Register	1008 0008	
Q22 Slave Error Address Register	1008 000C	
Q22-bus Map Base Register	1008 0010	
Reserved	1008 0014 - 1008 3FFF	
Interrupt Status Register	1008 4000	
Boot and Diagnostic Register	1008 4004	
Select Processor Register	1008 4008	
Interval Timer Register	1008 4010	
Reserved I/O Module Address Space	1008 4014 - 1008 7FFF	
Q22-bus Map Registers	1008 8000 - 1008 FFFF	
Reserved I/O Module Address Space	1009 0000 - 1009 FFFF	
DSSI Buffer RAM	1010 0000 - 1011 FFFF	
NI Station Address ROM	1012 0000 - 1012 007C	
Reserved I/O Module Address Space	1012 0080 - 1013 FFFF	
SSC Base Address Register	1014 0000	
SSC Configuration Register	1014 0010	
CDAL Bus Timeout Control Register	1014 0020	
Diagnostic LED Register	1014 0030	
Reserved I/O Module Address Space	1014 0034 - 1014 0068	
Time of Year Register	1014 006C	
Reserved	1014 0070 - 1014 007C	
CVAX Console Rcvr Control/Status	1014 0080	
CVAX Console Rcvr Data Buffer	1014 0084	
CVAX Console Xmit Control/Status	1014 0088	
CVAX Console Xmit Data Buffer	1014 008C	
Reserved	1014 0090 - 1014 00DB	

	I/O System Reset Register	1014 00DC
	Reserved	1014 00E0
	Rom Data Register	1014 00F0
	Bus Timeout Counter	1014 00F4
	Interval Timer	1014 00F8
	Reserved	1014 00FC - 1014 00FF
	Timer 0 Control Register	1014 0100
	Timer 0 Interval Register	1014 0104
	Timer 0 Next Interval Register	1014 0108
	Timer 0 Interrupt Vector	1014 010C
	Timer 1 Control Register	1014 0110
	Timer 1 Interval Register	1014 0114
	Timer 1 Next Interval Register	1014 0118
	Timer 1 Interrupt Vector	1014 011C
	MSIDB Address Decode Match Register	1014 0130
	MSIDB Address Decode Mask Register	1014 0134
	LIOD Address Decode Match Register	1014 0140
	LIOD Address Decode Mask Register	1014 0144
	Reserved	1014 0148 - 1014 033F
	CVAX Battery Backed-Up RAM	1014 0400 - 1014 07FF
	Reserved I/O Module Address Space	1014 0800 - 1015 FFFF
	SII Internal Registers	1016 0000 - 1016 007C
	Reserved I/O Module Address Space	1016 0080 - 1017 FFFF
	Reserved	1018 0000 - 13FF FFFF
	Local Q22-bus Memory Space	1400 0000 - 143F FFFF
	Reserved (4 copies Local Q22 Mem)	1440 0000 - 14FF FFFF
	Reserved	1500 0000 - 1600 004F
*	Vector Read Register 0	1600 0050
*	Vector Read Register 1	1600 0054
*	Vector Read Register 2	1600 0058
*	Vector Read Register 3	1600 005C
*	Vector Write Address	1610 005C
	Reserved	1600 0060 - 16FF FFFF
	I/O Presence Register	1700 0000 - 1703 FFFF
	Memory Error Syndrome Register	1704 0000 - 1707 FFFF
	Memory Error Address Register	1708 0000 - 170B FFFF
	Memory ID Register	170C 0000 - 170F FFFF
	SCSI 53C94 Registers	1710 0000 - 1710 0028
	Reserved	1710 002C - 1710 003C
	Reserved (copies of SCSI Regs)	1710 0040 - 1713 FFFF
	SCSI DMA Register	1714 0000
	Reserved (copies of SCSI DMA Reg)	1714 0004 - 1717 FFFF
	SCSI Buffer RAM	1718 0000 - 1719 FFFF
	Reserved (copy of SCSI RAM)	171A 0000 - 171B FFFF
	Reserved SCSI address space	171C 0000 - 171F FFFF
	R3000 LED Register	1720 0000 - 1723 FFFF
	Reserved I/O Module Address Space	1724 0000 - 17FF FFFF
	R3000 Non-Volatile RAM	1800 0000 - 1800 7FFF
	Reserved (copies of NVRAM)	1800 8000 - 1803 FFFF
	Reserved I/O Module Address Space	1804 0000 - 180F FFFF
	R3000 UART Registers	1810 0000 - 1810 003C
	Reserved (copies of UART Regs)	1810 0040 - 1813 FFFF
	Reserved I/O Module Address Space	1814 0000 - 18FF FFFF
	Reserved I/O Module Address Space	1D00 0000 - 1FBF FFFF
	R3000 ROM	1FC0 0000 - 1FC3 FFFF
	Reserved I/O Module Address Space	1FC4 0000 - 1FFF FFFF
	Reserved	2000 0000 - 2FFF FFFF
	Memory Space (up to 256 Mbytes)	3000 0000 - 3FFF FFFF
	Reserved	4000 0000 - FFFF FFFF

"*" - only accessible from Mips processor.

H.3 M7638-AA R3000 Physical I/O Address Space Map

```
        Reserved Q22-bus I/O Space          1000 0000 - 1000 0007 \
        Q22-bus Floating Address Space      1000 0008 - 1000 07FF  \  Local
        User Reserved Q22-bus I/O Space     1000 0800 - 1000 0FFF   \ Q22-bus
        Reserved Q22-bus I/O Space          1000 1000 -
1000 1F3F   / I/O Space
        Interprocessor Comm Reg             1000 1F40              /
        Reserved Q22-bus I/O Space          1000 1F48 - 1000 1FFF /
        Reserved I/O Module Address Space   1000 2000 - 1000 7FFF
        SGEC CSR0 - Vector, IPL, Mode       1000 8000
        SGEC CSR1 - Transmit Poll Demand    1000 8004
        SGEC CSR2 - Receive Poll Demand     1000 8008
        SGEC CSR3 - Receive Descriptor List 1000 800C
        SGEC CSR4 - Xmit Descriptor List    1000 8010
        SGEC CSR5 - Status Register         1000 8014
        SGEC CSR6 - Command and Mode Reg.   1000 8018
        SGEC CSR7 - System Base Register    1000 801C
        SGEC CSR8 - Reserved Register       1000 8020
        SGEC CSR9 - Watchdog Timers         1000 8024
        SGEC CSR10 - Revision Number
                     and Missed Frame Count 1000 8028
        SGEC Boot Message Registers         1000 802C - 1000 8034
        SGEC Diagnostic Registers           1000 8038 - 1000 803C
        Reserved (copies of SGEC Registers) 1000 8040 - 1001 FFFF
        Reserved I/O Module Address Space   1002 0000 - 1003 FFFF
        Two Copies of CVAX ROM              1004 0000 - 1007 FFFF
        Q22 System Configuration Register   1008 0000
        Q22 System Error Register           1008 0004
        Q22 Master Error Address Register   1008 0008
        Q22 Slave Error Address Register    1008 000C
        Q22-bus Map Base Register           1008 0010
        Reserved                            1008 0014 - 1008 3FFF
        Interrupt Status Register           1008 4000 (mips unique)
        Boot and Diagnostic Register        1008 4004
        Select Processor Register           1008 4008 (mips unique)
        Interval Timer Register             1008 4010
        Reserved I/O Module Address Space   1008 4014 - 1008 7FFF
        Q22-bus Map Registers               1008 8000 - 1008 FFFF
        Reserved I/O Module Address Space   1009 0000 - 1009 FFFF
        DSSI Buffer RAM                     1010 0000 - 1011 FFFF
        NI Station Address ROM              1012 0000 - 1012 007C
        Reserved I/O Module Address Space   1012 0080 - 1013 FFFF
        SSC Base Address Register           1014 0000
        SSC Configuration Register          1014 0010
        CDAL Bus Timeout Control Register   1014 0020
        Diagnostic LED Register             1014 0030
        Reserved I/O Module Address Space   1014 0034 - 1014 0068
        Time of Year Register               1014 006C
               Reserved                                1014 0070 - 1014 007C
        CVAX Console Rcvr Control/Status    1014 0080
        CVAX Console Rcvr Data Buffer       1014 0084
        CVAX Console Xmit Control/Status    1014 0088
        CVAX Console Xmit Data Buffer       1014 008C
        Reserved                            1014 0090 - 1014 00DB
        I/O System Reset Register           1014 00DC
        Reserved                            1014 00E0
        Rom Data Register                   1014 00F0
        Bus Timeout Counter                 1014 00F4
        Interval Timer                      1014 00F8
        Reserved                            1014 00FC - 1014 00FF
        Timer 0 Control Register            1014 0100
        Timer 0 Interval Register           1014 0104
        Timer 0 Next Interval Register      1014 0108
        Timer 0 Interrupt Vector            1014 010C
```

```
        Timer 1 Control Register              1014 0110
        Timer 1 Interval Register             1014 0114
        Timer 1 Next Interval Register        1014 0118
        Timer 1 Interrupt Vector              1014 011C
        DSSIDB Address Decode Match Reg       1014 0130
        DSSIDB Address Decode Mask Register   1014 0134
        LIOD Address Decode Match Register    1014 0140
        LIOD Address Decode Mask Register     1014 0144
        Reserved                              1014 0148 - 1014 033F
        CVAX Battery Backed-Up RAM            1014 0400 - 1014 07FF
        Reserved I/O Module Address Space     1014 0800 - 1015 FFFF
        DSSI Diagnostic Register 0            1016 0000
        DSSI Diagnostic Register 1            1016 0004
        DSSI Diagnostic Register 2            1016 0008
        DSSI Control and Status Register      1016 000C
        DSSI ID Register                      1016 0010
        Reserved DSSI Register                1016 0014
        Reserved DSSI Register                1016 0018
        DSSI Timeout Register                 1016 001C
        Reserved DSSI Register                1016 0020
        Reserved DSSI Register                1016 0024
        Reserved DSSI Register                1016 0028
        Reserved DSSI Register                1016 002C
        Reserved DSSI Register                1016 0030
        Reserved DSSI Register                1016 0034
        DSSI Short Target List Pointer        1016 0038
        DSSI Long Target List Pointer         1016 003C
        DSSI Initiator List Pointer           1016 0040
        DSSI DSSI Control REgister            1016 0044
        DSSI DSSI Status REgister             1016 0048
        Reserved DSSI Register                1016 004C
        Reserved DSSI Register                1016 0050
        DSSI Diagnostic Control Register      1016 0054
        DSSI Clock Control Register           1016 0058
        DSSI Internal State Register 0        1016 005C
        DSSI Internal State Register 1        1016 0060
        DSSI Internal State Register 2        1016 0064
                DSSI Internal State Register 2      1016 0068
        Reserved DSSI Register                1016 006C
        Reserved DSSI Register                1016 0070
        Reserved DSSI Register                1016 0074
        Reserved DSSI Register                1016 0078
        Reserved DSSI Register                1016 007C
        Reserved I/O Module Address Space     1016 0080 - 1017 FFFF
        Reserved                              1018 0000 - 13FF FFFF
        Local Q22-bus Memory Space            1400 0000 - 143F FFFF
        Reserved (4 copies Local Q22 Mem)     1440 0000 - 14FF FFFF
        Reserved                              1500 0000 - 1600 004F
    **  Vector Read Register 0                1600 0050 (mips unique)
    **  Vector Read Register 1                1600 0054 (mips unique)
    **  Vector Read Register 2                1600 0058 (mips unique)
    **  Vector Read Register 3                1600 005C (mips unique)
        Reserved                              1600 0060 - 1610 0058
    **  Vector Write Register                 1610 005C (mips unique)
        Reserved                              1610 0060 - 16FF FFFF
     *  I/O Presence Register                 1700 0000 -
1703 FFFF (mips unique)
     *  Memory Error Syndrome Register        1704 0000 -
1707 FFFF (mips unique)
     *  Memory Error Address Register         1708 0000 -
170B FFFF (mips unique)
     *  Memory ID Register                    170C 0000 -
```

```
170F FFFF (mips unique)
        53C94 Transfer Counter Lo Register  1710 0000 (Read Only)
        53C94 Transfer Count Lo Register    1710 0000 (Write Only)
        53C94 Transfer Counter Hi Register  1710 0004 (Read Only)
        53C94 Transfer Count Hi Register    1710 0004 (Write Only)
        53C94 FIFO Register                 1710 0008
        53C94 Command Register              1710 000C
        53C94 Status Register               1710 0010 (Read Only)
        53C94 Select Bus ID Register        1710 0010 (Write Only)
        53C94 Interrupt Status Register     1710 0014 (Read Only)
        53C94 Select Timeout Register       1710 0014 (Write Only)
        53C94 Sequence Step Register        1710 0018 (Read Only)
        53C94 Sync Transfer Period Register 1710 0018 (Write Only)
        53C94 FIFO Flags Register           1710 001C (Read Only)
        53C94 Synchronous Offset Register   1710 001C (Write Only)
        53C94 Configuration Register        1710 0020
        53C94 Clock Conversion Register     1710 0024 (Write Only)
        53C94 Test Register                 1710 0028 (Write Only)
        Reserved                            1710 002C - 1710 003C
        Reserved (copies of SCSI Regs)      1710 0040 - 1713 FFFF
        SCSI DMA Register                   1714 0000 (Write Only)
        Reserved (copies of SCSI DMA Reg)   1714 0004 - 1717 FFFF
        SCSI Buffer RAM                     1718 0000 - 1719 FFFF
        Reserved (copy of SCSI RAM)         171A 0000 - 171B FFFF
        Reserved SCSI address space         171C 0000 - 171F FFFF
    *   R3000 LED Register                  1720 0000 - 1723 FFFF
        Reserved I/O Module Address Space   1724 0000 - 17FF FFFF
    *   R3000 Non-Volatile RAM              1800 0000 - 1807 FFFF
        Reserved I/O Module Address Space   1808 0000 - 180F FFFF
    *   R3000 UART Registers                1810 0000 - 1810 003C
    *   Reserved (copies of UART Regs)      1810 0040 - 1813 FFFF
        Reserved I/O Module Address Space   1814 0000 - 18FF FFFF
        Reserved I/O Module Address Space   1910 0000 - 191F FFFF
        Reserved I/O Module Address Space   1940 0000 - 1FBF FFFF
    *   R3000 ROM                           1FC0 0000 - 1FC3 FFFF
        Reserved I/O Module Address Space   1FC4 0000 - 1FFF FFFF
        Reserved                            2000 0000 - 2FFF FFFF
```

H.4 M7638-AA CVAX Physical I/O Address Space Map

```
Reserved Q22-bus I/O Space          2000 0000 - 2000 0007 \
Q22-bus Floating Address Space      2000 0008 - 2000 07FF  \  Local
User Reserved Q22-bus I/O Space     2000 0800 - 2000 0FFF   \ Q22-bus
Reserved Q22-bus I/O Space          2000 1000 - 2000 1F3F   / I/O Space
Interprocessor Comm Reg             2000 1F40              /
Reserved Q22-bus I/O Space          2000 1F48 - 2000 1FFF /
Reserved I/O Module Address Space   2000 2000 - 2000 7FFF
SGEC CSR0 - Vector, IPL, Mode       2000 8000
SGEC CSR1 - Transmit Poll Demand    2000 8004
SGEC CSR2 - Receive Poll Demand     2000 8008
SGEC CSR3 - Receive Descriptor List 2000 800C
SGEC CSR4 - Xmit Descriptor List    2000 8010
SGEC CSR5 - Status Register         2000 8014
SGEC CSR6 - Command and Mode Reg.   2000 8018
SGEC CSR7 - System Base Register    2000 801C
SGEC CSR8 - Reserved Register       2000 8020
SGEC CSR9 - Watchdog Timers         2000 8024
SGEC CSR10 - Revision Number
             and Missed Frame Count 2000 8028
SGEC Boot Message Registers         2000 802C - 2000 8034
SGEC Diagnostic Registers           2000 8038 - 2000 803C
Reserved (copies of SGEC Registers) 2000 8040 - 2001 FFFF
Reserved I/O Module Address Space   2002 0000 - 2003 FFFF
Two Copies of CVAX ROM              2004 0000 - 2007 FFFF
Q22 System Configuration Register   2008 0000
Q22 System Error Register           2008 0004
Q22 Master Error Address Register   2008 0008
Q22 Slave Error Address Register    2008 000C
Q22-bus Map Base Register           2008 0010
Reserved                            2008 0014 - 2008 3FFF
Interrupt Status Register           2008 4000 (mips unique)
Boot and Diagnostic Register        2008 4004
Select Processor Register           2008 4008 (mips unique)
Interval Timer Register             2008 4010
Reserved I/O Module Address Space   2008 4014 - 2008 7FFF
Q22-bus Map Registers               2008 8000 - 2008 FFFF
Reserved I/O Module Address Space   2009 0000 - 2009 FFFF
DSSI Buffer RAM                     2010 0000 - 2011 FFFF
NI Station Address ROM              2012 0000 - 2012 007C
Reserved I/O Module Address Space   2012 0080 - 2013 FFFF
SSC Base Address Register           2014 0000
SSC Configuration Register          2014 0010
CDAL Bus Timeout Control Register   2014 0020
Diagnostic LED Register             2014 0030
Reserved I/O Module Address Space   2014 0034 - 2014 0068
Time of Year Register               2014 006C
     Reserved                             2014 0070 - 2014 007C
CVAX Console Rcvr Control/Status    2014 0080
CVAX Console Rcvr Data Buffer       2014 0084
CVAX Console Xmit Control/Status    2014 0088
CVAX Console Xmit Data Buffer       2014 008C
Reserved                            2014 0090 - 2014 00DB
I/O System Reset Register           2014 00DC
Reserved                            2014 00E0
Rom Data Register                   2014 00F0
Bus Timeout Counter                 2014 00F4
Interval Timer                      2014 00F8
Reserved                            2014 00FC - 2014 00FF
Timer 0 Control Register            2014 0200
Timer 0 Interval Register           2014 0104
Timer 0 Next Interval Register      2014 0108
Timer 0 Interrupt Vector            2014 010C
Timer 1 Control Register            2014 0110
Timer 1 Interval Register           2014 0114
Timer 1 Next Interval Register      2014 0118
```

	Timer 1 Interrupt Vector	2014 011C	
	DSSIDB Address Decode Match Reg	2014 0130	
	DSSIDB Address Decode Mask Register	2014 0134	
	LIOD Address Decode Match Register	2014 0140	
	LIOD Address Decode Mask Register	2014 0144	
	Reserved	2014 0148 - 2014 033F	
	CVAX Battery Backed-Up RAM	2014 0400 - 2014 07FF	
	Reserved I/O Module Address Space	2014 0800 - 2015 FFFF	
	DSSI Diagnostic Register 0	2016 0000	
	DSSI Diagnostic Register 1	2016 0004	
	DSSI Diagnostic Register 2	2016 0008	
	DSSI Control and Status Register	2016 000C	
	DSSI ID Register	2016 0010	
	Reserved DSSI Register	2016 0014	
	Reserved DSSI Register	2016 0018	
	DSSI Timeout Register	2016 001C	
	Reserved DSSI Register	2016 0020	
	Reserved DSSI Register	2016 0024	
	Reserved DSSI Register	2016 0028	
	Reserved DSSI Register	2016 002C	
	Reserved DSSI Register	2016 0030	
	Reserved DSSI Register	2016 0034	
	DSSI Short Target List Pointer	2016 0038	
	DSSI Long Target List Pointer	2016 003C	
	DSSI Initiator List Pointer	2016 0040	
	DSSI DSSI Control REgister	2016 0044	
	DSSI DSSI Status REgister	2016 0048	
	Reserved DSSI Register	2016 004C	
	Reserved DSSI Register	2016 0050	
	DSSI Diagnostic Control Register	2016 0054	
	DSSI Clock Control Register	2016 0058	
	DSSI Internal State Register 0	2016 005C	
	DSSI Internal State Register 1	2016 0060	
	DSSI Internal State Register 2	2016 0064	
	DSSI Internal State Register 2	2016 0068	
	Reserved DSSI Register	2016 006C	
	Reserved DSSI Register	2016 0070	
	Reserved DSSI Register	2016 0074	
	Reserved DSSI Register	2016 0078	
	Reserved DSSI Register	2016 007C	
	Reserved I/O Module Address Space	2016 0080 - 2017 FFFF	
	Reserved	2018 0000 - 23FF FFFF	
	Reserved	2400 0000 - 24FF FFFF	
	Reserved	2500 0000 - 26FF FFFF	
*	I/O Presence Register	2700 0000 - 2703 FFFF	(mips unique)
*	Memory Error Syndrome Register	2704 0000 - 2707 FFFF	(mips unique)
*	Memory Error Address Register	2708 0000 - 270B FFFF	(mips unique)
*	Memory ID Register	270C 0000 - 270F FFFF	(mips unique)
	53C94 Transfer Counter Lo Register	2710 0000	(Read Only)
	53C94 Transfer Count Lo Register	2710 0000	(Write Only)
	53C94 Transfer Counter Hi Register	2710 0004	(Read Only)
	53C94 Transfer Count Hi Register	2710 0004	(Write Only)
	53C94 FIFO Register	2710 0008	
	53C94 Command Register	2710 000C	
	53C94 Status Register	2710 0010	(Read Only)
	53C94 Select Bus ID Register	2710 0010	(Write Only)
	53C94 Interrupt Status Register	2710 0014	(Read Only)
	53C94 Select Timeout Register	2710 0014	(Write Only)
	53C94 Sequence Step Register	2710 0018	(Read Only)
	53C94 Sync Transfer Period Register	2710 0018	(Write Only)
	53C94 FIFO Flags Register	2710 001C	(Read Only)
	53C94 Synchronous Offset Register	2710 001C	(Write Only)
	53C94 Configuration Register	2710 0020	
	53C94 Clock Conversion Register	2710 0024	(Write Only)
	53C94 Test Register	2710 0028	(Write Only)

	Reserved	2710 002C - 2710 003C
	Reserved (copies of SCSI Regs)	2710 0040 - 2713 FFFF
	SCSI DMA Register	2714 0000 (Write Only)
	Reserved (copies of SCSI DMA Reg)	2714 0004 - 2717 FFFF
	SCSI Buffer RAM	2718 0000 - 2719 FFFF
	Reserved (copy of SCSI RAM)	271A 0000 - 271B FFFF
	Reserved SCSI address space	271C 0000 - 271F FFFF
*	R3000 LED Register	2720 0000 - 2723 FFFF
	Reserved I/O Module Address Space	2724 0000 - 27FF FFFF
*	R3000 Non-Volatile RAM	2800 0000 - 2807 FFFF
	Reserved I/O Module Address Space	2808 0000 - 280F FFFF
*	R3000 UART Registers	2810 0000 - 2810 003C
*	Reserved (copies of UART Regs)	2810 0040 - 2813 FFFF
	Reserved I/O Module Address Space	2814 0000 - 28FF FFFF
	Reserved I/O Module Address Space	2910 0000 - 291F FFFF
	Reserved I/O Module Address Space	2940 0000 - 2FBF FFFF
*	R3000 ROM	2FC0 0000 - 2FC3 FFFF
	Reserved I/O Module Address Space	2FC4 0000 - 2FFF FFFF
	Local Q22-bus Memory Space	2400 0000 - 243F FFFF

I

Prestoserve

I.1 Overview

The DECsystem 5500 contains .5 Mbyte of non-volatile battery-backup-up RAM memory for use by Prestoserve disk cache software. This NVRAM and the Prestoserve software can be used together to increase the performance of the ULTRIX Network File System (NFS). The NVRAM acts as a write cache for synchronous disk I/O. In Section I.3 is a brief description of some of the firmware commands associated with Prestoserve software.

For more detailed information on Prestoserve and the commands used under the ULTRIX operating system, refer to the *Guide to Prestoserve* (AA-PE4YA-TE).

I.2 Operation

If Prestoserve is being run on the DECsystem 5500 and normal shutdown procedures are followed, the data is automatically recovered and moved to the appropriate disks.

If Prestoserve is being used on the DECsystem 5500 and the system is not shut down normally, because of a power failure, hardware failure or software failure, the data in the NVRAM will be backed up by a battery on the KN220-AA CPU module. When ULTRIX is brought back up, the data will be moved to the appropriate locations on the disk(s) by Prestoserve software. If the system cannot be re-booted, due to a hardware problem, the data may be able to be saved on tape and restored to the replacement CPU module.

NOTE

If a DECsystem 5500 is running Prestoserve and ULTRIX is not shut down in a orderly fashion, the following MUST be done in order to preserve critical data in the NVRAM and to prevent possible DATA CORRUPTION (only if the KN220-AA CPU module is to be replaced and ULTRIX booted during repair or troubleshooting). No special action needs to be taken if ULTRIX is not booted and ONLY diagnostics are going to be run on the spare module. If ULTRIX is going to be booted on the spare CPU module, it is CRITICAL that the data be moved from the NVRAM on the bad CPU module to tape (if the hardware problem is not too severe to perform this task) and restored to the replacement module before ULTRIX is booted. Section I.3 list the firmware commands that will allow you to perform this task.

I.3 Firmware Commands

The following sections give a brief description of some of the firmware commands associated with Prestoserve.

I.3.1 The dc Command

To determine if the Prestoserve cache contains data, use the dc command.

The following is an example of the dc command when the Prestoserve cache contains data and is considered dirty:

```
>>> dc RET
Disk Cache - Dirty
>>>
```

If the cache does not contain any data, it is considered clean, and the following message is displayed on the console terminal:

```
Disk Cache - Clean
```

If the re is not data in the cache, you can follow normal procedures for rebooting or troubleshooting.

NOTE
Always run the dc command before removing or using any DECsystem 5500 CPU module to avoid losing data or corrupting your disk(s).

I.3.2 The dc/save Command

The following is an example of the dc/save command when the cache contains data:

```
>>> dc/save mua0 RET
Disk Cache - Dirty
Do you want to continue (y/n)? y RET
-MUA0
Zero Disk Cache (y/n)? y RET
>>>
```

The following is an example of the dc/save command when the cache contains no data, and there is nothing to write to the disk:

```
>>> dc/save mua0 RET
Disk Cache - Clean
>>>
```

I.3.3 The dc/zero Command

If you want to clear the contents of the Prestoserve cache because the data is not wanted or because the data cannot be saved to tape with the dc/save command, use the dc/zero command.

The dc/zero command prompts you to confirm that you want to continue and, if you answer Y, fills the cache with zeroes. Use this command as a security measure to ensure that the cache is cleared.

An example of the command is below:

```
>>> dc/zero RET
Do you want to continue (y/n)? n RET
>>>
```

I.3.4 The dc/restore Command

The following is an example of the dc/restore command when the Prestoserve cache contains data to be restore to the replacement KN220-AA CPU module:

```
>>> dc/restore mua0 RET
Disk Cache - Dirty
Do you want to continue (y/n)? n RET
>>>
```

The following is an example of the dc/restore command when the cache contains no data:

```
>>> dc/restore mua0 RET

-MUA0
>>>
```

Index

D

E

F

M

N

O

P

Q

R

S

T

READER'S COMMENTS

1. **How did you use this manual? (Circle your response.)**

   (a) Installation (c) Maintenance (e) Training
   (b) Operation/use (d) Programming (f) Other (Please Specify.) _____

2. **Did the manual meet your needs? Yes ☐ No ☐ Why?** _____

   _____

3. **Please rate the manual on the following categories. (Circle your responses.)**

	Excellent	Good	Fair	Poor	Unacceptable
Accuracy	5	4	3	2	1
Clarity	5	4	3	2	1
Completeness	5	4	3	2	1
Table of Contents, Index	5	4	3	2	1
Illustrations, examples	5	4	3	2	1
Overall ease of use	5	4	3	2	1

4. **What did you like *most* about this manual?** _____

   _____

5. **What did you like *least* about this manual?** _____

   _____

6. **Please list and describe any errors you found in the manual.**

Page	Description/Location of Error
_____	_____
_____	_____
_____	_____
_____	_____

7. **Which of the following most clearly describes your job? (Circle your response.)**

   (a) Administrative Support (e) System Manager (i) Educational/Trainer
   (b) Manager/Supervisor (f) Computer Operator (j) Sales/Marketing
   (c) Scientist/Engineer (g) Software Support (k) Other (Please specify)
   (d) Programmer/Analyst (h) Hardware Support _____

OPTIONAL INFORMATION

Name _____ Street _____
Company _____ City/State _____
Department _____ Country _____ Postal (ZIP) code (_____)
Job Title _____ Telephone Number _____

THANK YOU FOR YOUR COMMENTS AND SUGGESTIONS

Please do not use this form to order manuals. Contact your representative at Digital Equipment Corporation or (in the USA) call our DECdirect™ department at this toll-free number: 800–344–4825.

FOLD HERE AND TAPE. DO NOT STAPLE

digital™

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD, MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
Educational Services/Quality Assurance
12 Crosby Drive BU0/E08
Bedford, MA 01730-1493

FOLD HERE AND TAPE. DO NOT STAPLE