EK-CIBCI-UG-0001

CIBCI Adapter User/Installation Guide

Prepared by Educational Services
of
Digital Equipment Corporation

1st Edition, October 1986

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

Printed in U.S.A.

This document was set on a DIGITAL DECset Integrated Publishing System.

- Class B Computing Devices:

Notice: This equipment generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such radio frequency interference when operated in a commercial environment. Operation of this equipment in a residential area may cause interference in which case the user at his own expense may be required to take measures to correct the interference.

CONTENTS

PREFACE

CHAPTER 1	INTRODUCTION	Page
1.1	SCOPE	1-1
1.2	GENERAL DESCRIPTION	1-1
1.2.1	Components	1-1
1.2.2	Features	1-2
1.2.3	Configurations	1-3
1.3	GENERAL SPECIFICATIONS	1-4
1.3.1	Environmental Specifications	1-4
1.3.2	Mechanical Specifications	1-5
1.3.3	Electrical Specifications	1-5
1.3.3.1	VAXBI Bus Specifications	1-6
1.3.3.2	CIPA Bus Specifications	1-6
1.3.3.3	CI Bus Specifications	1-7
1.4	REFERENCE DOCUMENTS (Table 1-2)	1-8
1.5	PHYSICAL HARDWARE DESCRIPTION	1-9
1.5.1	Host Processor Interface Hardware	1-9
1.5.2	Computer Interconnect Interface Hardware	1-12
1.5.2.1	CI Port Adapter Assembly	1-12
1.5.2.2	CIPA Cabinet	1-13

CHAPTER 2

2.1	INTRODUCTION	2-1
2.2	OPERATING ENVIRONMENT	2-1
2.2.1	Physical Elements	2-1
2.2.2	Environmental Elements	2-1
2.2.3	Grounding Elements	2-2
2.3	CIBCI SYSTEM CONFIGURATIONS	2-2
2.4	UNPACKING AND INVENTORYING	2-2
2.4.1	VAX 8200/8300 Systems	2-2
2.4.1.1	Verifying Shipment Inventory	2-2
2.4.1.2	Unpacking the Shipping Boxes	2-3
2.4.1.3	Unpacking the H9642 CIPA Cabinet Carton	2-5
2.4.2	VAX 8500/8550/8700/8800 Systems	2-7
2.4.2.1	Verifying Shipment Inventory	2-7
2.4.2.2	Unpacking the Shipping Boxes	2-8
2.4.2.3	Unpacking the H9652 Expander Cabinet	2-8

CONTENTS (Cont)

		Page
2.5	MECHANICAL INSTALLATION (System Level)	2-10
2.5.1	VAX 8200/8300 Systems	2-10
2.5.1.1	CPU Cabinet Preparation	2-10
2.5.1.2	H9642 CIPA Cabinet Preparation	2-14
2.5.2	VAX 8500/8550 Systems	2-23
2.5.3	VAX 8700/8800 Systems	2-28
2.6	MECHANICAL INSTALLATION (Add-On Level)	2-38
2.6.1	VAX 8200/8300 Systems	2-41
2.6.2	VAX 8500/8550/8700/8800 Systems	2-45
2.7	ELECTRICAL INSTALLATION AND CONFIGURATION	2-48
2.7.1	T1017 and T1018 Module Installation	2-48
2.7.2	CIPA Backplane Jumpers Verification	2-54
2.7.3	Node Address Switch Verification	2-58
CHAPTER 3		
3.1	INTRODUCTION	3-1
3.2	POWER SETUP AND VERIFICATION	3-1
3.2.1	VAX 8200/8300 Systems	3-1
3.2.2	VAX 8500/8550/8700/8800 Systems	3-4
3.3	DIAGNOSTIC VERIFICATION	3-5
3.3.1	VAX 8200/8300 Systems	3-6
3.3.1.1	Preliminary Setup	3-6
3.3.1.2	Loading the Diagnostic Supervisor Program	3-6
3.3.1.3	Repair Level Testing	3-12
3.3.1.4	CI Bus Cable Testing	3-13
3.3.1.5	Functional Level Testing	3-13
3.3.2	VAX 8500/8550/8700/8800 Systems	3-16
3.3.2.1	Preliminary Setup	3-16
3.3.2.2	Loading the Diagnostic Supervisor Program	3-17
3.3.2.3	Repair Level Testing	3-17
3.3.2.4	CI Bus Cable Testing	3-23
3.3.2.5	Functional Level Testing	3-23
3.4	MAINTENANCE VERIFICATION	3-27
CHAPTER 4	REGISTER SUMMARY	
4.1	VAXBI ADDRESS SPACE	4-1
4.1.1	VAXBI I/O Address Space	4-1
4.2	CIBCI ADAPTER NODE	4-1
4.2.1	Addressing	4-1
4.2.2	Partitioning	4-4
4.2.3	Registers	4-4
4.3	VAXBI REQUIRED REGISTERS	4-4
4.3.1	Device Type Register (DTR)	4-6
4.3.2	VAXBI Control and Status Register (BICSR)	4-6
4.3.3	Bus Error Register (BER)	4-9
4.3.4	Error Interrupt Control Register (EICR)	4-11
4.3.5	Interrupt Destination Register (IDR)	4-12

CONTENTS (Cont)

		Page
4.3.6	Inter-Processor Interrupt Mask Register (IPIMR)	4-12
4.4	BIIC SPECIFIC DEVICE REGISTERS	4-13
4.4.1	BCI Control Register (BCICR)	4-14
4.4.2	User Interrupt Control Register (UICR)	4-15
4.5	BIIC USER PORT REGISTERS	4-17
4.5.1	BCIA Configuration Register (CNFGR)	4-18
4.5.2	BICA Address Register (BCAR)	4-20
4.5.3	BICA Command/Byte Mask Register (BCMR)	4-22
4.5.4	DMA Register File (DMAF)	4-23
4.6	CIPA REGISTERS	4-23
4.6.1	Port Maintenance Control/Status Register (PMCSR)	4-25
4.6.2	Maintenance Address Register (MADR)	4-27
4.6.3	Maintenance Data Register (MDATR)	4-29
4.6.4	Port Status Register (PSR)	4-29
4.6.5	Port Queue Block Base Register (PQBBR)	4-29
4.6.6	Port Command Queue 0 Control Register (PCQ0CR)	4-32
4.6.7	Port Command Queue 1 Control Register (PCQ1CR)	4-32
4.6.8	Port Command Queue 2 Control Register (PCQ2CR)	4-32
4.6.9	Port Command Queue 3 Control Register (PCQ3CR)	4-33
4.6.10	Port Status Release Control Register (PSRCR)	4-34
4.6.11	Port Enable Control Register (PECR)	4-34
4.6.12	Port Disable Control Register (PDCR)	4-35
4.6.13	Port Initialize Control Register (PICR)	4-35
4.6.14	Port Datagram Free Queue Control Register (PDFQCR)	4-36
4.6.15	Port Message Free Queue Control Register (PMFQCR)	4-36
4.6.16	Port Maintenance Timer Control Register (PMTCR)	4-37
4.6.17	Port Maintenance Timer Expiration Control Register (PMTECR)	4-37
4.6.18	Port Failing Address Register (PFAR)	4-38
4.6.19	Port Error Status Register (PESR)	4-39
4.6.20	Port Parameter Register (PPR)	4-40
APPENDIX A	CI TERMINATION	
APPENDIX B	CI BACKPLANE JUMPER	
B.1	BOOT TIMER PARAMETERS	B-1
B.2	EXTENDED HEADER/TRAILER (W5)	B-1
B.3	ALTER DELTA TIME (W6)	B-1
B.4	DISABLE ARBITRATION (W7)	B-1
B.5	EXTENDED ACKNOWLEDGEMENT TIMEOUT (W8)	B-2
INDEX		

FIGURES

Figure No.	Title	Page
1-1	Simplified CIBCI Adapter Connection	1-[illegible]
1-2	Simplified CIBCI Interface Block Diagram	1-10
1-3	Hardware Components of the Host Processor Interface	1-11
1-4	Hardware Components of the CIPA Mounting Box	1-12
2-1	Single CIBCI Adapter Configuration – VAX 8200/8300	2-3
2-2	Single CIBCI Adapter Configuration – VAX 8800	2-4
2-3	Dual CIBCI Adapter Configuration – VAX 8200/8300	2-5
2-4	Unpacking the H9642 Cabinet	2-6
2-5	Removing the H9642 Cabinet from the Shipping Pallet	2-7
2-6	Unpacking the H9652 Cabinet	2-9
2-7	Removing the H9652 Cabinet from the Shipping Pallet	2-10
2-8	Accessing the BA32 CPU Mounting Box Hardware Components	2-12
2-9	CPU Cabinet End Panel Removal	2-13
2-10	CIPA Cabinet – Front and Rear Panel Removal	2-14
2-11	CIPA Cabinet – Expansion Panel Removal	2-15
2-12	CIPA Cabinet – RFI Shield Panel Removal	2-16
2-13	RFI Shield Panel – Knockout Plug Removal	2-17
2-14	RFI Shield Panel – Waveguide Installation	2-18
2-15	H9642 Cabinet – Rear Interior View	2-19
2-16	CPIA Bus Cable Routing	2-21
2-17	CPU and CIPA Cabinet Mating	2-22
2-18	CPU and Expansion Cabinet Mating	2-23
2-19	CPU and Front-End Cabinet Mating	2-29
2-20	CIPA Bus Cable Routing	2-38
2-21	CIPA Mounting Box Differences	2-39
2-22	Mounting Box Installation Within An H9642 Cabinet	2-41
2-23	Mounting Box Installation Within An H9652 Cabinet	2-45
2-24	VAXBI Backplane – Module Installation	2-49
2-25	VAXBI Backplane – BCI Cable Connections	2-51
2-26	VAXBI Backplane – CIPA Bus Cable Connections	2-52
2-27	CIPA Mounting Box – Access	2-54
2-28	CIPA Mounting Box – Backplane Jumpers	2-57
2-29	CIPA Mounting Box – Module Access	2-58
3-1	Control Panel Switches and LEDs	3-2
3-2	Diagnostic Loopback Cable Connections	3-6
3-3	Diagnostic Loopback Cable Connections	3-16
4-1	VAXBI Physical Address Space	4-2
4-2	VAXBI Physical I/O Address Space	4-2
4-3	VAXBI Node Space	4-3
4-4	30-Bit I/O Address Bit Map	4-3
4-5	CIBCI Address Node Space	4-5
4-6	CIBCI Adapter Register Address Space	4-5
4-7	VAXBI Interface Registers and Adapter Registers	4-6
4-8	VAXBI Required Registers	4-6
4-9	Device Type Register Bit Map	4-7
4-10	VAXBI Control and Status Register Bit Map	4-7
4-11	Bus Error Register Bit Map	4-9
4-12	Error Interrupt Control Register Bit Map	4-11
4-13	Interrupt Destination Register Bit Map	4-13

FIGURES (Cont)

Figure No.	Title	Page
4-14	Inter-Processor Interrupt Mask Register Bit Map	4-13
4-15	VAXBI Specific Device Registers	4-13
4-16	BCI Control Register Bit Map	4-14
4-17	User Interrupt Control Register Bit Map	4-16
4-18	BI User Port Registers	4-17
4-19	BICA Configuration Register Bit Map	4-18
4-20	BICA Address Register Bit Map	4-20
4-21	BICA Command/Byte Mask Register Bit Map	4-22
4-22	DMA Register File Bit Map	4-24
4-23	CIPA Adapter Registers	4-24
4-24	Port Maintenance Control/Status Register Bit Map	4-25
4-25	Maintenance Address Register Bit Map	4-28
4-26	Maintenance Data Register Bit Map	4-29
4-27	Port Status Register Bit Map	4-30
4-28	Port Queue Block Base Register Bit Map	4-31
4-29	Port Command Queue 0 Control Register Bit Map	4-32
4-30	Port Command Queue 1 Control Register Bit Map	4-32
4-31	Port Command Queue 2 Control Register Bit Map	4-33
4-32	Port Command Queue 3 Control Register Bit Map	4-33
4-33	Port Status Release Control Register Bit Map	4-34
4-34	Port Enable Control Register Bit Map	4-34
4-35	Port Disable Control Register Bit Map	4-35
4-36	Port Initialize Control Register Bit Map	4-35
4-37	Port Datagram Free Queue Control Register Bit Map	4-36
4-38	Port Message Free Queue Control Register Bit Map	4-36
4-39	Port Message Control Register Bit Map	4-37
4-40	Port Maintenance Timer Expiration Control Register Bit Map	4-37
4-41	Port Failing Address Register Bit Map	4-38
4-42	Port Error Status Register Bit Map	4-39
4-43	Port Parameter Register Bit Map	4-40

EXAMPLES

Example No.	Title	Page
3-1	Trace Printout for Repair Diagnostic EVCKA	3-8
3-2	Trace Printout for Repair Diagnostic EVCKB	3-9
3-3	Trace Printout for Repair Diagnostic EVCKC	3-10
3-4	Trace Printout for Repair Diagnostic EVCKD	3-11
3-5	Trace Printout for Repair Diagnostic EVCKE	3-12
3-6	Trace Printout for Repair Diagnostic EVCKF	3-12
3-7	Trace Printout for Functional Diagnostic EVGAA	3-14
3-8	Trace Printout for Functional Diagnostic EVGAB	3-15
3-9	Trace Printout for Repair Diagnostic EVCKA	3-18
3-10	Trace Printout for Repair Diagnostic EVCKB	3-19
3-11	Trace Printout for Repair Diagnostic EVCKC	3-20

EXAMPLES (Cont)

		Page
3-12	Trace Printout for Repair Diagnostic EVCKD	3-21
3-13	Trace Printout for Repair Diagnostic EVCKE	3-22
3-14	Trace Printout for Repair Diagnostic EVCKF	3-22
3-15	Trace Printout for Functional Diagnostic EVGAA	3-25
3-16	Trace Printout for Functional Diagnostic EVGAB	3-26

TABLES

Table No.	Title	Page
1-1	CIBCI Hardware Variations	1-3
1-2	Related CIBCI Adapter Documentation	1-8
1-3	Hardware Components of the CIBCI Adapter	1-9
2-1	Node Address Switch Settings	2-62
3-1	List of the CIBCI Diagnostic Programs	3-5
3-2	Summary of the Diagnostic Testing Hierarchy	3-5
3-3	Summary of the Functions of VAXcluster System Maintenance and Management Tools	3-27
4-1	Node Space Address Assignments	4-4
4-2	VAXBI Control and Status Register Bits	4-8
4-3	Bus Error Register Bits	4-10
4-4	Error Interrupt Control Register Bits	4-12
4-5	BCI Control Register Bits	4-14
4-6	User Interrupt Control Register Bits	4-16
4-7	BCIA Configuration Register Bits	4-18
4-8	BCIA Address Register Bits	4-21
4-9	BCIA Command/Byte Mask Register Bits	4-22
4-10	Port Control and Status Register Bits	4-26
4-11	Maintenance Address Register Bits	4-28
4-12	Port Status Register Bits	4-30
4-13	Port Failing Address Register Bits	4-38
4-14	Port Error Status Register Bits	4-39
4-15	Port Parameter Register Bits	4-40

PREFACE

INTENDED READER

This document provides an introduction to Digital Equipment Corporation's computer interconnect hardware adapter option (CIBCI). It presents information required by the user for the configuration, installation, and acceptance verification of the CIBCI hardware on a VAX 8000-Series system incorporating the VAX backplane interconnect (VAXBI)* bus architecture.

NOTE

Throughout this manual the term "VAX 8000-Series" shall be used to represent the following processors that support the CIBCI hardware: VAX 8200, VAX 8300, VAX 8500, VAX 8550, VAX 8700, and VAX 8800.

GUIDE STRUCTURE

Chapter 1 – Describes the CIBCI hardware and lists its specifications.

Chapter 2 – Describes the procedures for installing the hardware on a VAX 8000-Series system.

Chapter 3 – Describes the procedures for verifying the functionality of the hardware.

Chapter 4 – Provides a reference section describing the programmer visible registers.

Appendix A – Provides information on CI termination.

Appendix B – Provides information on the CI backplane jumpers.

---

* VAXBI is a trademark of Digital Equipment Corporation.

CHAPTER 1
INTRODUCTION

1.1 SCOPE
This chapter introduces the reader to the computer interconnect hardware adapter option (CIBCI) used on the VAX 8000-Series systems that incorporate a VAX backplane interconnect (VAXBI) bus architecture. The chapter also contains a physical description and the specifications of the hardware. Additional documents are listed for the user who wishes more information concerning VAXclusters.

1.2 GENERAL DESCRIPTION

1.2.1 Components
The computer interconnect adapter is shown in Figure 1-1 and is designated the CIBCI adapter. It is centrally controlled by a single, on-board data processor to provide buffered parallel-to-serial communications between two corporate interconnect bus architecture protocols: the VAX backplane interconnect (VAXBI) bus of the VAX 8000-Series host processor and the dual-path computer interconnect (CI) bus.

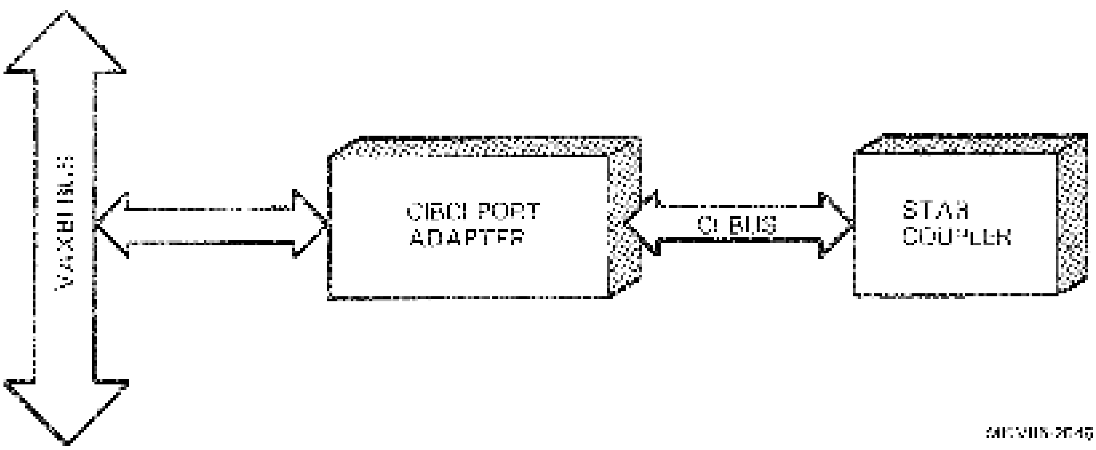


Figure 1-1 Simplified CIBCI Adapter Connection

As a buffered communications port, the CIBCI adapter completes high-level computer communications, thereby reducing software processing overhead. This is accomplished with hardware that provides all of the necessary data buffering, address translation, and serial data encoding/decoding. The CIBCI uses queue structures provided by the VAX/VMS operating system to transfer packet messages and to initiate the transfer of blocks of data between the VAX 8000-Series host memory system and/or other nodes within the VAXcluster.

The CIBCI adapter is partitioned into two separate hardware interfaces: one host processor interface and one computer interconnect port adapter interface. These interfaces consist of the following major components:

- Two Flatcard T-series modules
  - Adapter control module (BAC) T1017
  - Adapter data module (BAD) T1018
- Three extended hex L-series modules
  - Link interface module (ILI) L0100
  - Packet buffer module (IPB) L0101
  - Data path module (CDP) L0400

1.2.2 Features

- VAX backplane interconnect design
- 2M bytes/s performance at 40K packet transmission rate
- 3M bytes/s performance at 90K packet transmission rate
- Diagnostic data loopback (internal/external) capability
- Data integrity via cyclic redundancy checking
- Round-robin arbitration at heavy loading
- Contention arbitration at light loading
- Packet-oriented data transmission
- Operational modes:
  - Disabled
  - Enabled
  - Uninitialized
  - Disabled/maintenance
  - Enabled/maintenance
  - Uninitialized/maintenance

1.2.3 Configurations

The CIBCI adapter is available in either a single or a dual configuration. The single configuration is available without a star coupler, whereas the dual configuration is available with or without a star coupler.

Table 1-1 lists the model variations available with the CIBCI. The model variations are specified according to the host system type, electrical requirements, configuration type, and the major hardware components.

NOTE

The CI bus cables and SC008 star coupler are separately ordered options. They are NOT included as part of the bill of materials for all CIBCI adapter models.

Table 1-1 CIBCI Hardware Variations

CIBCI Model	VAX Host Processor	Input Power	Configuration Type	Hardware Components
CIBCI-AA CIBCI-AB	8200, 8300	120 Vac 240 Vac	Single	Host processor interface 17-01029 interface cables Rack-mountable CI box assembly
CIBCI-AC CIBCI-AD	8500, 8550, 8700, 8800	120 Vac 240 Vac	Single	Host processor interface 17-01029 interface cables Rack-mountable CI box assembly
CIBCI-BC CIBCI-BD	8200, 8300	120 Vac 240 Vac	Single	Host processor interface 17-01029 interface cables H9642 FCC cabinet Model 877-D/B power controller Rack-mountable CI box assembly
CIBCI-CA CIBCI-CB	8200, 8300	120 Vac 240 Vac	Dual	Two host processor interfaces 17-01029 interface cables H9642 FCC cabinet Two model 877-D/B power controllers SC008 star coupler Two rack-mountable CI box assemblies
CIBCI-DA CIBCI-DB	8200, 8300	120 Vac 240 Vac	Dual	Two host processor interfaces 17-01029 interface cables H9642 FCC cabinet Two model 877-D/B power controllers Two rack-mountable CI box assemblies

1.3 GENERAL SPECIFICATIONS

Priority arbitration

Light loading	Round-robin
Heavy loading	Contention

Parity	Cyclic redundancy check
Data format	Manchester-encoded serial packet
Data transfer rate	5M bytes/s maximum
Data throughput	2M to 3M bytes/s (typical sustained)
Operational modes	Disabled Disabled/maintenance Enabled Enabled/maintenance Uninitialized Uninitialized/maintenance

1.3.1 Environmental Specifications

Temperature

Operating	10°C to 40°C (50°F to 104°F) ambient temperature with a gradient of 10°C (18°F)/hr
Storage/shipping	−40°C to 70°C (−40°F to 158°F) ambient temperature with a gradient of 2°C (36°F)/hr

Relative humidity

Operating	10% to 90% with a maximum wet bulb temperature of 28°C (82°F) and a minimum dew point of 2°C (36°F) with no condensation
Storage/shipping	5% to 95% with no condensation

Altitude

Operating	Sea level to 2.4 km (8,000 ft) Maximum operating temperatures decrease by a factor of 1.8°C/1000 ft (1°F/1000 ft) for operation above sea level
Storage/shipping	Up to 9.1 km (30,000 ft) above sea level (actual or effective by means of cabin pressurization)

Shock	5 g peak at 7 to 13 ms duration in three axes mutually perpendicular (maximum)

Heat dissipation

Cooling	External forced air cooling at 2 m/s (400 linear ft/min)
Noise level	53 dB at 1 m

1.3.2 Mechanical Specifications

CI Box Assembly

Height	44.5 cm (17.5 in)
Width	66.7 cm (26.25 in)
Depth	86.4 cm (34.0 in)
Weight	37.5 kg (82.5 lb)

CI Cabinet

Height	106.1 cm (40.6 in)
Width	53.9 cm (21.2 in)
Depth	76.2 cm (30 in)
Weight	98.8 kg (200 lb)

1.3.3 Electrical Specifications

Power consumption

Host processor interface:

T1017 module	+5.0 Vdc at 3.74 A, 19.6 W
T1018 module	+5.0 Vdc at 3.75 A, 19.7 W

CI interface:

Mounting box	90–128 Vac, 47-63 Hz at 7.5 A 180–240 Vac, 47-63 Hz at 4.2 A

Plug type

120 Vac	NEMA 5-15P
220-240 Vac	NEMA 6-15P
H7202-B power supply rating	+5.0 Vdc at 60.0 A 5.3 Vdc at 10.0 A +12.3 Vdc at 2.0 A

Interface modules

L0100 module	−5.0 Vdc at 10.0 A, 50.0 W −5.2 Vdc at 5.3 A, 27.6 W
L0101 module	−5.0 Vdc at 10.5 A, 52.5 W
L0400 module	5.0 Vdc at 12.5 A, 62.5 W

1.3.3.1 VAXBI Bus Specifications –

Bus characteristics

Type	Synchronous
Width	32 data bits
Cycle time	200 ns
Priority arbitration	Distributed embedded arbitration
Parity	Odd
Data transfers	Block mode (masked) Longword Quadword Octaword

Transmission characteristics

Bandwidth	
Master port	11.4M bytes/s
Slave port	13.3M bytes/s
Length (maximum)	1.5 m (5 ft)
Bus loading (maximum)	16 logical nodes

1.3.3.2 CIPA Bus Specifications –

Bus characteristics

Type	Synchronous
Width	16 data bits
Cycle time	400 ns
Data parity	Odd
Data transfers	Packet

Transmission characteristics

Bandwidth	2M to 3M bytes/s
Length (Maximum)	4.5 m (15 ft)
Bus loading (maximum)	2
Impedance	120 Ohms

1.3.3.3 CI Bus Specifications

Bus characteristics

Type	Synchronous
Width	Serial
External length (maximum)	45 m (147.64 ft) radius (from the star coupler)
Data transfer rate	70M bits/s (maximum)
Bus loading (maximum)	16 logical nodes
Cable type (BNC1A-XX)	Shielded coaxial
Cable impedance	50 Ohms

1.4 REFERENCE DOCUMENTS (Table 1-2)

Table 1-2 Related [illegible] Documentation

Part Number	Title	Description
MP-0[illegible]-01	*CIBCI Print Set*	Contains all of the electrical and mechanical engineering drawings.
EK-CIBCI-TD	*CIBCI Hardware Technical Description*	Contains a technical description of the hardware and the registers.
EK-SC008-UG	*SC008 Star Coupler User's Guide*	Contains a description of the SC008 star coupler including instructions for unpacking and installing the various star coupler configurations.
SP-H7202D	*H7202D Power Supply Specification*	Contains complete mechanical and electrical specifications for the H7202D power supply including a general description of the H7202D power supply.
EK-PS730-TD	*H7202B Power System Technical Description*	Contains a physical and functional description of the H7202B power supply.
[illegible]	*VAX 8200/8300 Installation Guide*	Contains the procedures for unpacking, installing, configuring, and verifying the VAX 8200/8300 system.
EK-8500I-IN	*VAX 8500/8550 Installation Manual*	Contains the procedures for unpacking, installing, configuring, and verifying the VAX 8500/8550 system.
EK-8800I-IN	*VAX 8700/8800 Installation Manual*	Contains the procedures for unpacking, installing, configuring, and verifying the VAX 8700/8800 system.

1.5 PHYSICAL HARDWARE DESCRIPTION

Refer to Table 1-3 and Figure 1-2 for an overview of the hardware components for the host processor and computer interconnect port adapter interfaces.

Table 1-3 Hardware Components of the CIBCA Adapter

Interface	Component Type	Part Number
Host Processor	Adapter control module	T1017
	Adapter data module	T1018
	Two 5.5 inch BCI cables	17-01029-02
	Two 9.35 inch BCI cables	17-01029-01
Computer Interconnect Port Adapter	CIPA backplane [illegible]	[illegible]
	Link interface module	L0100
	Packet buffer module	L0101
	Data path module	L0102
	Power supply	H7202-B
	Backplane connector assembly	70-2[illegible]
	CIPA bus cable	17-01027-01

1.5.1 Host Processor Interface Hardware

Refer to Figure 1-3. The host processor interface consists of two T-series type modules. The T-series modules are housed in two adjacent slots within an H9400-A VAXBI cardcage of the VAX 8000-Series host system. These modules are used to interface the host system's VAXBI bus to the CIPA bus.

The Adapter Control Module

The adapter control module (BAC), part number T1017, contains the VAXBI protocol and the VAXBI control logic in addition to the CIPA control logic.

The Adapter Data Module

The adapter data module (BAD), part number T1018, is the major interface to the CI port adapter (CIPA) bus and consists of transaction buffers between the VAXBI and CIPA buses.

BCI Cables

The four BCI cables, part numbers 17-01029-01 and 17-01029-02, are used to electrically interconnect the T1017 and T1018 modules. Each cable consists of two 30-pin female connectors and two pull tabs or loops. The cables are mated to cable connectors located on the VAXBI cardcage corresponding to zones C and D of the modules. Two short cables (17-01029-02) complete the innermost electrical connection while two longer cables (17-01029-01) complete the outermost electrical connection between zones C and D of the two modules.

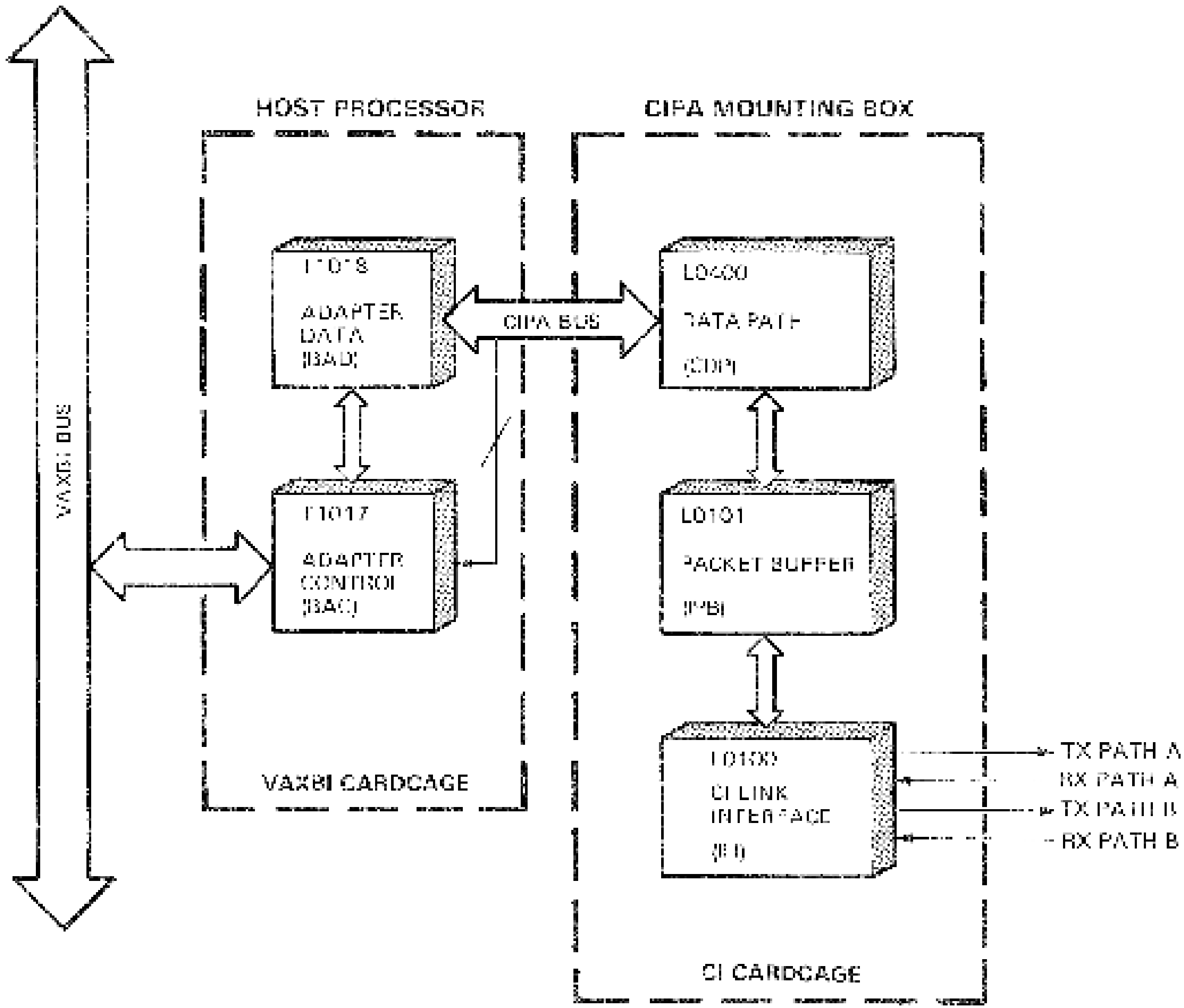


Figure 1-2 Simplified CIBCI Interface Block Diagram

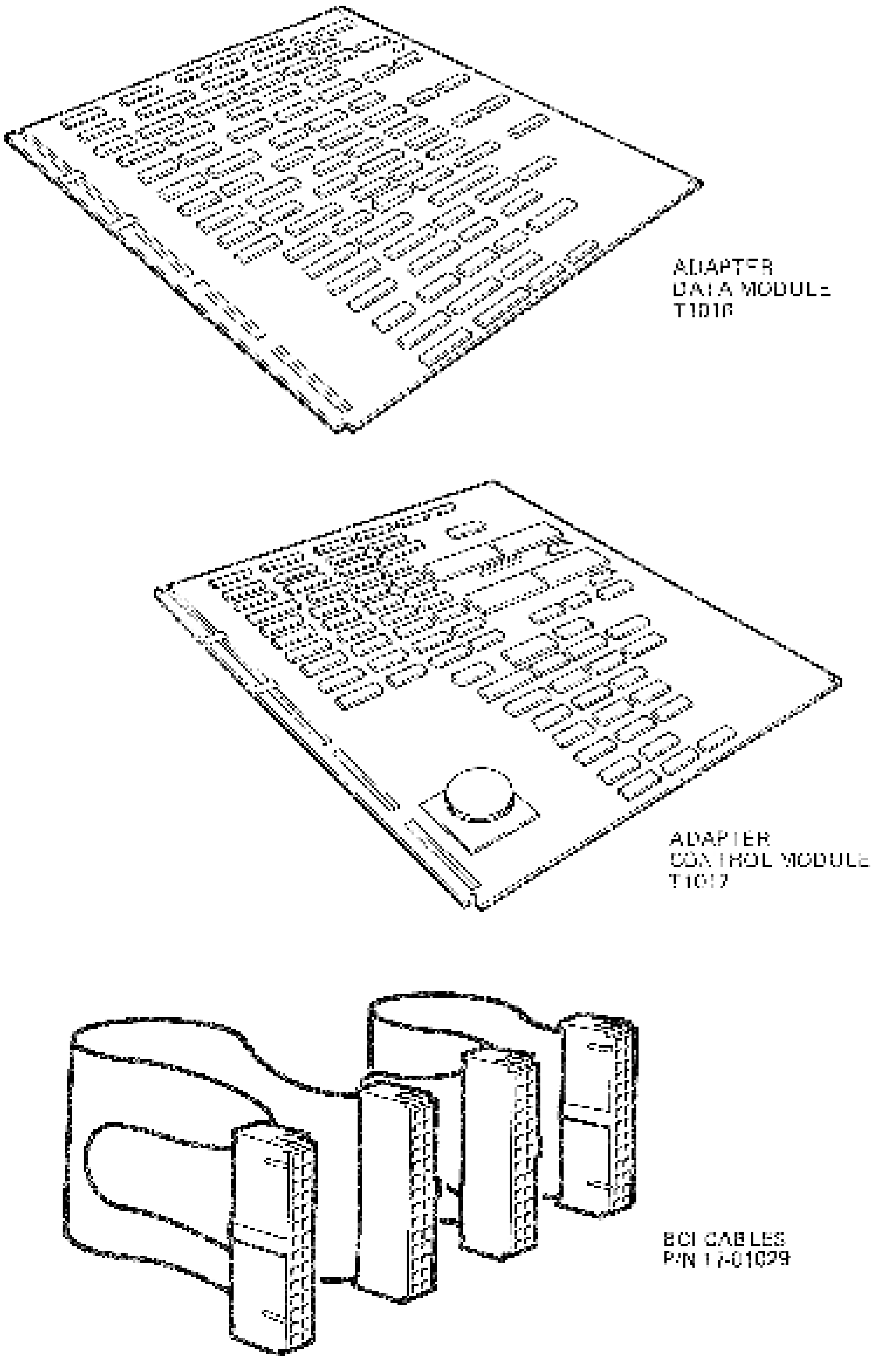


Figure 1-3 Hardware Components of the Host Processer Interface

1.5.2 Computer Interconnect Interface Hardware

1.5.2.1 CI Port Adapter Assembly Refer to Figure 1-4. The CI (computer interconnect) port adapter interface hardware is housed in a dedicated but universal mounting box is designated the CI box assembly. This CI box assembly is also referred to as the CIPA mounting box. It may reside either within the H9642 free-standing cabinet of a VAX 8200/8300 system or within the H9652 expander cabinet of a VAX 8500/8550/8700/8800 system

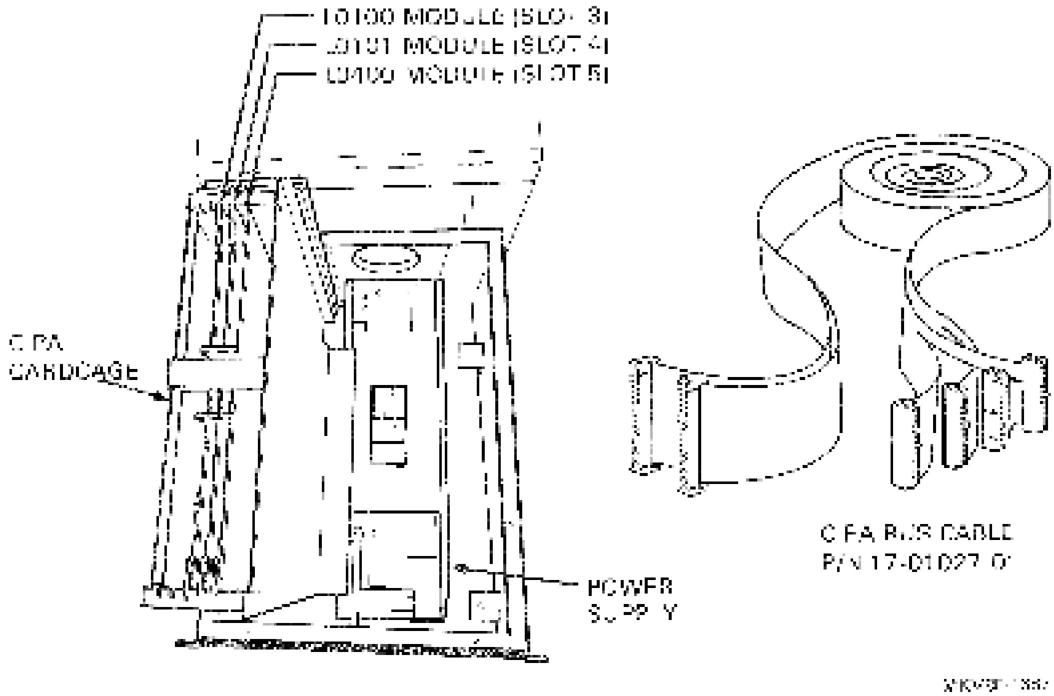


Figure 1-4 Hardware Components of the CIPA Mounting Box

The Power Supply
The CI box assembly uses a model H7202-D switching power supply to power the three L-series modules: the data path, packet buffer, and the CI link interface. The power supply receives its ac power from a power controller located within the H9642 free-standing cabinet or the H9652 expander cabinet.

The CIPA Cardcage
The CIPA cardcage, housed within the CI box assembly, is a five slot backplane used to house the three L-series modules: the data path, the packet buffer, and the CI link interface modules.

The Data Path Module
The data path module, part number L0400, provides the necessary arithmetic and logical processing of general port functions, as well as local storage for the port. It also provides transceivers and buffer registers as the interface for the CIPA bus.

The Packet Buffer Module
The packet buffer module (IPB), part number L0101, contains the port control store microcode and two transmit and receive CI packet buffers. Each CI packet buffer has a storage capacity of 1K bytes.

The CI Link Interface Module
The CI link interface module (ILI), part number L0100, is the actual interface to the CI bus and is capable of servicing dual CI paths. The module provides the necessary serialization and deserialization of data, data validity, CI bus protocol handling, and distributed priority arbitration. In addition, the module only permits transmission and reception of data packets over one CI data path at any given time. However, when four or more CI ports exist in the cluster, both CI data paths may be in use simultaneously. For example, node 0 to node 1 uses CI data path A while node 2 to node 3 uses CI data path B.

All data packets are appended with header and trailer information. The header information identifies the source and destination of the packet. Node address switches provide the node with an address on the CI cluster. The packet header contains this address as a source ID. The trailer information serves to keep the node receiver locked up while the last data bytes in the packet are being processed.

The CI Port Adapter Bus
The CI port adapter (CIPA) bus cable assembly, part number 17-01027-01, is a control and data bus used for backplane-to-backplane communication between the VAXBI and CIPA cardcages. The cable assembly consists of two 64.5 m (15 ft) flat ribbon cables separated by a foam material and contains a total of six female electrical connectors, two 40-pin connectors for connection to the CIPA backplane and four 30-pin connectors for connection to the VAXBI backplane.

1.5.2.2 CIPA Cabinet – When the CI mounting box assembly is mounted in a free-standing cabinet, the cabinet is identified as the CIPA cabinet. The CIPA cabinet is mounted immediately adjacent to the VAX 8200/8300 host system's VAXBI cabinet where the T-series type modules are housed. Contained within the CIPA cabinet are the power controller, CIPA mounting box assembly, and two CI bulkhead connector panels.

The Power Controller
The CIPA cabinet uses a model 877 (/D for 60 Hz or /B for 50 Hz), single-phase, ac power controller to provide electrical isolation and ac power for the CIPA mounting box assembly.

The CI Bulkhead Connector Panel
The CIPA backplane assembly is connected internally from the backplane to two CI bulkhead connector panels via two pairs of coaxial cables. The CI bulkhead connector panels provide the electrical isolation for the system by creating an EMI/RFI shield without compromising signal integrity. The panels are mounted in the cable connector openings located on the rear inside I/O panels of the cabinet. Two pairs of double-shielded coaxial cables connect the CI paths of the node from the CI bulkhead connector panels to the star coupler. One cable of each pair is for transmitting data; the other is for receiving data. Each cable pair connects to one CI bulkhead connector panel assembly.

CHAPTER 2
SITE PREPARATION AND INSTALLATION

2.1 INTRODUCTION

Chapter 2 contains information on site preparation and installation, including:

Operating Environment – Verifying that the CIBCI adapter hardware meets all of the minimum physical, environmental, and grounding specifications

System Configuration – Illustrating the various CIBCI adapter hardware configurations supported on a VAX 8000-Series system

Unpacking and Inventorying – Unpacking and verifying that the shipment is complete and undamaged

Mechanical Installation (System Level) – Installing an H9642 or H9652 cabinet onto a VAX 8000-Series system

Mechanical Installation (Add-on Level) – Installing a 10.5 inch CIPA mounting box into an existing cabinet

Electrical Installation – Installing the CIPA bus cable and configuring the node address of the CIBCI adapter hardware

2.2 OPERATING ENVIRONMENT

2.2.1 Physical Elements

The CIBCI adapter hardware requires adequate floor space and/or mounting space for the following:

- H9642 CIPA cabinet (VAX 8200/8300 system)
- H9652 expander cabinet (VAX 8500/8550/8700/8800 system)
- CI mounting box within either the H9642 or H9652 cabinet

2.2.2 Environmental Elements

A VAX 8000-Series system and its associated computer interconnect port adapter hardware are designed to operate in a "Class B" environment.

2.2.3 Grounding Elements

Careful grounding is essential in order to avoid ground loops and poor noise rejection. To eliminate ground loops and to have proper noise rejection, ensure the following:

- VAX 8000-Series system, expander cabinets, and all equipment share a common ac power source.
- Earth ground for the VAX 8000-Series system and expander cabinets are common.
- No electrically noisy equipment shares the same ac power source.
- Systems connected by the CI bus should not be connected for grounding unless another reason requires it.

CAUTION

The chassis for a VAX 8500/8550/8700/8800 system is isolated and floating, whereas the VAX 8200/8300 system chassis is grounded. Both affect the grounding constraints listed above.

2.3 CIBCI SYSTEM CONFIGURATIONS

Refer to Figures 2-1 through 2-3 for the VAX 8000-Series system configurations.

NOTE

Ensure that the CIBCI hardware and microcode revision level is consistent with the revision level of the cluster, and vice versa.

2.4 UNPACKING AND INVENTORYING

The CIBCI hardware is shipped in corrugated cartons and mounted on a pallet. Customers are responsible for the actual moving of the equipment to the installation site.

2.4.1 VAX 8200/8300 Systems

2.4.1.1 Verifying Shipment Inventory –

Procedure:

1. Inventory all equipment against the shipping list accompanying the equipment.
2. Notify the customer of any opened cartons or boxes and document this fact on the installation report.
3. Notify the field service unit manager of any missing or incorrect items.
4. Request that the customer contact the shipping carrier to locate any missing items.
5. Request that the field service unit manager check with the Digital Equipment Corporation Traffic and Shipping Department if the shipping carrier does not have the missing items.
6. Check all boxes for external damage (dents, holes, or crushed corners).
7. Notify the customer of all damages and list all damages on the installation report.

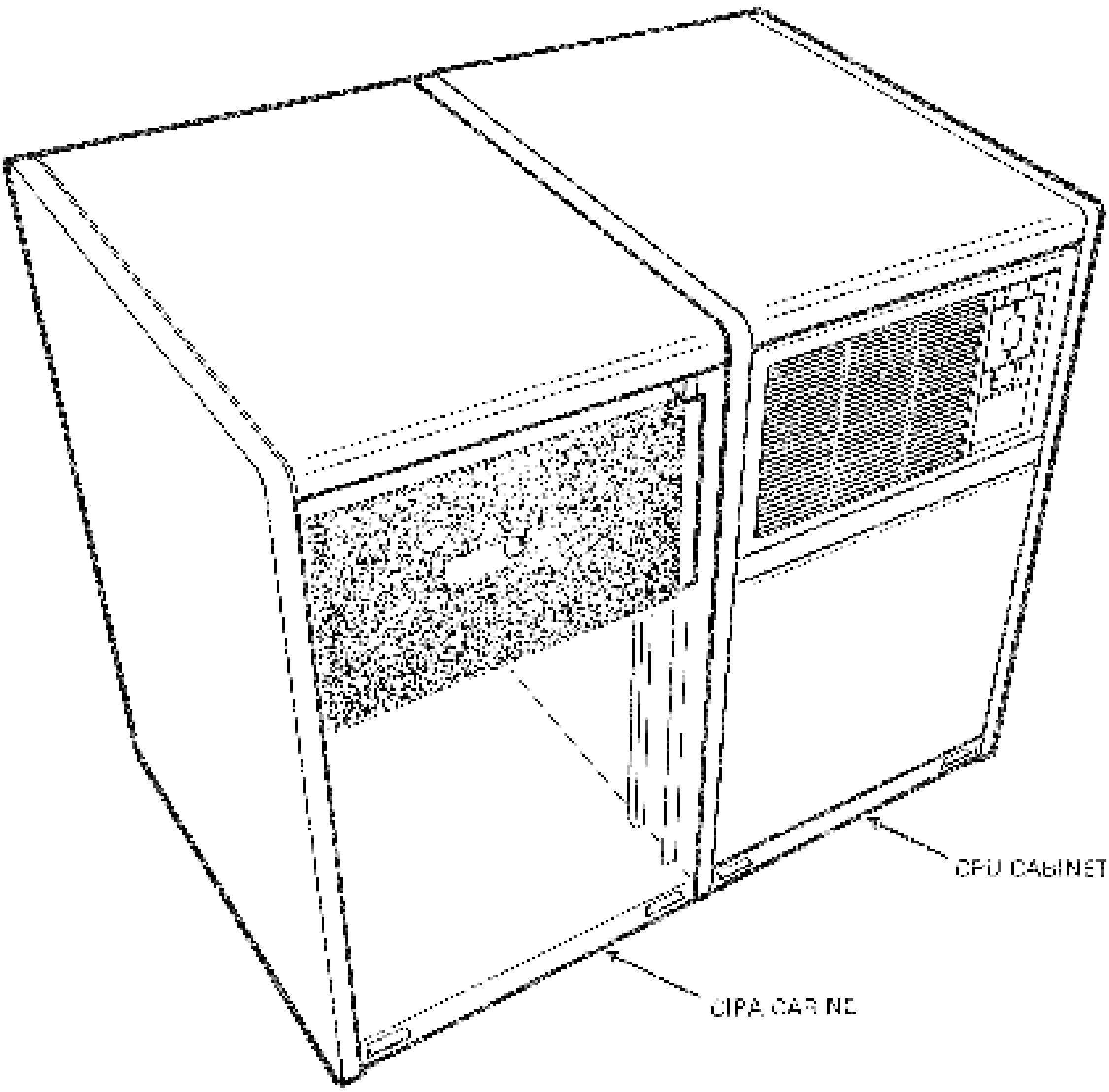


Figure 2-1 Single CIBCI Adapter Configuration – VAX 8200/8300

2.4.1.2 Unpacking the Shipping Boxes

Procedure:

1. Locate the box marked "OPEN ME FIRST".

2. Open all boxes and inventory the contents against the shipping/accessory list in the "OPEN ME FIRST" box.

3. Inspect the equipment for damage. Report any damage and note it on the the installation report.

4. If damage is extensive, notify Digital Equipment Corporation for instructions on how to proceed.

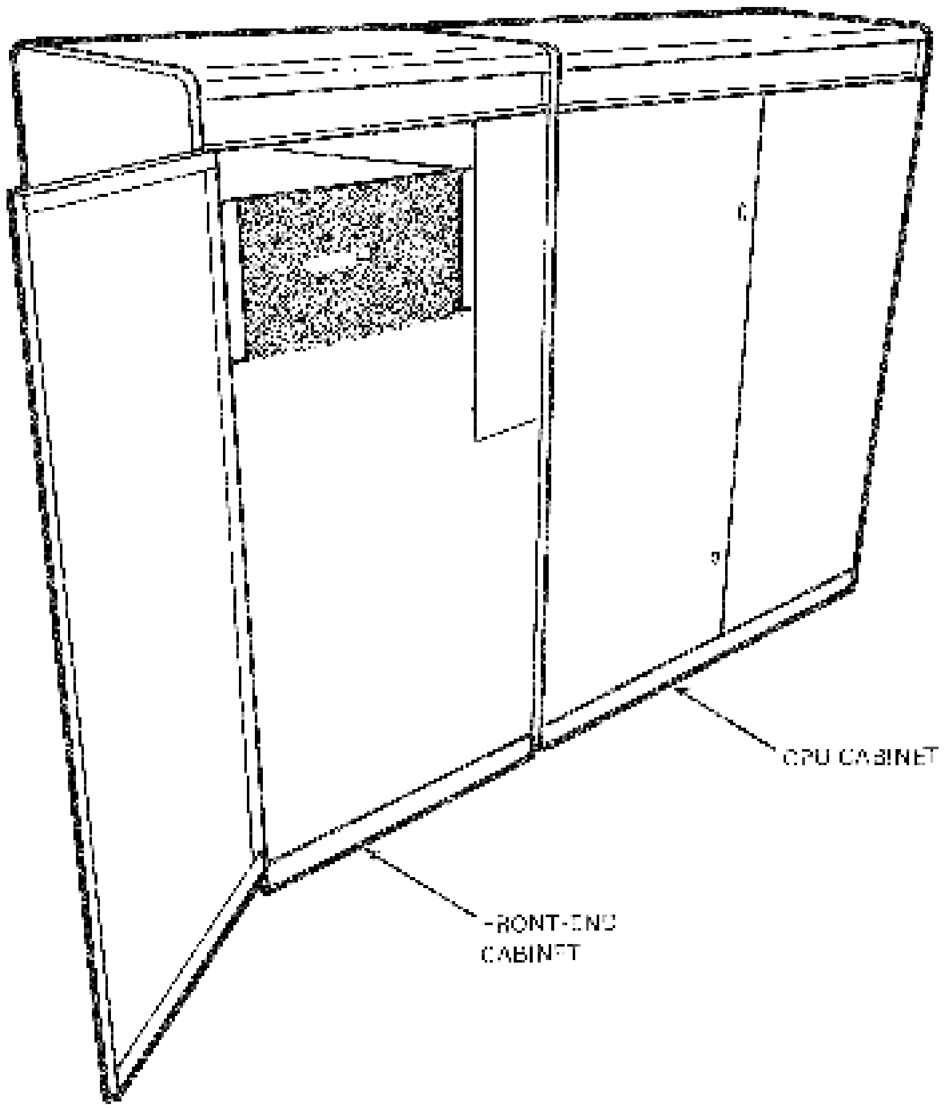


Figure 2-2 Single CIBCI Adapter Configuration – VAX 8800

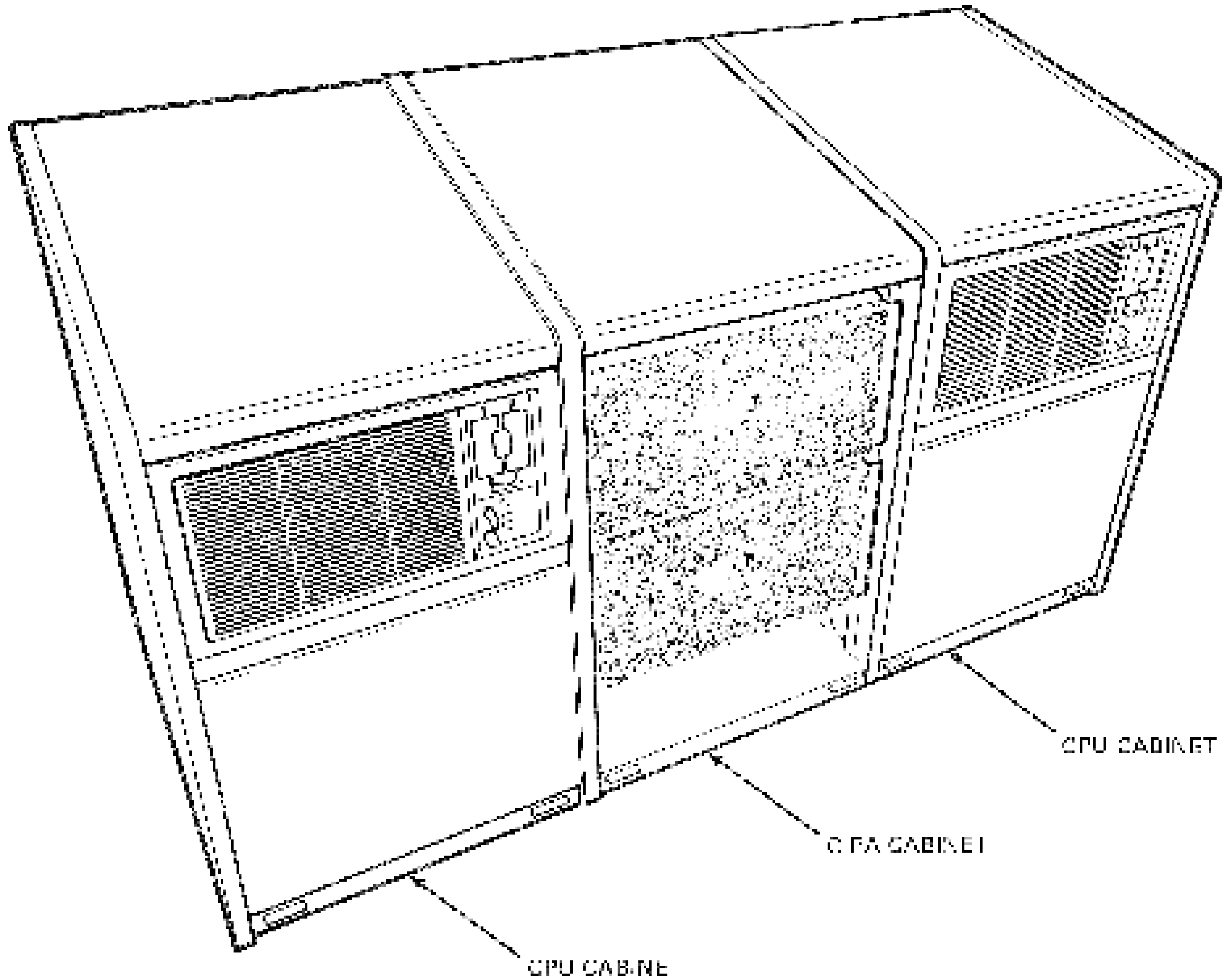


Figure 2-3 Dual CIBCI Adapter Configuration – VAX 8200/8300

2.4.1.3 Unpacking the H9642 CIPA Cabinet Carton –

Procedure:

1. Refer to Figure 2-4 and cut the two polyester straps.
2. Remove the cap, two ramps, and the cardboard spacer from the packaging container.
3. Lift and remove the cardboard tube and plastic bag covering the cabinet.
4. Refer to Figure 2-5 and with a 9/16 inch open-end wrench, remove the four shipping brackets located at the bottom of each corner of the cabinet.
5. Loosen the locking nuts on the four leveler feet located on the bottom corners of the cabinet and raise the leveler feet until the cabinet is resting on its casters.

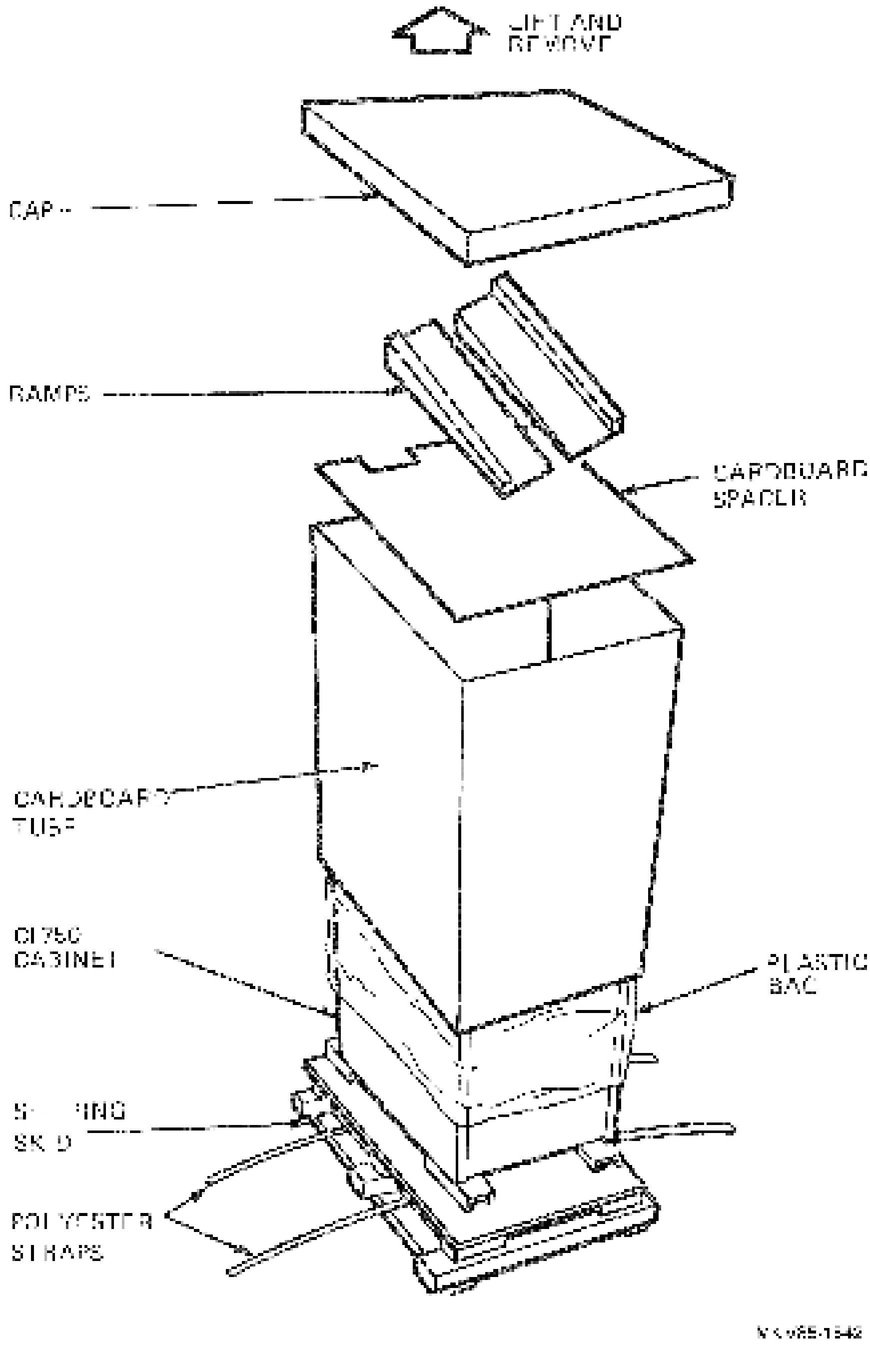


Figure 2-4 Unpacking the H9642 Cabinet

6. Allow a 3 m (10 ft) clearance from the back of the shipping pallet to remove the cabinet.

7. Attach the two ramps to the back of the shipping pallet by sliding the large end of the ramp into the groove of the pallet.

8. Ensure that the ramps are straight and secure. Then, gently roll the cabinet down the ramps.

2.4.2 VAX 8500/8550/8700/8800 Systems

2.4.2.1 Verifying Shipment Inventory –

Procedure:

1. Inventory all equipment against the shipping list accompanying the equipment.

2. Notify the customer of any opened cartons or boxes and document this fact on the installation report.

3. Notify the field service unit manager of any missing or incorrect items.

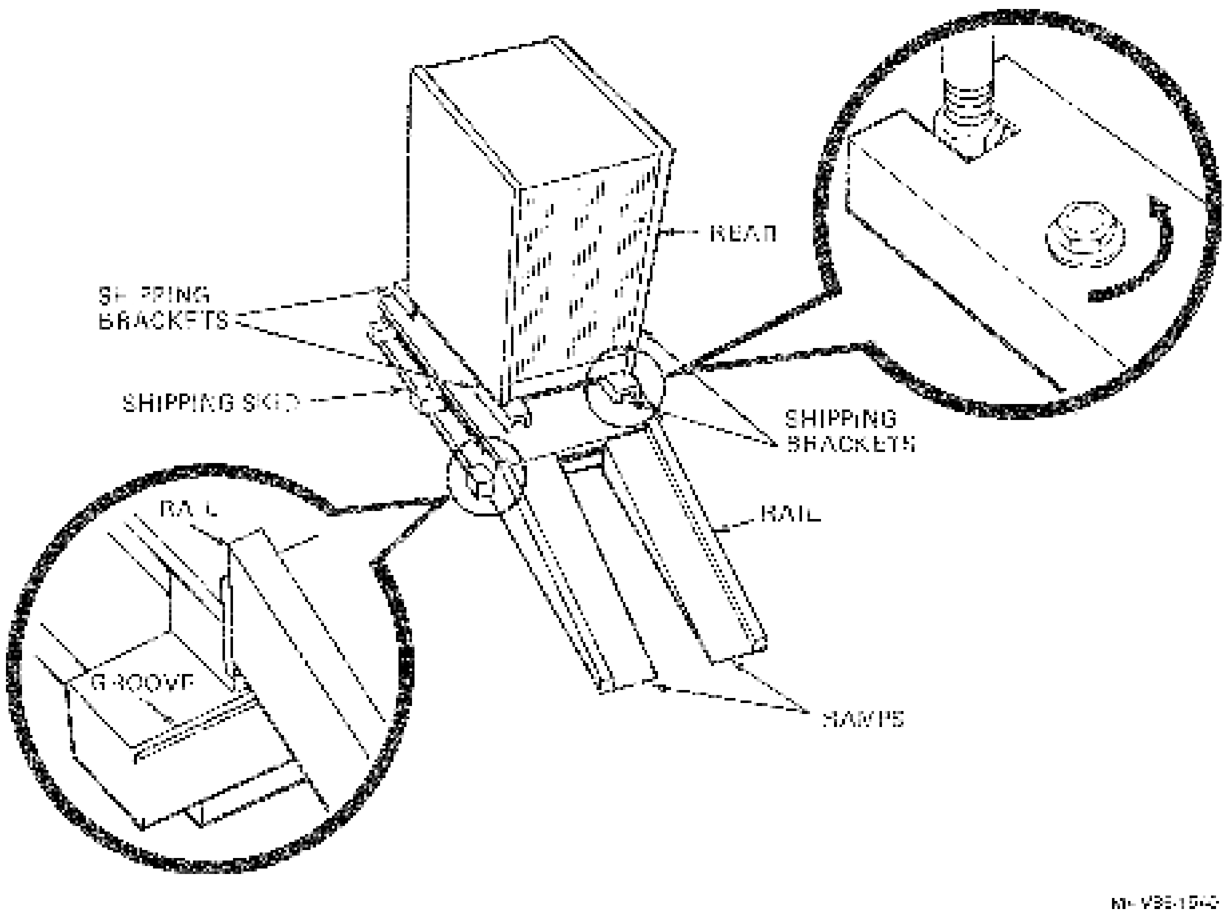


Figure 2-5 Removing the H9642 Cabinet from the Shipping Pallet

4. Request that the customer contact the shipping carrier to locate any missing items.

5. Request that the field service unit manager check with the Digital Equipment Corporation Traffic and Shipping Department if the shipping carrier does not have the missing items.

6. Check all boxes for external damage (dents, holes, or crushed corners).

7. Notify the customer of all damages and list all damages on the installation report.

2.4.2.2 Unpacking the Shipping Boxes

Procedure:

1. Locate the box marked "OPEN ME FIRST".

2. Open all boxes and inventory the contents against the shipping/accessory list in the "OPEN ME FIRST" box.

3. Inspect the equipment for damage. Report any damage and note it on the the installation report.

4. If damage is extensive, notify Digital Equipment Corporation for instructions on how to proceed.

2.4.2.3 Unpacking the H9652 Expander Cabinet

Procedure:

1. Refer to Figure 2-6 and cut the two polyester straps.

2. Remove the cap, two ramps, and the cardboard spacer from the packaging container.

3. Lift and remove the cardboard tube and plastic bag covering the cabinet.

4. Refer to Figure 2-7 and with a 9/16 inch open-end wrench, remove the four shipping brackets located at the bottom of each corner of the cabinet.

5. Loosen the locking nuts on the four leveler feet located on the bottom corners of the cabinet and raise the leveler feet until the cabinet is resting on its casters.

6. Allow a 3 m (10 ft) clearance from the back of the shipping pallet to remove the cabinet.

7. Attach the two ramps to the back of the shipping pallet by sliding the large end of the ramp into the groove of the pallet.

8. Ensure that the ramps are straight and secure. Then, gently roll the cabinet down the ramps.

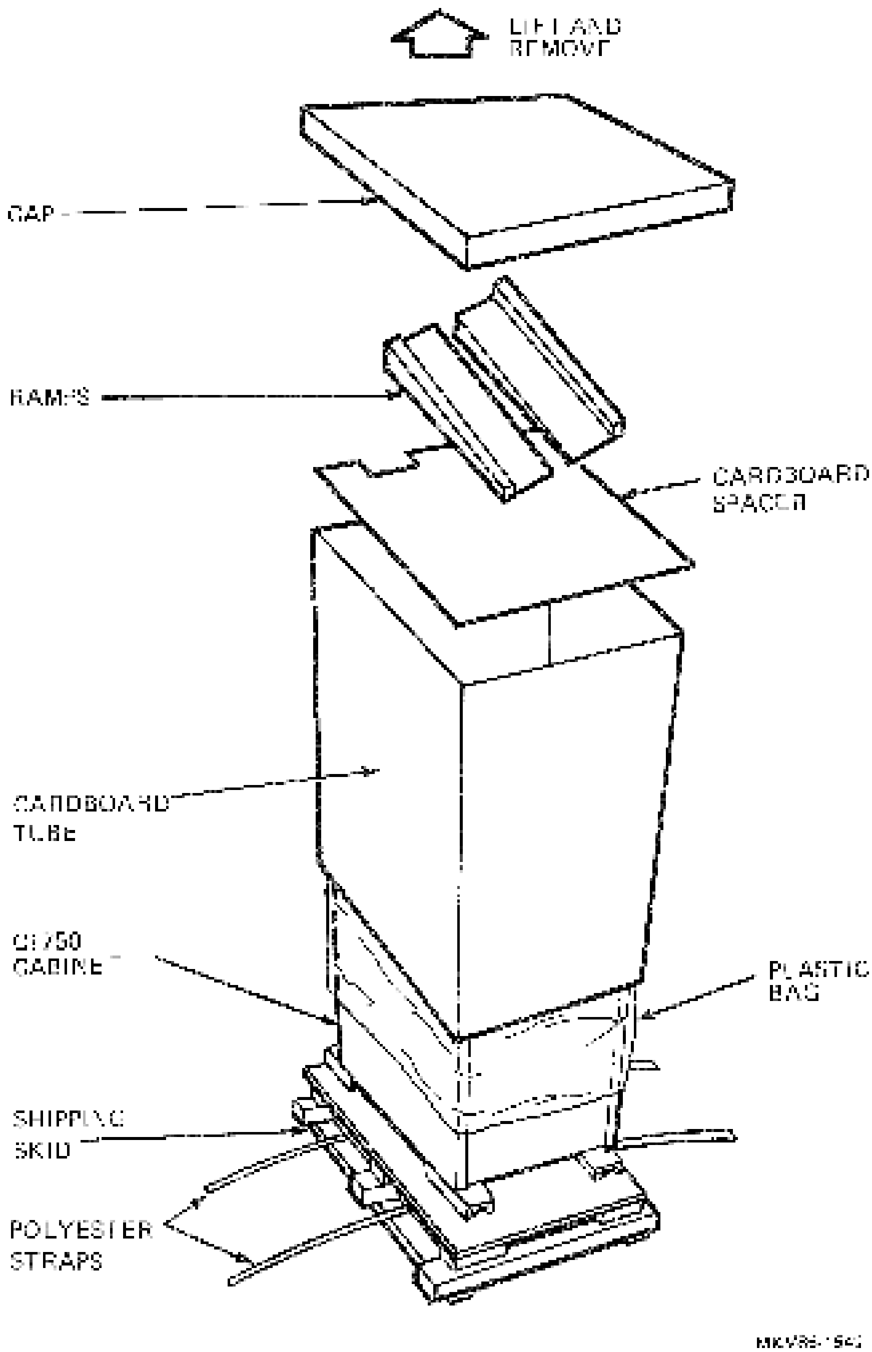


Figure 2-6 Unpacking the H9652 Cabinet

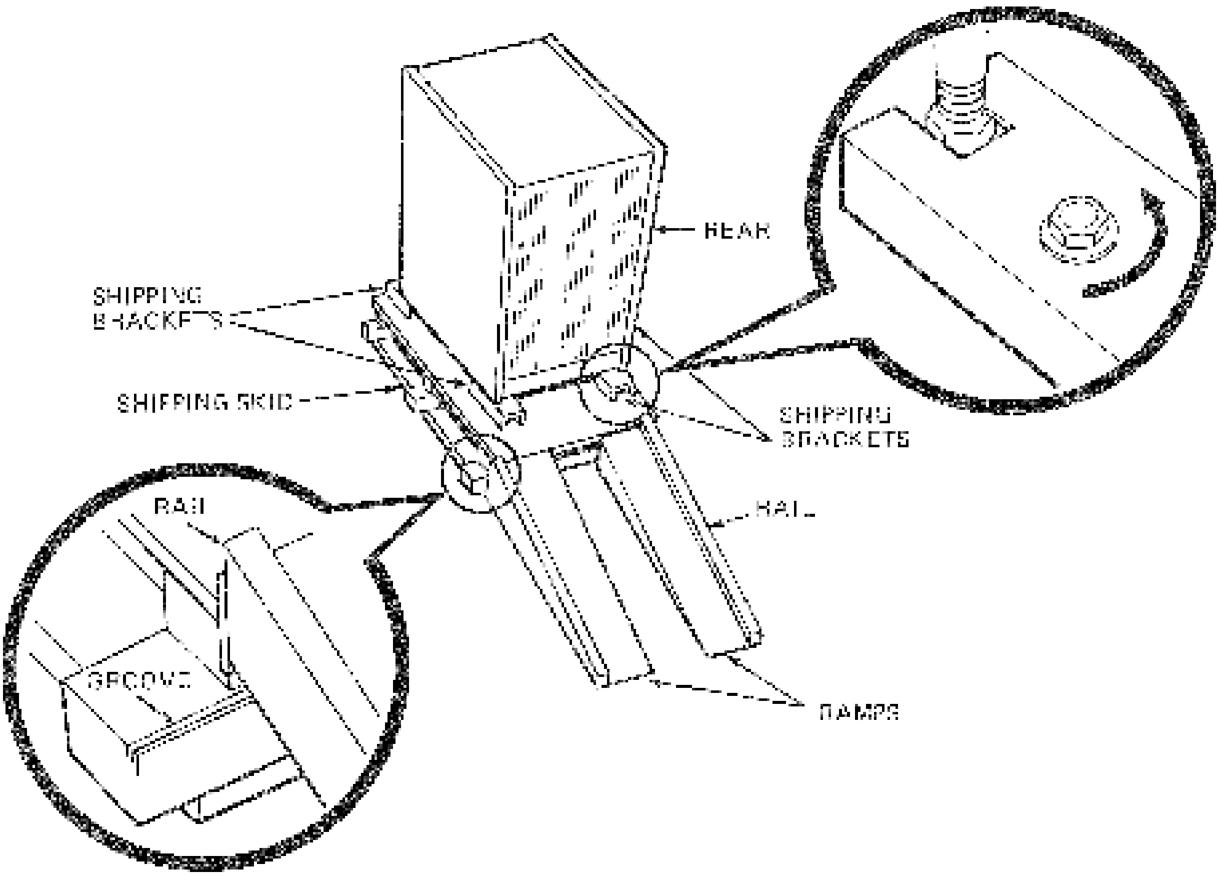


Figure 2-7 Removing the H9652 Cabinet from the Shipping Pallet

2.5 MECHANICAL INSTALLATION (System Level)

2.5.1 VAX 8200/8300 System

The system is shipped from the manufacturer with the T1017 and T1015 modules, the VAXBI node plug, and the four BCI cables already installed in the VAXBI cardcage.

The remaining CIBCI hardware resides in the H9642 CIPA cabinet. The CIPA cabinet may be configured either to the left or right of the processor cabinet. The procedures for joining the two cabinets are detailed in Sections 2.5.1.1 and 2.5.1.2.

For a CIBCI-FA/FB/HA/HB installation, repeat Steps 1 through 6 for the other VAX 8200 or VAX 8300 system.

2.5.1.1 CPU Cabinet Preparation

Procedure:

1. Refer to Figure 2-8 and then face the front of the cabinet. Carefully slide the BA32 mounting box fully outward from the cabinet.

CAUTION

Extend the stabilizer bar before sliding the BA32 mounting box from the cabinet. Failure to do so may cause personal injury if the cabinet tips forward when BA32 mounting box is fully extended.

Exercise care when extending the mounting box. Pass through the primary safety lock mechanism until the secondary safety lock mechanism is reached.

2. Remove the module access cover of the VAXBI cardcage by loosening the holding screws and lifting the cover.

3. Place the BA32 mounting box in a 90 degree position (see Figure 2-8) by grasping and pulling the slide-rail release mechanism, and then by rotating the mounting box up and towards the rear of the cabinet until it is securely locked into place.

4. Remove the bottom access cover (see Figure 2-8) by removing several holding screws and then by lifting the cover. Replace the screws in their holes for safekeeping.

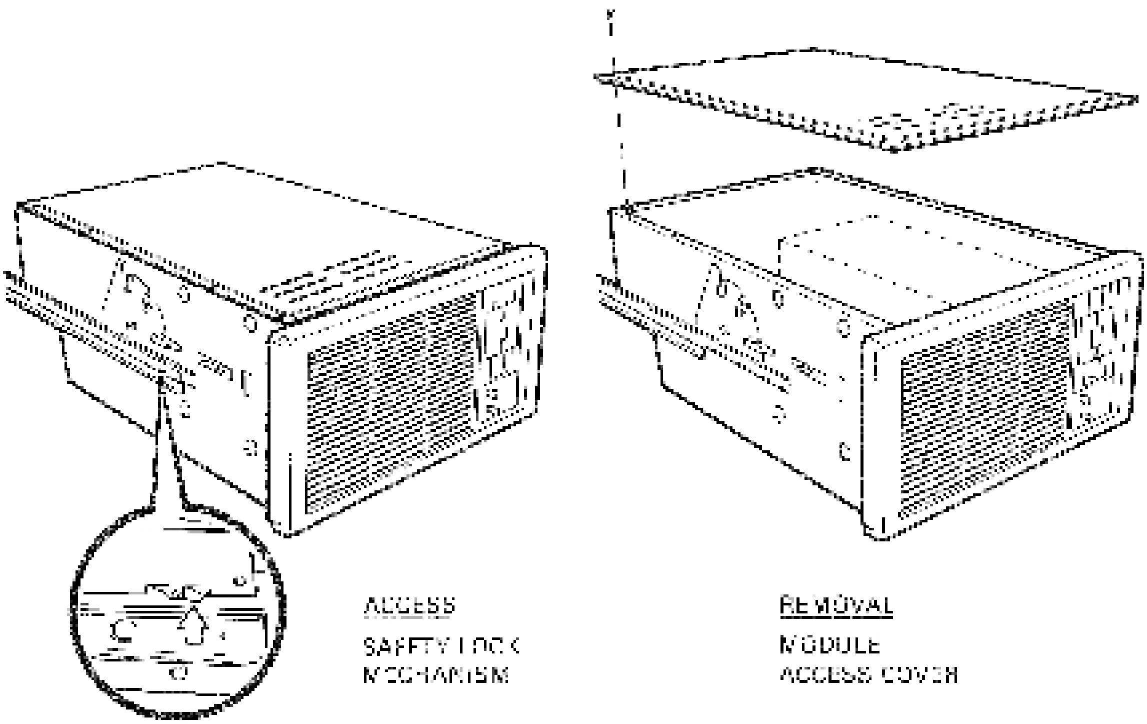


Figure 2-8 Accessing the BA32 CPU Mounting Box Hardware Components (Sheet 1 of 2)

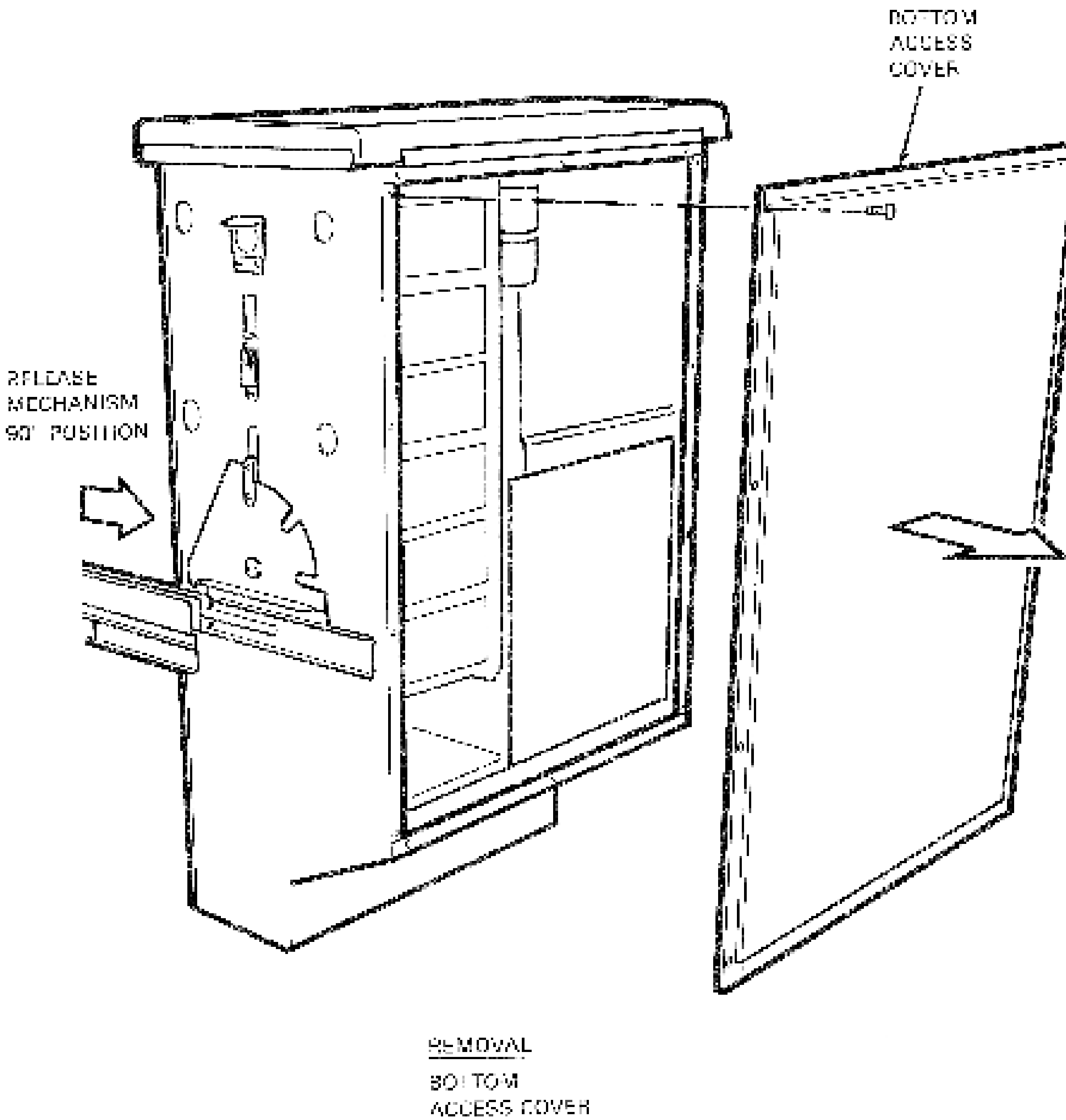


Figure 2-8 Accessing the BA32 CPU Mounting Box Hardware Components (Sheet 2 of 2)

5. Face the front of the cabinet and remove the end panel (the side where the H9642 CIPA cabinet will be joined) by grasping the panel at the front and rear, lifting it approximately 2.5 cm (1 in), and pulling it away from the cabinet (see Figure 2-9).

NOTE

If the cabinet is NOT resting on its wheels, loosen the locking nuts on the four leveler feet located at the bottom corners of the cabinet and raise the leveler feet until the cabinet is resting on its wheels.

6. Remove one of the two knockouts from the RFI shield panel by using a knife and cutting away the copper foil to expose the knockout opening.

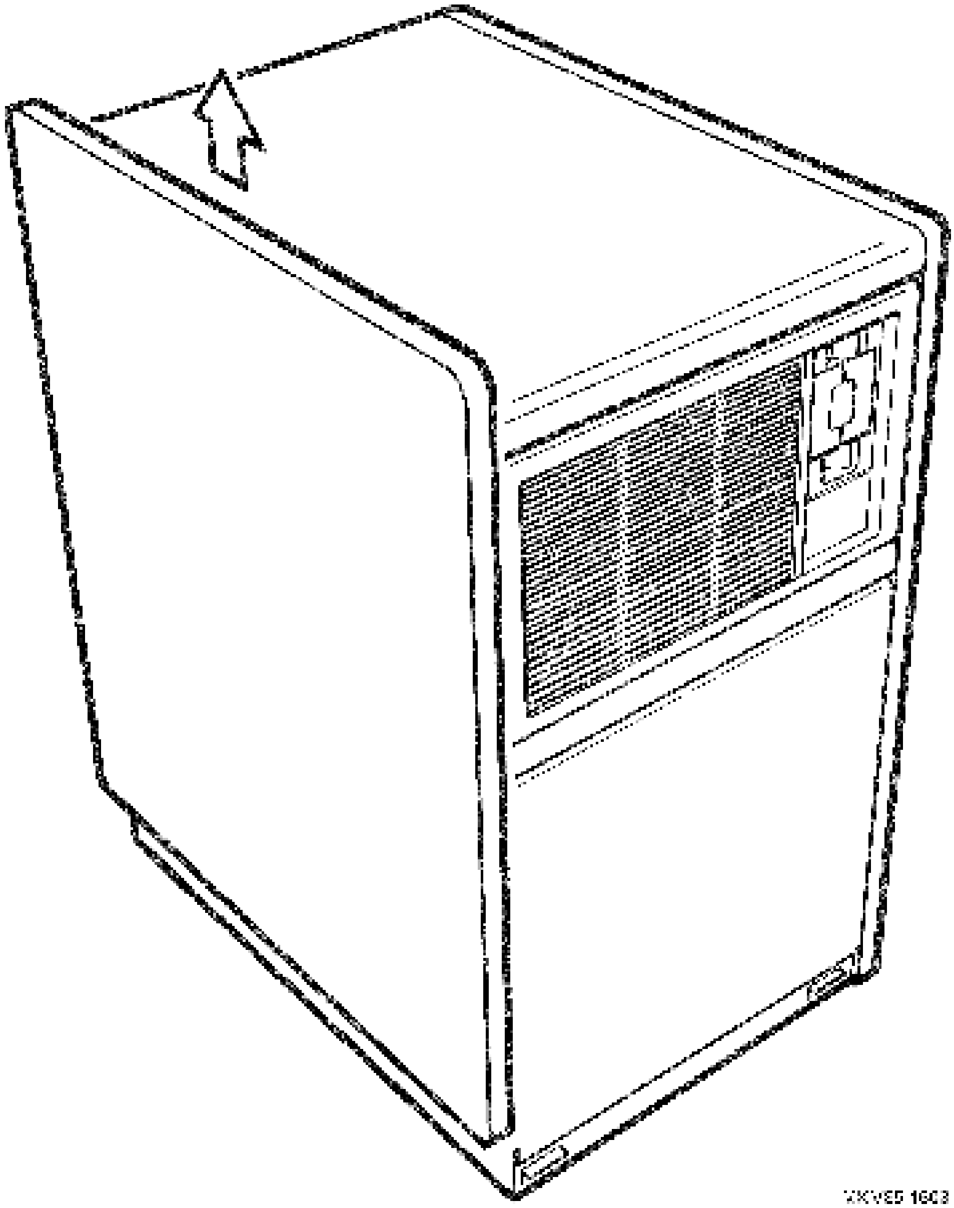


Figure 2-9 CPU Cabinet End Panel Removal

2.5.1.2 H9642 CIPA Cabinet Preparation – The CIPA (computer interconnect port adapter) hardware is housed in a shielded cabinet that, like the processor cabinet, has been specially designed to attenuate electromagnetic interference (EMI) and radio frequency interference (RFI) signals by absorbing radiated energy. Therefore, attention to the details of the mechanical installation procedure is vital when installing the CIPA hardware onto a VAX 8200 or VAX 8300 system.

For a CIBCI-FA/FB/HA/HB installation, repeat Steps 1 through 14 for the opposite side of the H9642 CIPA cabinet.

Procedure:

1. Using a 5/32 inch Allen wrench, remove the front and rear doors (see Figure 2-10).
2. Remove the expansion panel attached to the side of the cabinet that will be joined to the system cabinet by grasping it at the front and back and then lifting it up and away from the chassis (see Figure 2-11).

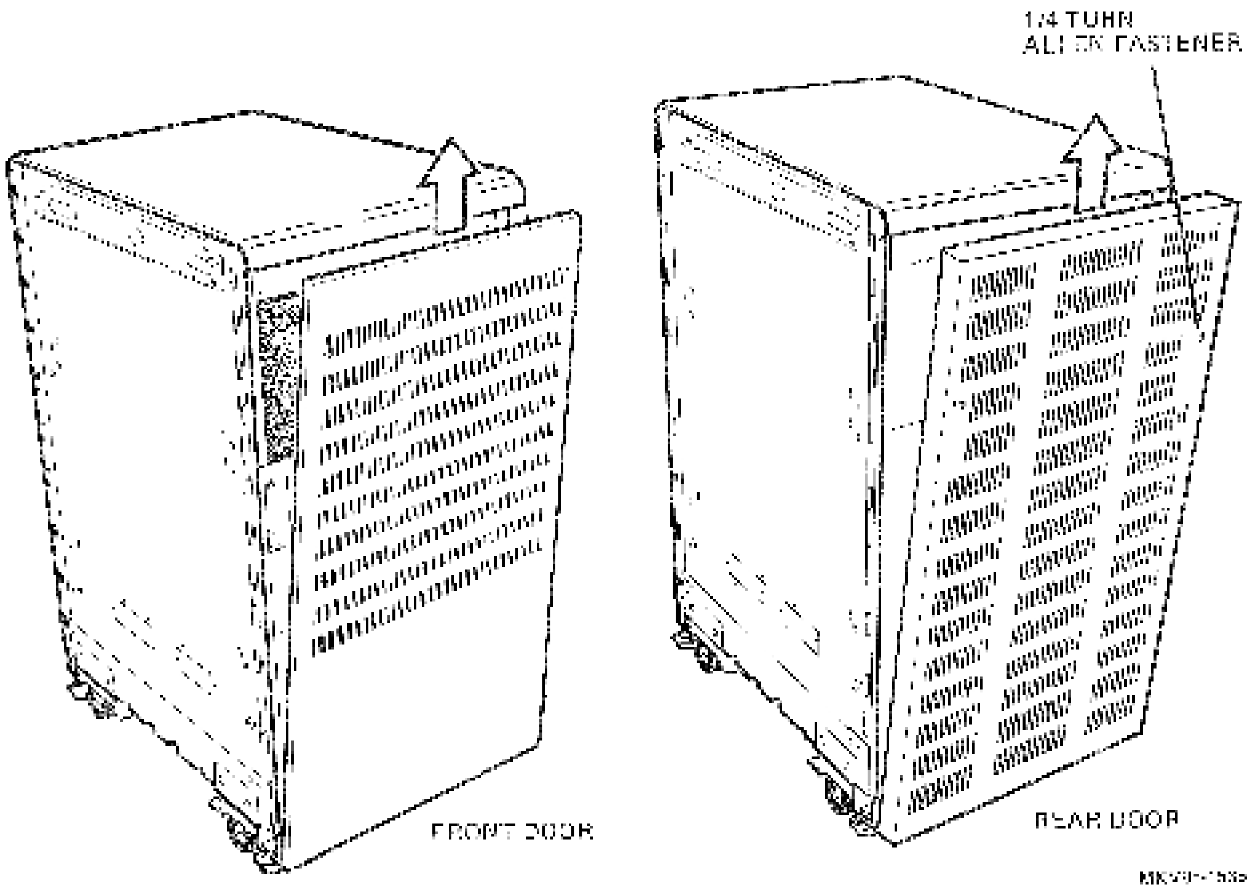


Figure 2-10 CIPA Cabinet – Front and Rear Panel Removal

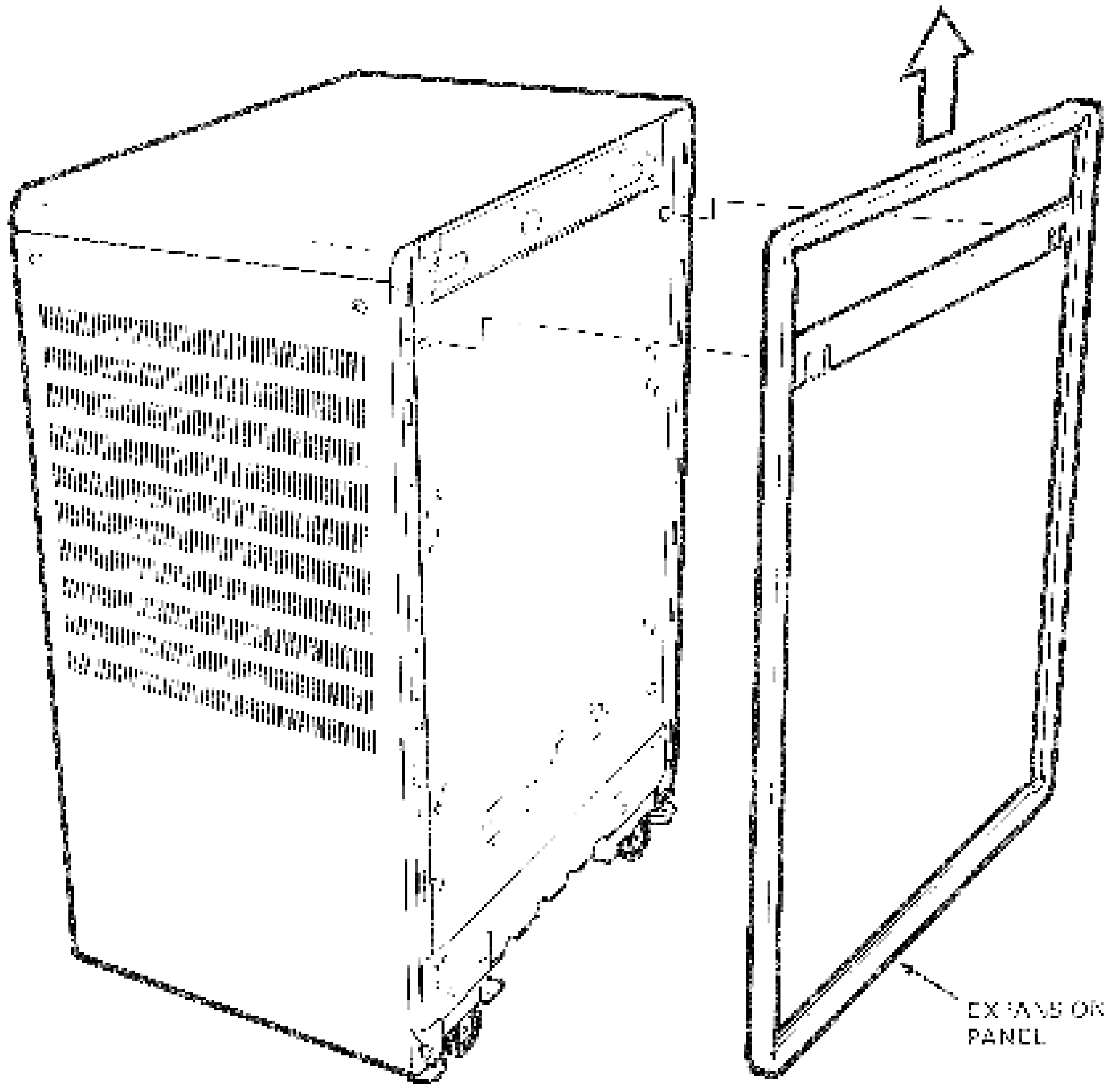


Figure 2-13 CIPA Cabinet – Expansion Panel Removal

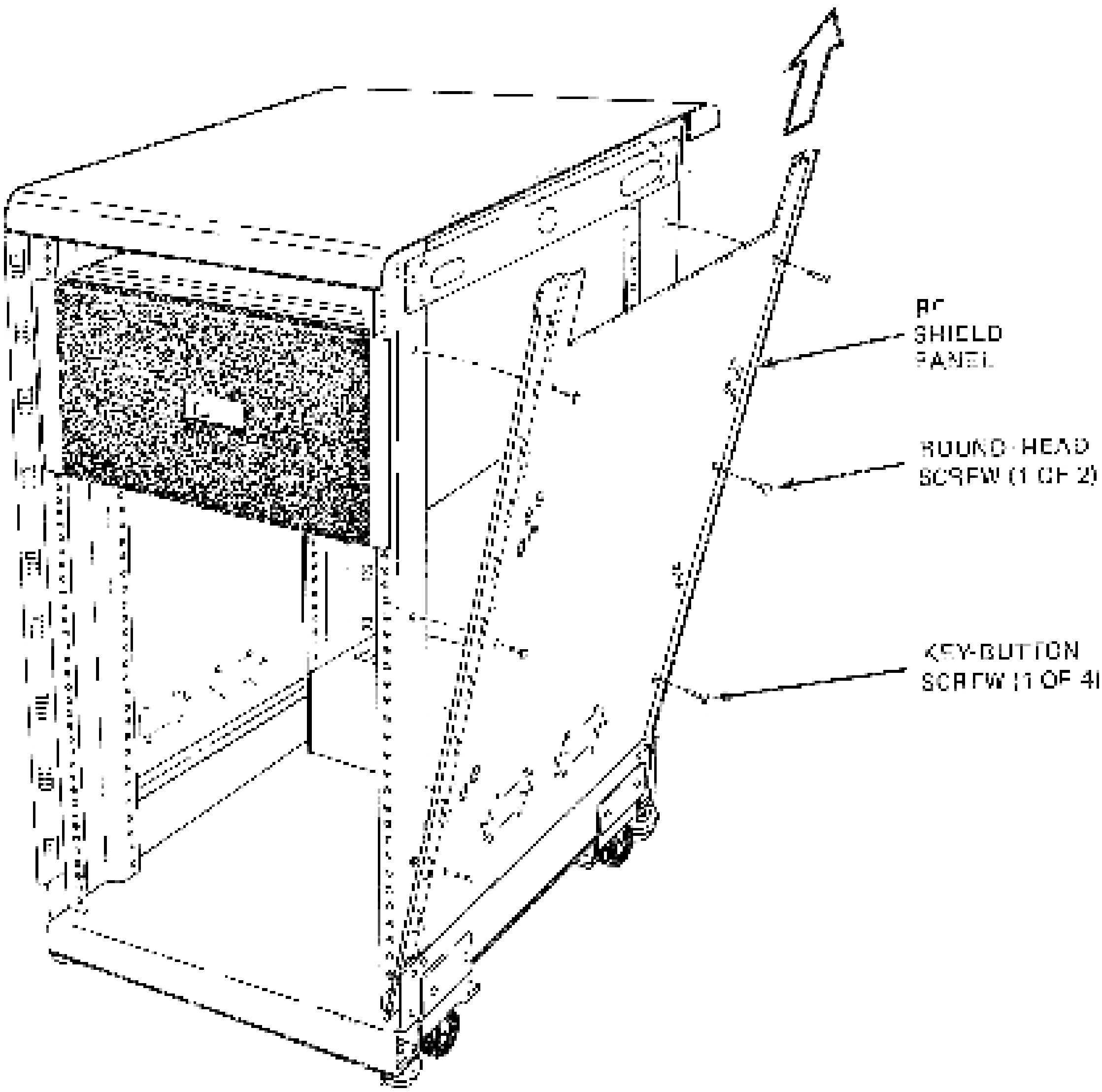


Figure 2-12 CIPA Cabinet – RFI Shield Panel Removal

3. Remove the RFI shield panel from the side of the cabinet where the H9642 CIPA cabinet will be joined (Figure 2-12), as follows:

NOTE

Bypass this procedural step if the knockout plug on the RFI shield panel is made of copper foil. The copper foil is removed using a knife.

CAUTION

Exercise care when handling the RFI shield panel to avoid damage to the RFI gasket springs located on its front and rear edges.

a. Remove and save the two round-head screws with lock washers.

b. Remove and save the four key-button screws.

c. Grasp the top of the RFI shield panel and pull it away from the chassis approximately .3 m (1 ft).

d. Lift the RFI shield panel up and away from the unit until the projecting legs located at the bottom of the panel are clear of the chassis.

4 Remove one of the two knockouts from the RFI shield panel (Figure 2-13), as follows:

NOTE
Bypass this procedural step if the knockout plug is made of copper foil. The copper foil is removed using a knife.

a. Place two wooden blocks (removed from the shipping pallet) on the floor and lay the RFI shield panel down with the knockouts positioned over the wooden blocks.

b. Using a hammer and a flat-blade screwdriver, break the upper edge of each knockout free of the RFI shield panel.

c. Lift the RFI shield panel off the wooden blocks and push inward on the upper edge of each knockout until the lower edges break free.

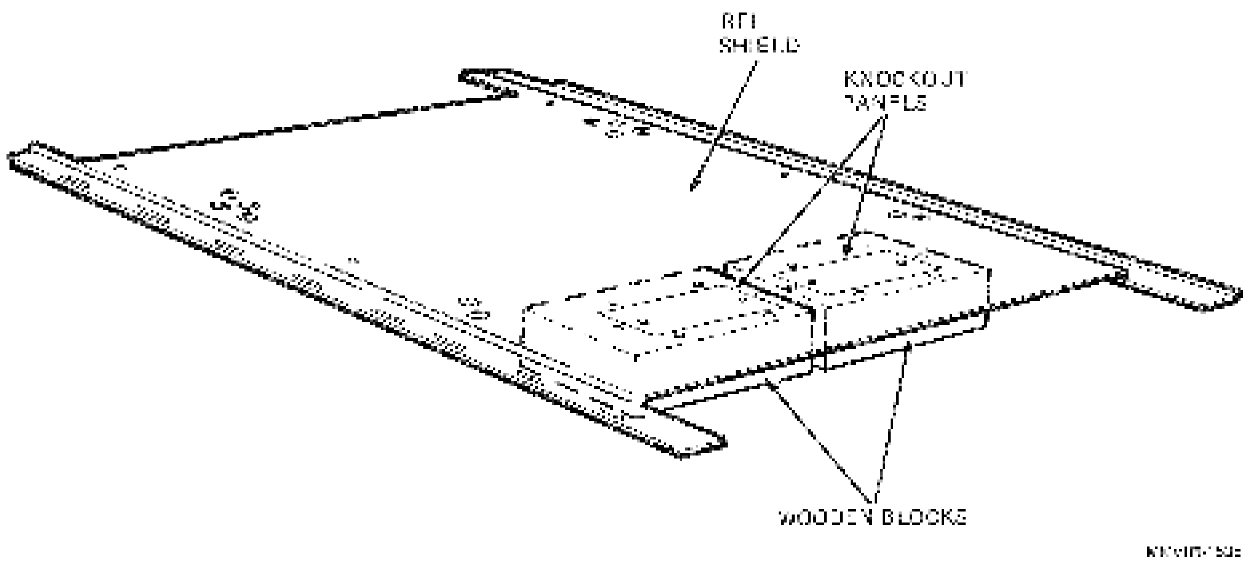


Figure 2-13 RFI Shield Panel – Knockout Plug Removal

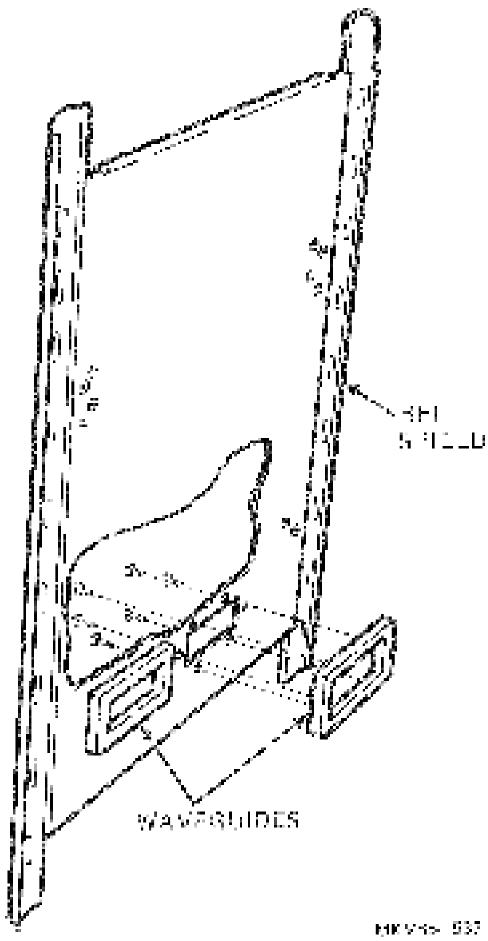


Figure 2-14 RFI Shield Panel – Waveguide Installation

5. Install one of the two waveguides onto the RFI shield panel (see Figure 2-14).

   a. Place one of the waveguides from the waveguide/joiner bar kit (P/N H9544-JE) over a knockout hole on the outside of the RFI shield panel.

   b. Align the screw holes in the waveguide with the screw holes in the RFI shield panel. The waveguide lip should be positioned into the knockout hole.

   c. Insert six 8-32 x 1/4 inch screws into the waveguide screw holes from the inside of the RFI shield panel and then tighten the screws.

6. Place the RFI shield panel in an upright position and carefully lean it against the side of the cabinet where the H9642 CIPA cabinet will be joined. Install the RFI shield panel back onto the side of the H9642 CIPA cabinet by reversing the procedure detailed in Step 3.

CAUTION

Avoid damage to the RFI gasket springs on the shield panel. Use care when inserting the projecting legs of the shield panel over the lower part of the chassis frame at the front and back.

7. Install the expansion panel on the side of the cabinet where the CPU cabinet will be joined by reversing the procedure detailed in Step 2.

CAUTION

Be sure to use the expansion panel shipped with the CIBCI option. This has a single (upper) locking bar. Do NOT use an expansion panel that has both upper and lower locking bars.

NOTE

The longer set of key slots on the expansion panel should be attached to the H9642 CIPA cabinet.

8. Move the cabinet adjacent to the CPU cabinet. Leave approximately 1 m (3 ft) between the cabinets to allow access to the facing sides.

9. From the rear-inside of the H9642 CIPA cabinet, locate the free end of the flat ribbon CIPA bus cable (see Figure 2-15).

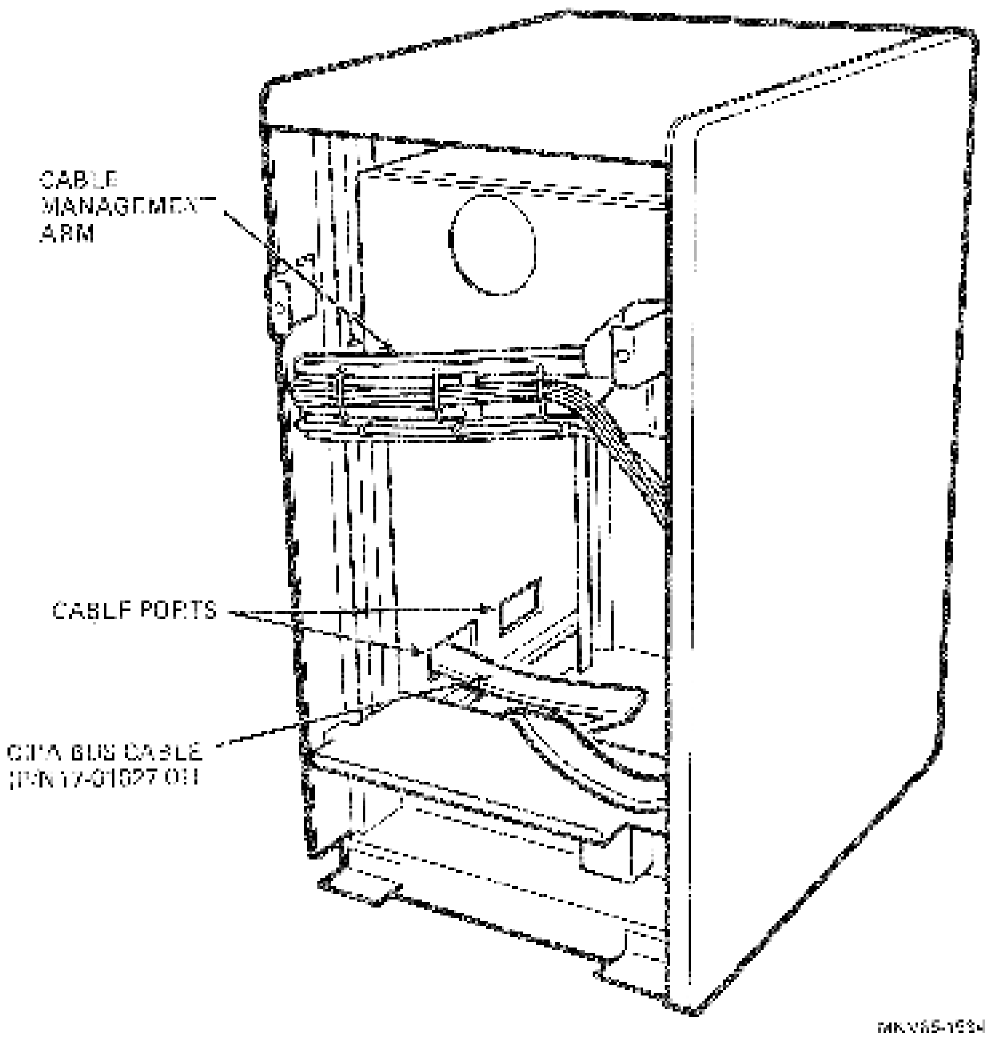


Figure 2-15 H9642 Cabinet – Rear Interior View

10. Refer to Figure 2-16 and perform the following:

    a. Carefully move the H9642 CIPA cabinet alongside the CPU cabinet. Take care that the CIPA bus cable does not bunch up between the cabinets.

    b. Route the CIPA bus cable through the RFI shield panel knockout hole and waveguide, and then through the RFI shield panel knockout hole of the CPU cabinet.

    c. Route the CIPA bus cable up through the interior of the CPU cabinet to the bottom of the BA32 mounting box and then outward from the cabinet. If necessary, move the H9642 CIPA cabinet closer to the CPU cabinet to allow the CIPA bus cable to reach.

    d. Remove the cable strain relief clamp on the BA32 mounting box.

    e. Route the CIPA bus cable through the I/O cable slot located in the lower rear of the BA32 mounting box.

11. Refer to Figure 2-17 and, working from the exterior of the H9642 CIPA cabinet:

    a. Raise the expansion panel on the side of the H9642 CIPA cabinet approximately 2.5 cm (1 in).

    b. While holding the cabinets together, push the expansion panel down slightly until key slots just begin to engage the upper buttons on both cabinets at the front and rear.

    c. Push down firmly on the expansion panel to securely lock the cabinets together.

    d. Bolt the cabinets together at the front using one of the joiner bars provided.

    e. Bolt the cabinets together at the rear using the second joiner bar.

    f. Install the end panel previously removed from the CPU cabinet onto the open side of the H9642 CIPA cabinet by reversing the procedure used in Step 5.

12. Route and connect the 70-08288-06 power control bus cable between the CPU and H9642 CIPA cabinet power controller assemblies.

13. Place the power controller's local/remote switch in its remote position.

14. Route, but at this time do NOT connect the BNCIA-xx coaxial (CI bus) cables. Refer to the *Star Coupler User's Guide*.

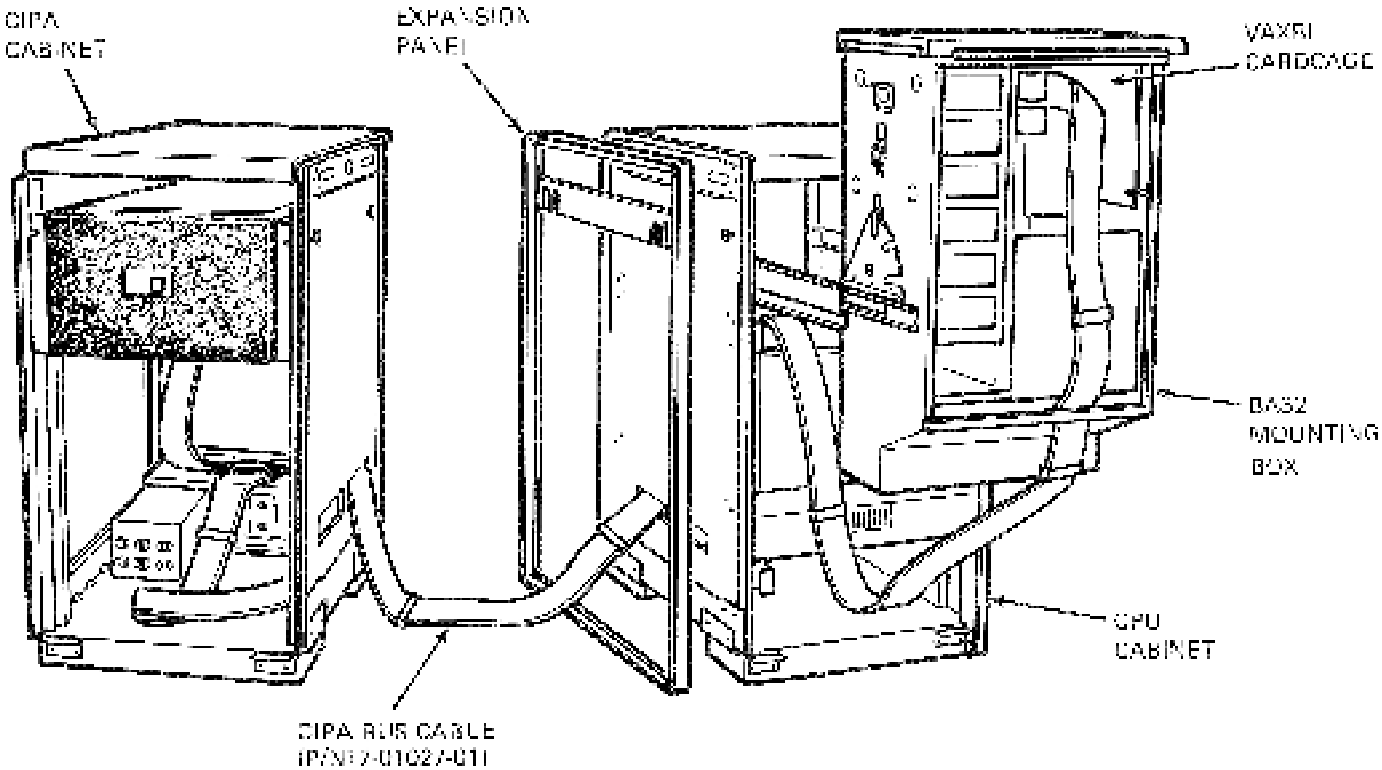


Figure 2-16 CIPA Bus Cable Routing

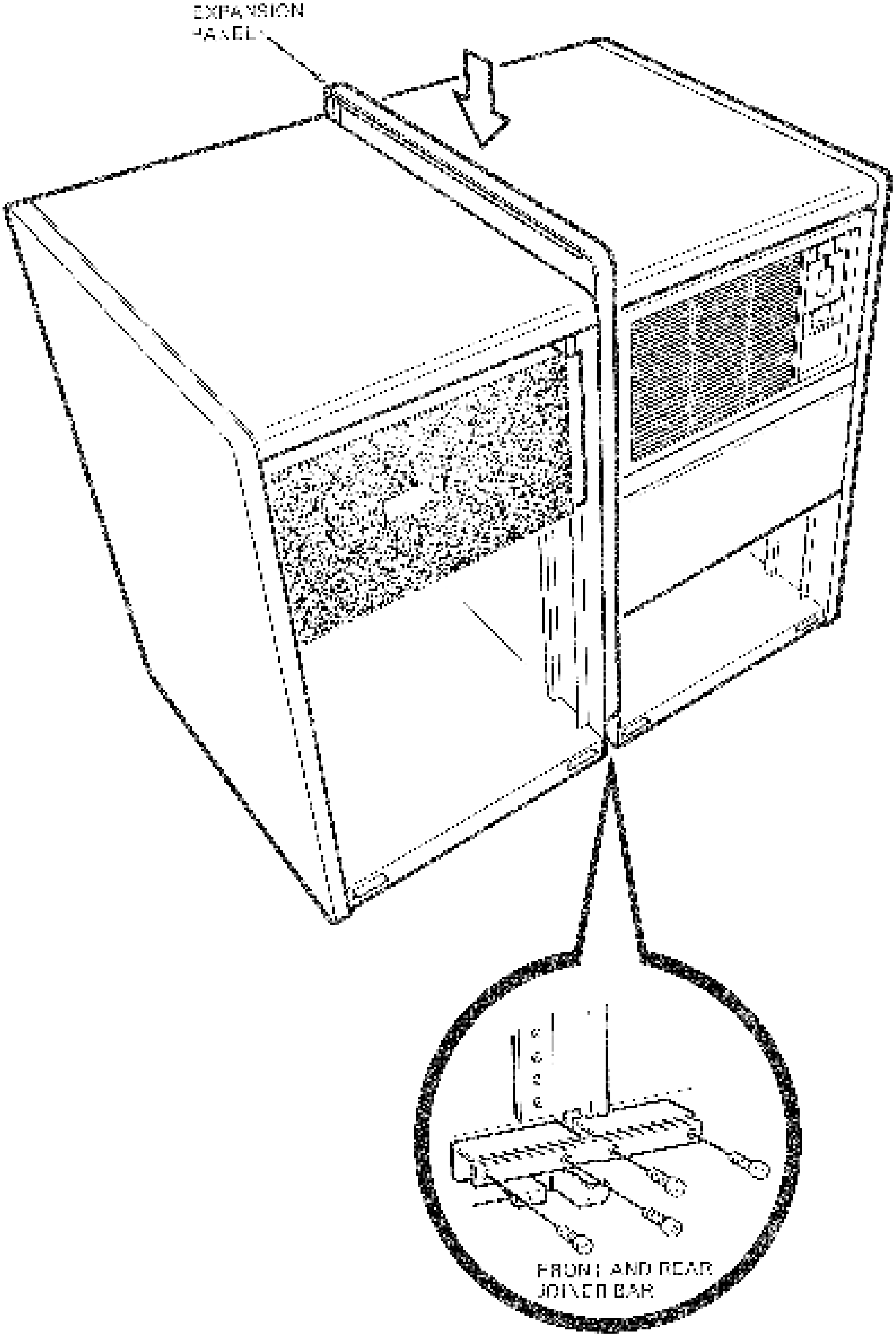


Figure 2-17 CPU and CPA Cabinet Mating

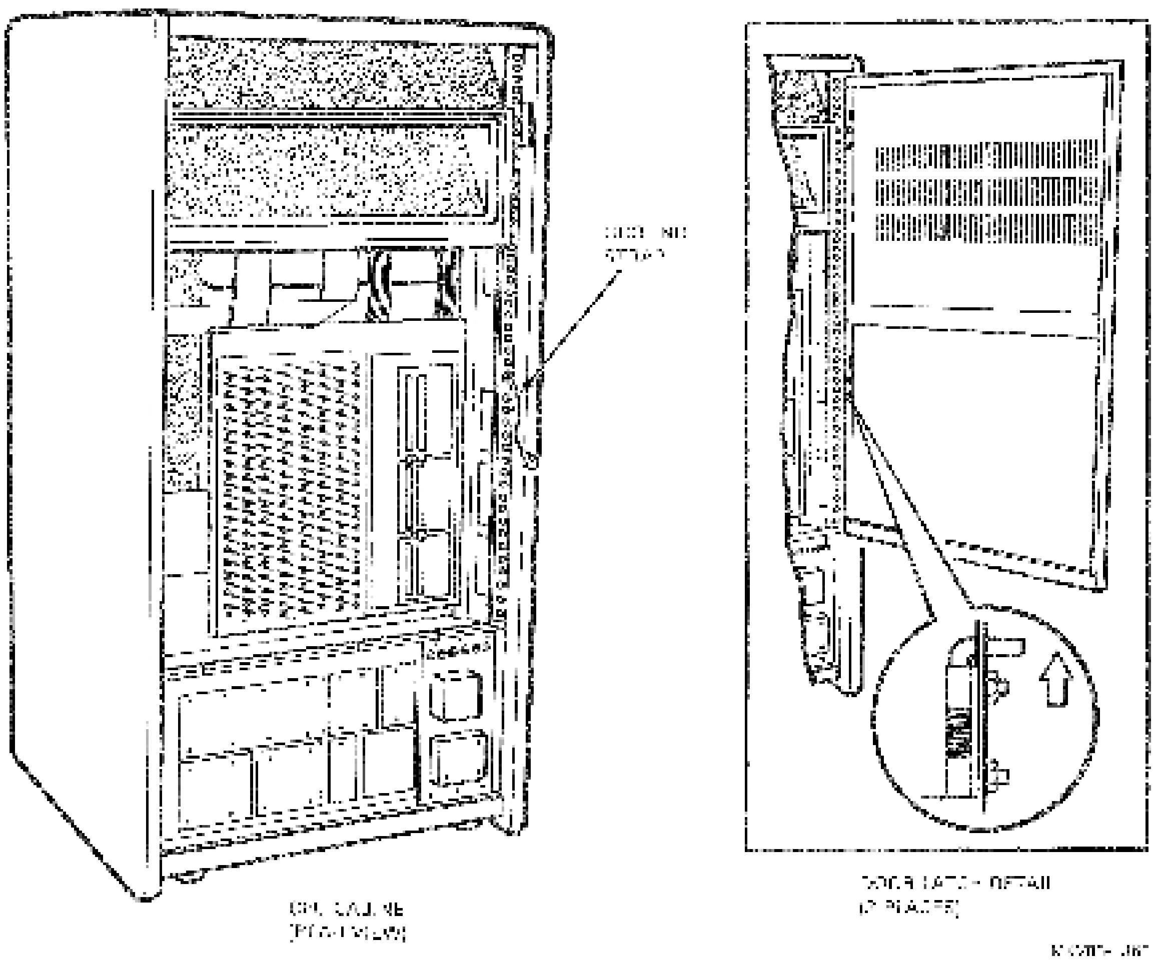


Figure 2-18 CPU and Expansion Cabinet Mating (Sheet 1 of 5)

2.5.2 VAX 8500/8550 Systems

The system is shipped from the manufacturer with the T1017 and T1018 modules, the VAXBI node plug, and the four the BCI cables already installed in the VAXBI cardcage.

The remaining CIBCI hardware resides in an expansion cabinet. The expansion cabinet always mounts to the right of the processor cabinet. The procedures for joining the two cabinets are detailed below.

Procedure: (Refer to Figure 2-18)

1. Remove the CPU cabinet front door by removing a single hex screw that secures the door's ground strap to the cabinet frame, pulling the release mechanism on the two door hinges, and then lifting the door off its hinges.

2. Remove the CPU cabinet rear door by removing a single hex screw that secures the door's ground strap to the cabinet frame, pulling the release mechanism on the two door hinges, and then lifting the door off its hinges.

3. Working from the rear of the CPU cabinet, remove the top cabinet cover by removing the two screws located on the underside of the top cover.

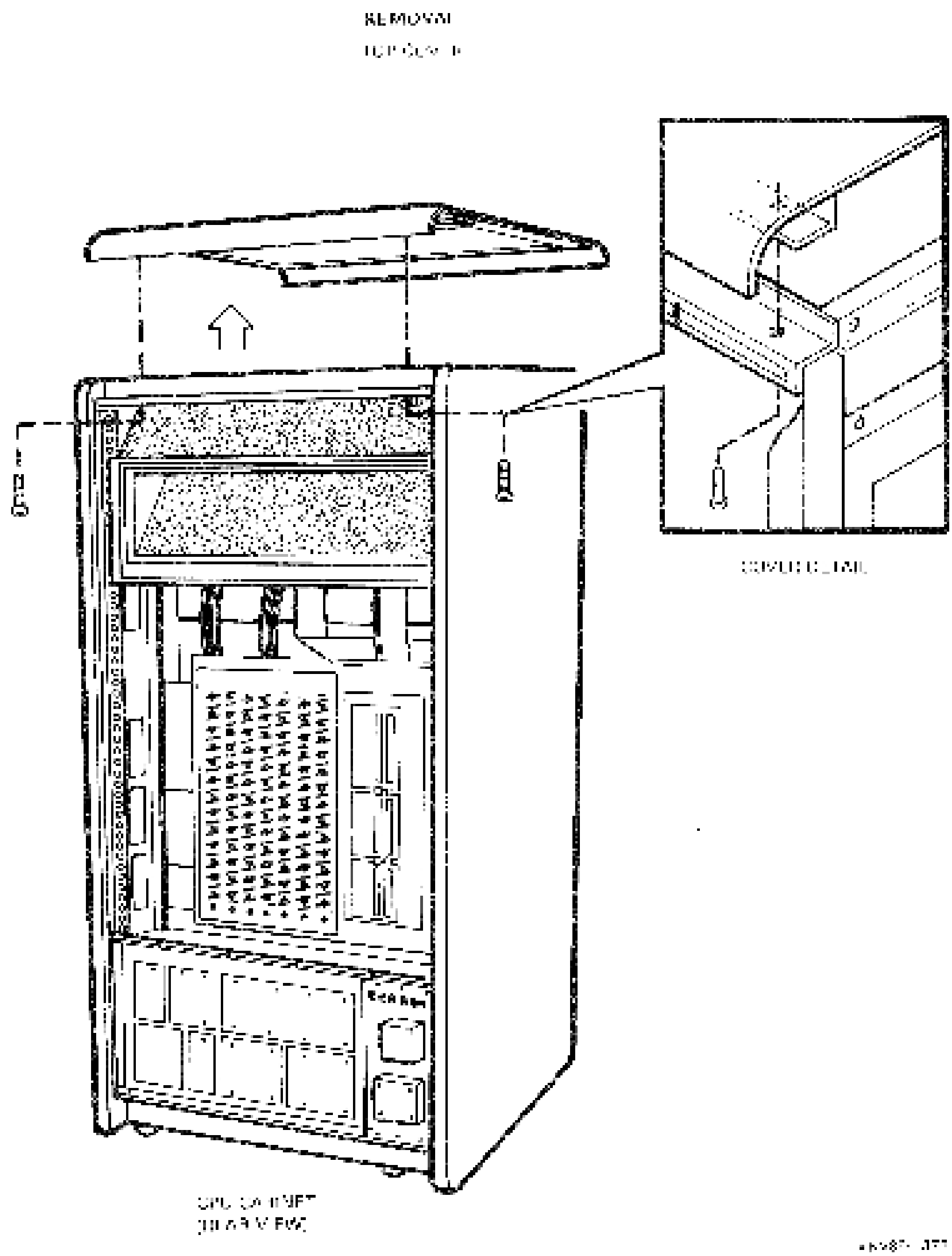


Figure 2-18 CPU and Expansion Cabinet Mating (Sheet 2 of 5)

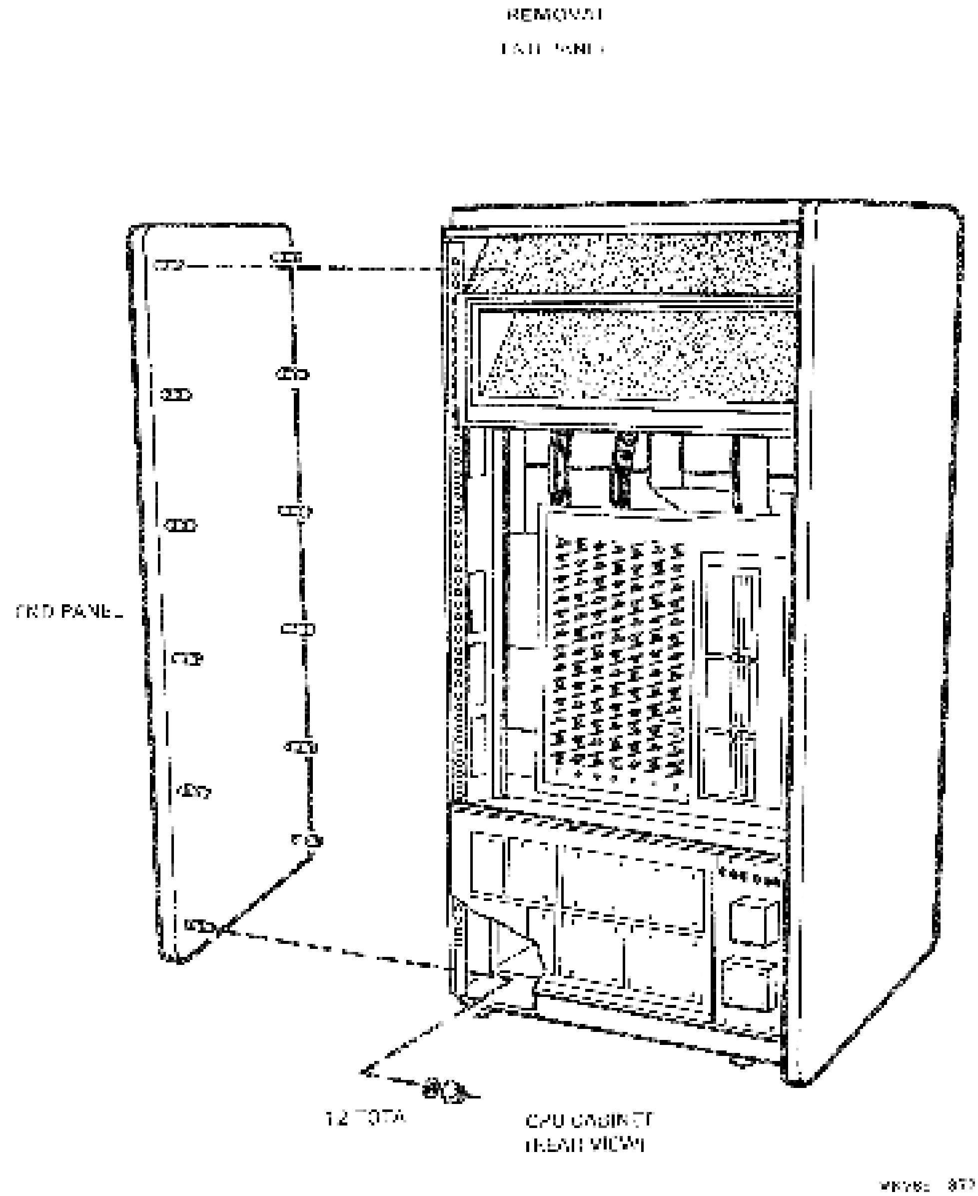


Figure 2-18 CPU and Expansion Cabinet Mating (Sheet 3 of 5)

4. Remove the rear-left end panel from the CPU cabinet by removing the twelve kepnuts (six on each side).

5. Open the front and rear doors of the H9652 expansion cabinet.

6. Install the CPU end panel on the far side of the H9652 expansion cabinet by securing twelve kepnuts (six on each side).

7. Join the two cabinets by aligning the twelve studs and holes on the cabinet frames, and then by replacing the twelve kepnuts (six on each side).

8. Working from the rear of the CPU cabinet, replace the top cabinet cover by inserting two screws on the underside of the top cover.

9. Replace the CPU cabinet front door by pulling the release mechanism on the two door hinges and by lifting the door onto its hinges. Then, insert the single hex screw that secures the door's ground strap to the cabinet frame.

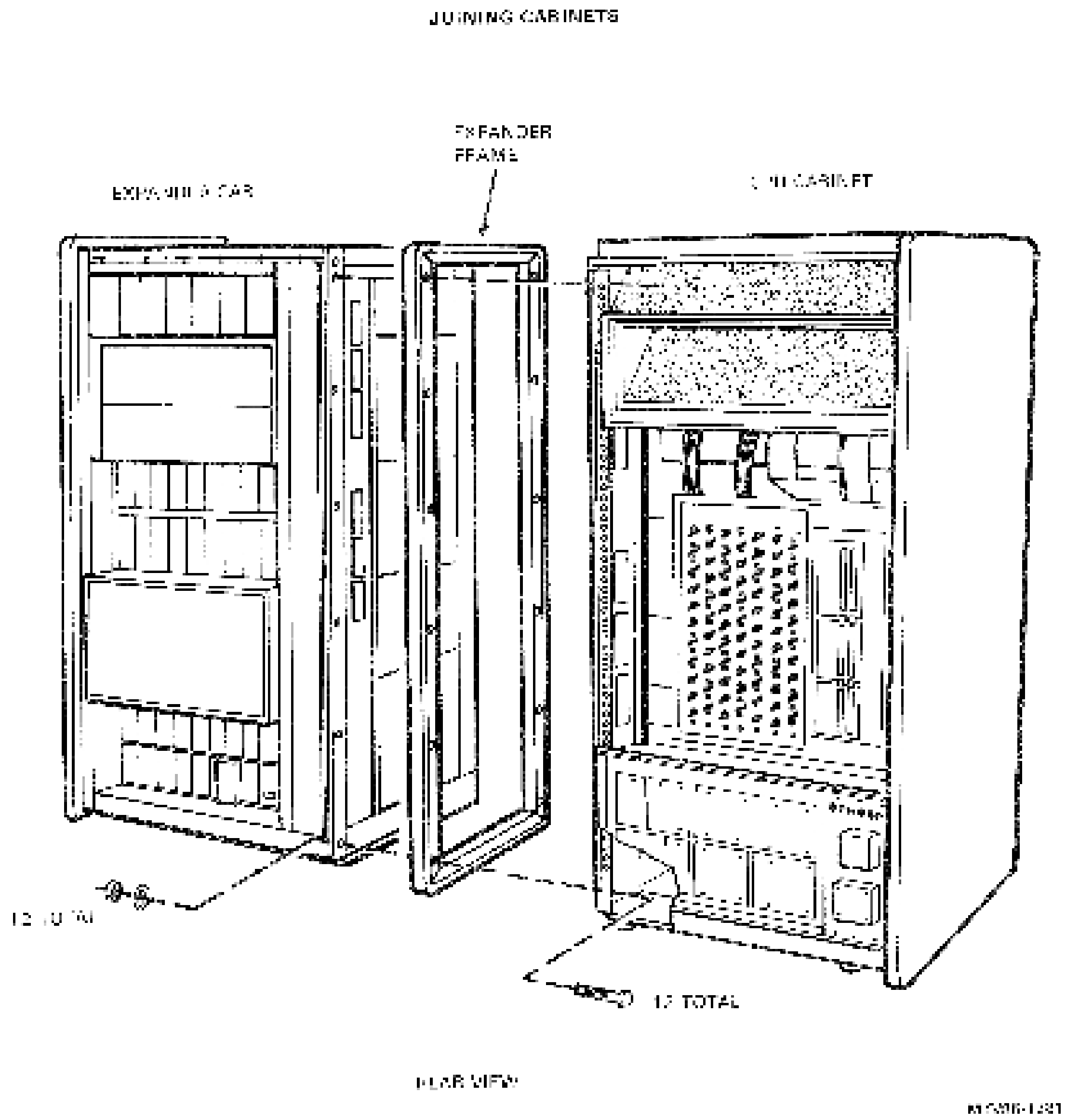


Figure 2-18 CPU and Expansion Cabinet Mating (Sheet 4 of 5)

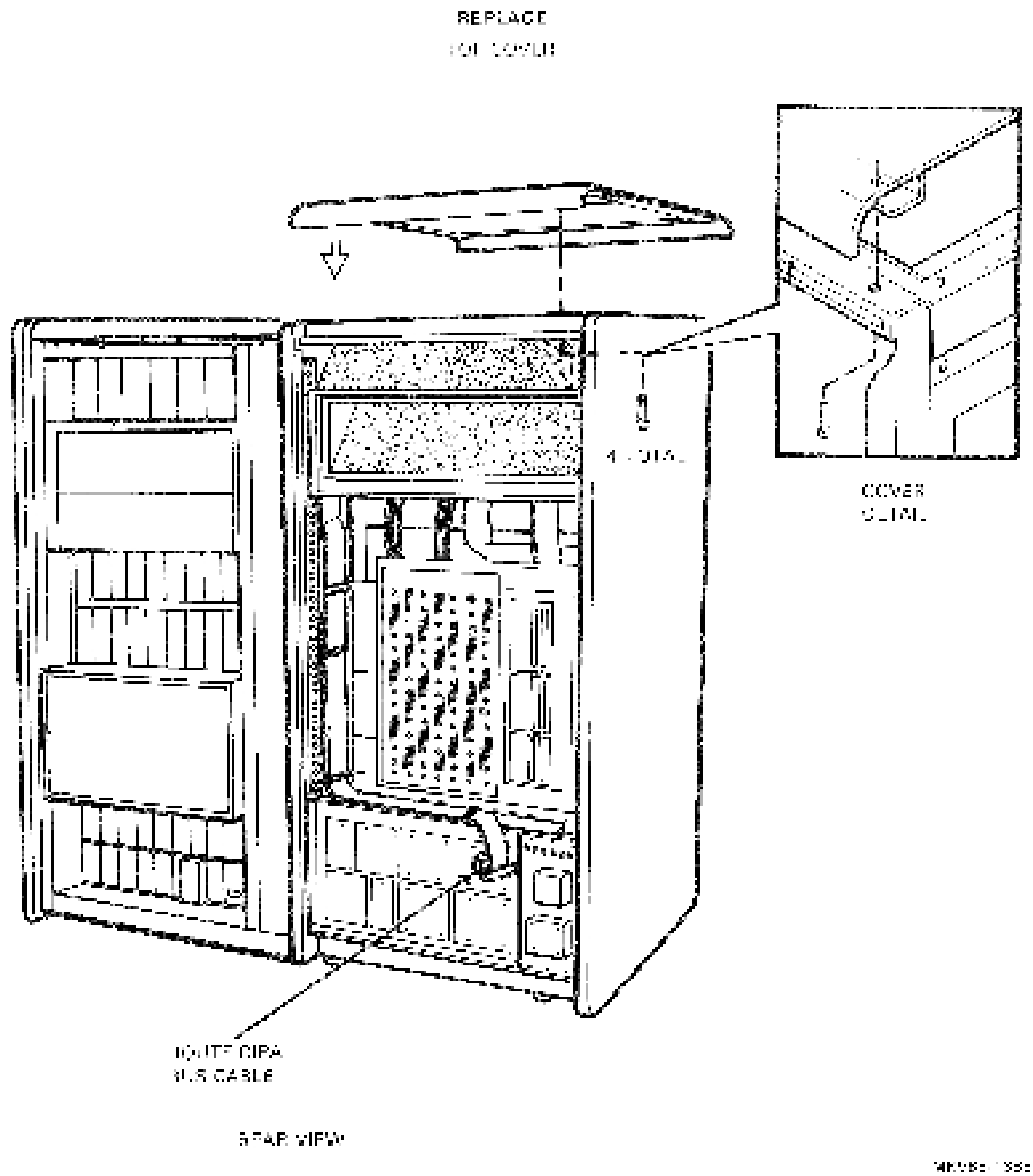


Figure 2-18 CPU and Expansion Cabinet Mating (Sheet 5 of 5)

10. Replace the CPU cabinet rear door by pulling the release mechanism on the two door hinges and by lifting the door onto its hinges. Then, insert the single hex screw that secures the door's ground strap to the cabinet frame.

11. Route the CIPA bus cables from the H9652 expansion cabinet to the VAXBI cardcage located in the left-rear of the CPU cabinet.

12. Refer to the *VAX 8500/8550 Installation Guide*, part number EK-8500I-IN, for additional intracabinet cabling.

2.5.3 VAX 8700/8800 Systems

The system is shipped from the manufacturer with the T1017 and T1018 modules, the VAXBI node plug, and the four the BCI cables already installed in the VAXBI cardcage.

The remaining CIBCI hardware resides in either a front-end cabinet or an expansion cabinet. The front-end cabinet always mounts to the left of the processor cabinet. The expansion cabinet always mounts to the right of the processor cabinet. The procedure for joining the cabinets is detailed below.

Procedure: (Refer to Figure 2-19)

1. Remove the appropriate CPU cabinet front door by removing a single hex screw that secures the door's ground strap to the cabinet frame, pulling the release mechanism on the two door hinges, and then lifting the door off its hinges.

2. Remove the appropriate CPU cabinet rear door by removing a single hex screw that secures the door's ground strap to the cabinet frame, pulling the release mechanism on the two door hinges, and then lifting the door off its hinges.

3. Working from the rear of the CPU cabinet, remove the top cabinet cover by removing the two screws located on the underside of the top cover.

4. Remove the appropriate end panel from the CPU cabinet by removing the twelve kepnuts (six on each side).

5. Open the front and rear doors of the H9652 cabinet.

6. Install the CPU end panel on the far side of the H9652 cabinet by securing twelve kepnuts (six on each side).

7. Join the two cabinets by aligning the twelve studs and holes on the cabinet frames, and then by replacing the twelve kepnuts (six on each side).

8. Working from the rear of the cabinet, replace the top cabinet cover by inserting two screws on the underside of the top cover.

9. Replace the appropriate CPU cabinet front door by pulling the release mechanism on the two door hinges and by lifting the door onto its hinges. Then, insert the single hex screw that secures the door's ground strap to the cabinet frame.

10. Replace the appropriate CPU cabinet rear door by pulling the release mechanism on the two door hinges and by lifting the door onto its hinges. Then, insert the single hex screw that secures the door's ground strap to the cabinet frame.

11. Route the CIPA bus cables from the H9652 cabinet to the VAXBI cardcage located in the left-rear of the CPU cabinet (see Figure 2-20).

12. Refer to the *VAX 8800/8700 Installation Guide*, part number EK-8800I-IN, for additional intracabinet cabling.

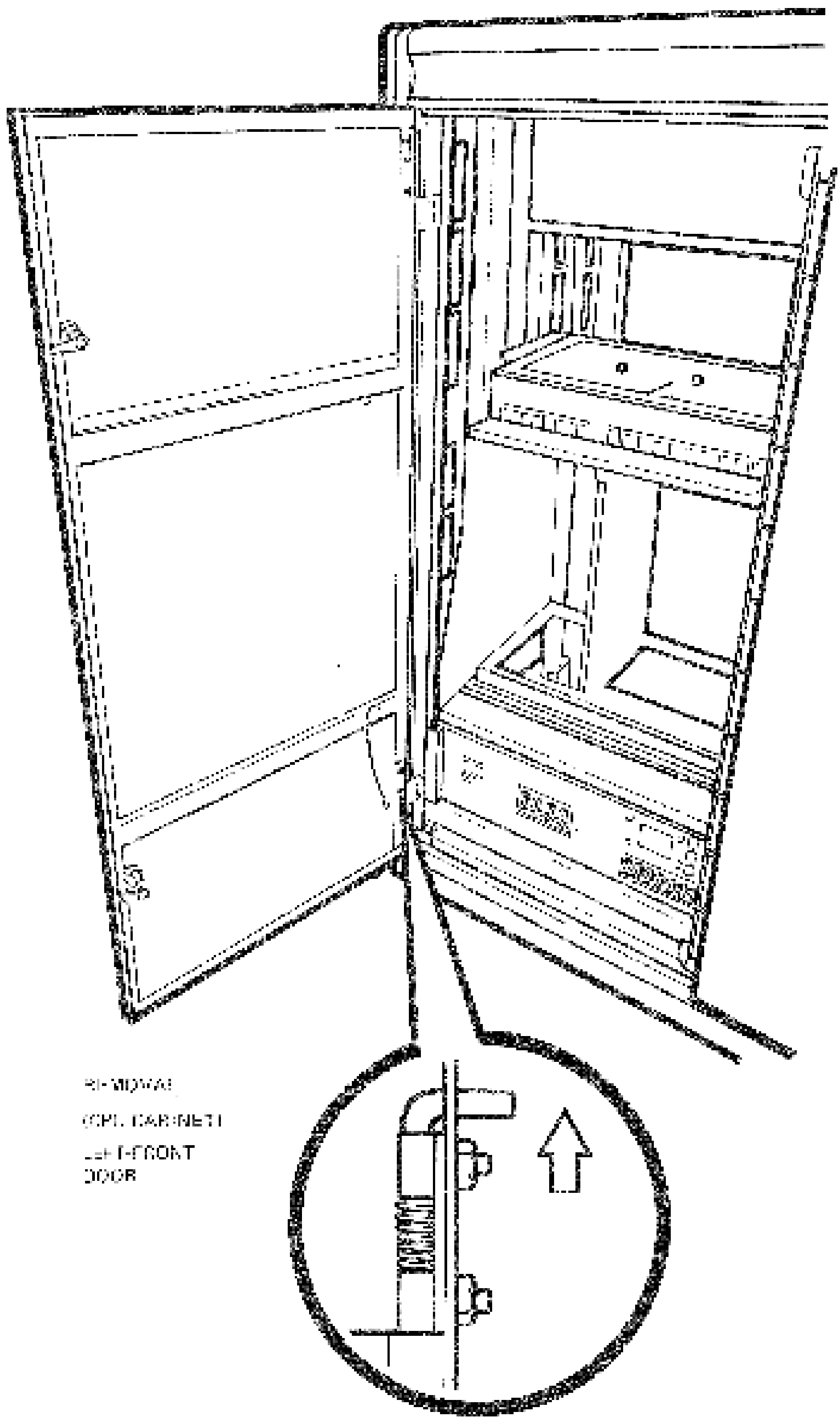


Figure 2-19 CPU and Front-End Cabinet Mating (Sheet 1 of 9)

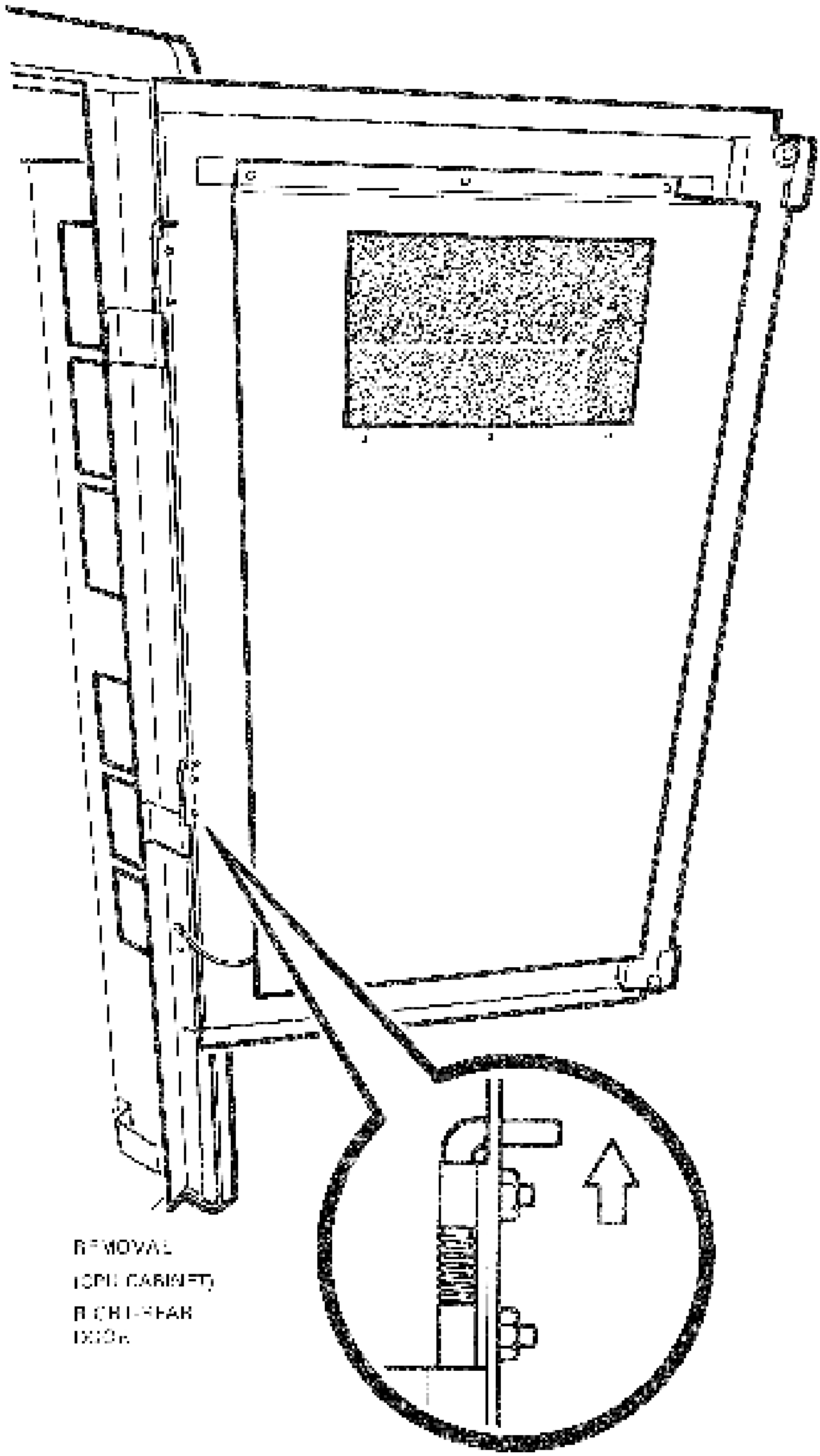


Figure 2-19 CPU and Front-End Cabinet Mating (Sheet 2 of 9)

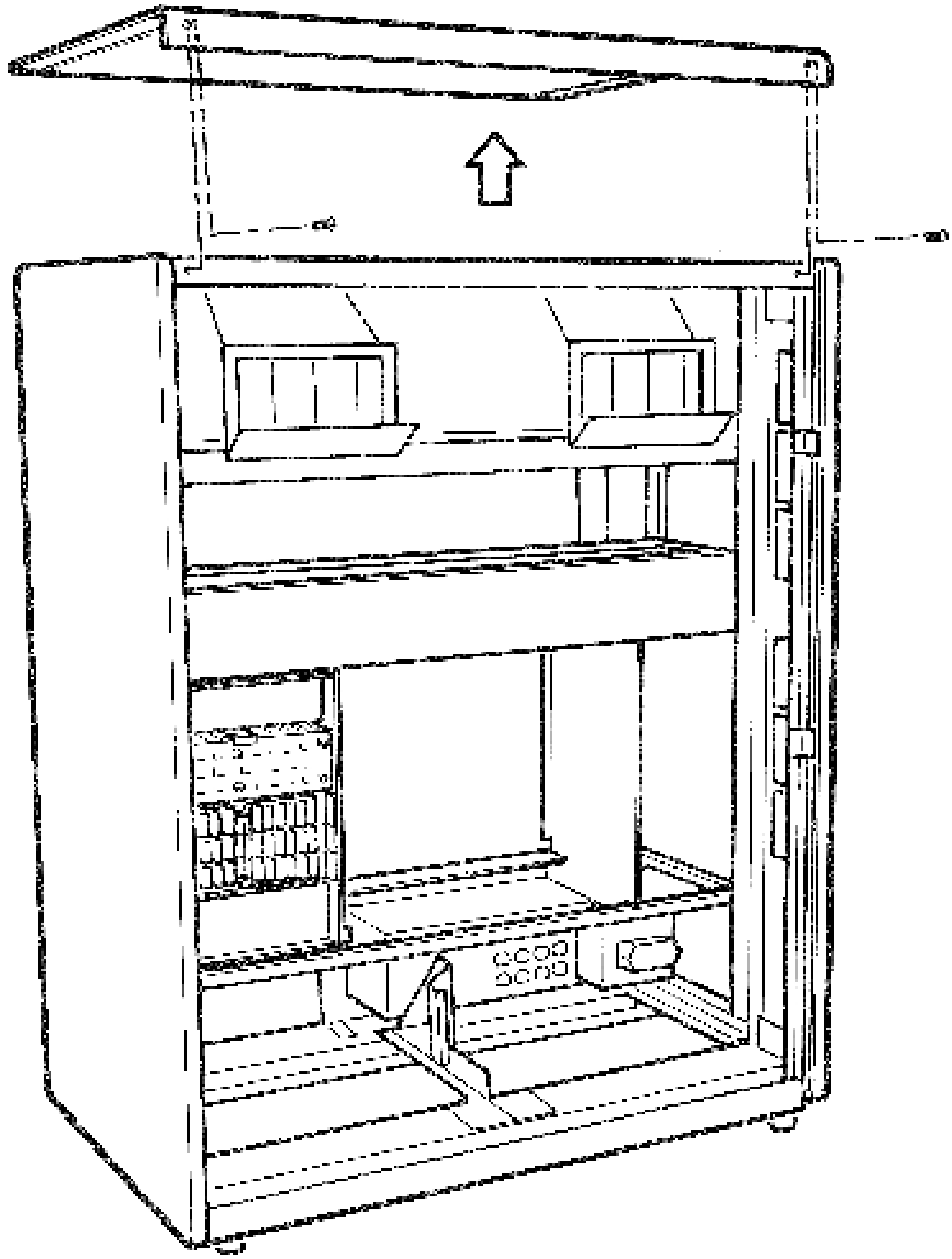


Figure 2-19 CPU and Front-End Cabinet Mating (Sheet 3 of 9)

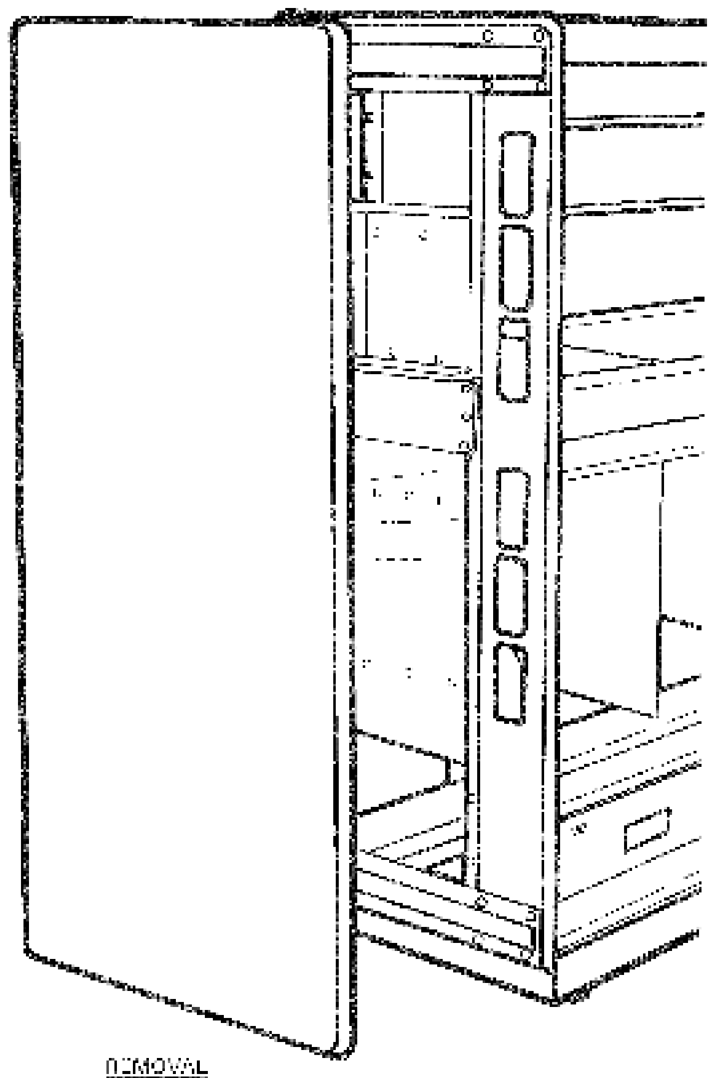


Figure 2-19 CPU and Front-End Cabinet Mating (Sheet 4 of 9)

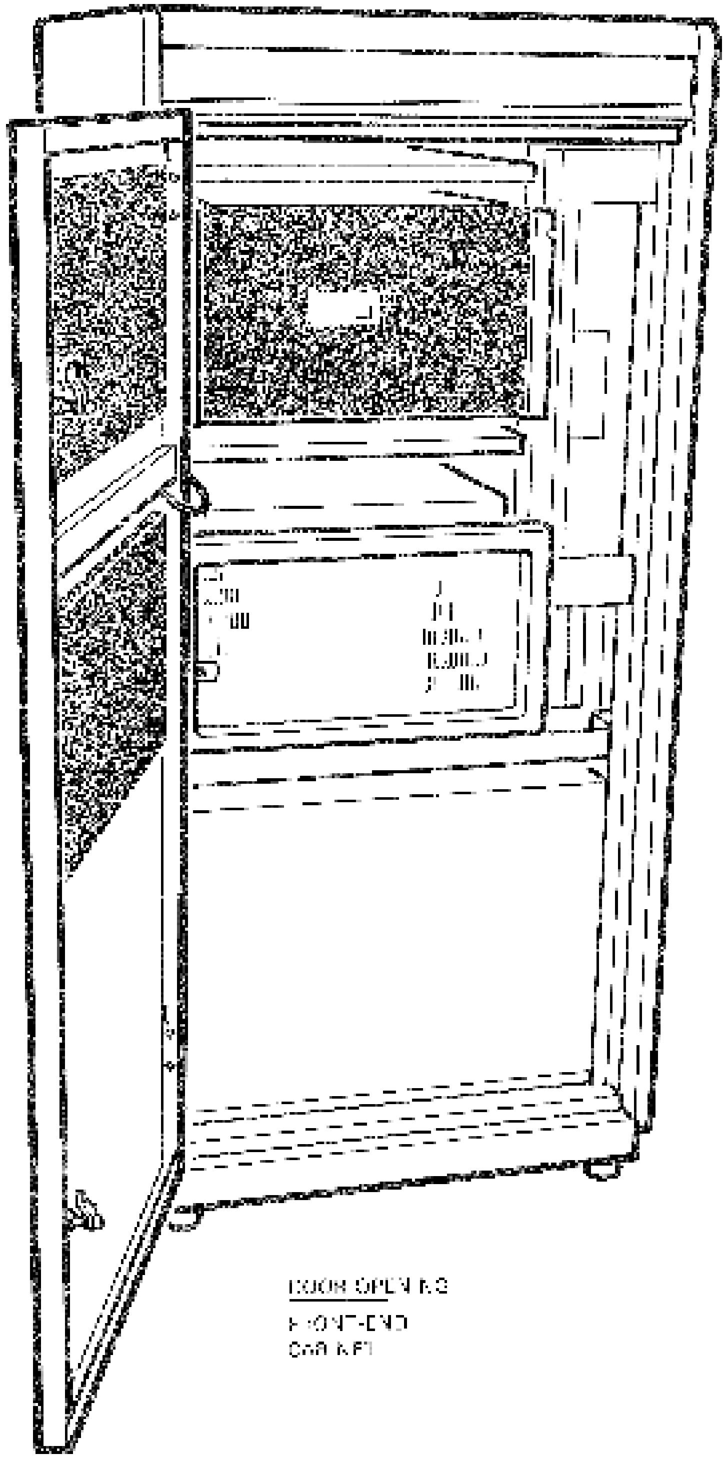


Figure 2-19 CPU and Front-End Cabinet Mating (Sheet 5 of 9)

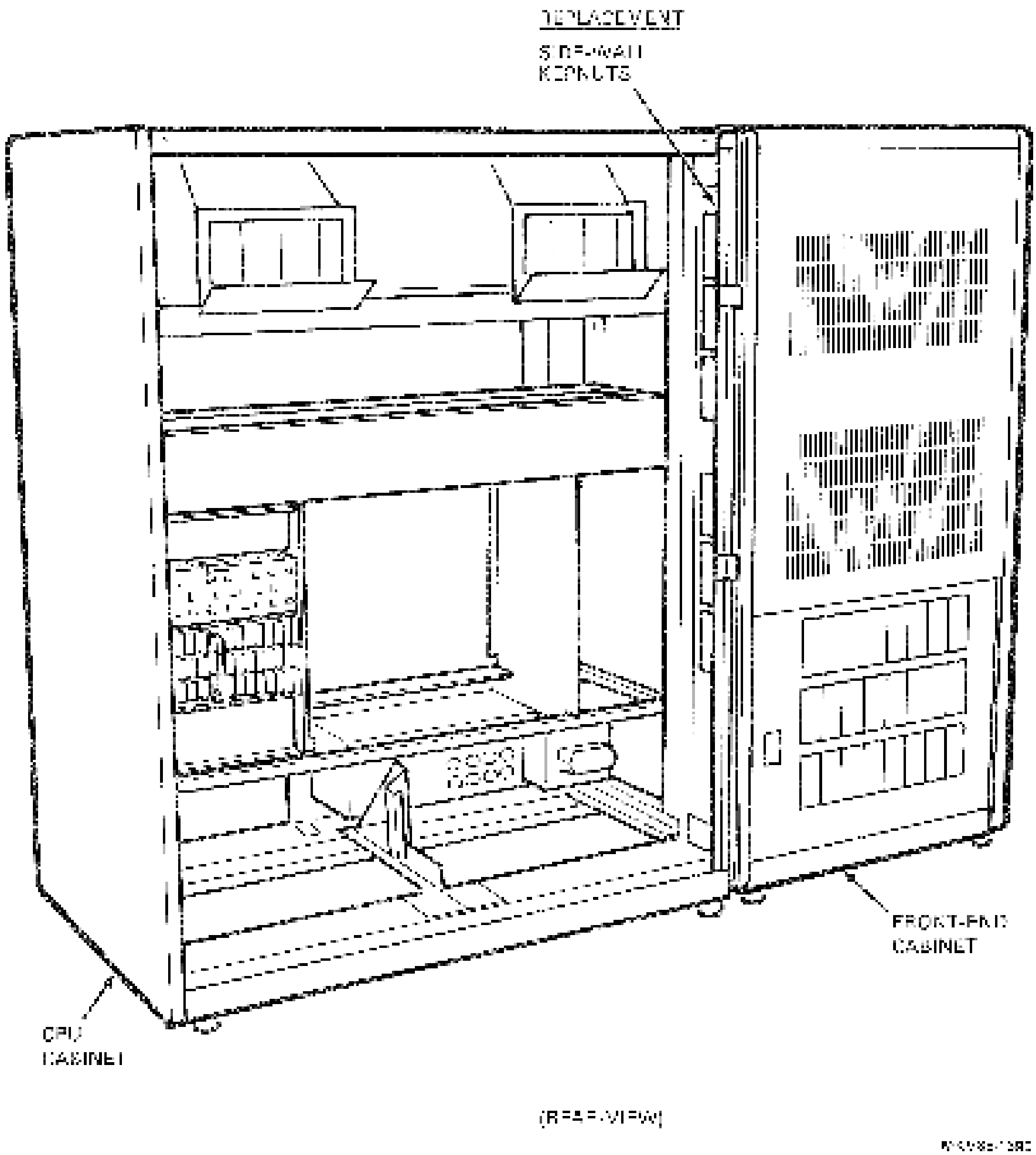


Figure 2-19 CPU and Front-End Cabinet Mating (Sheet 6 of 9)

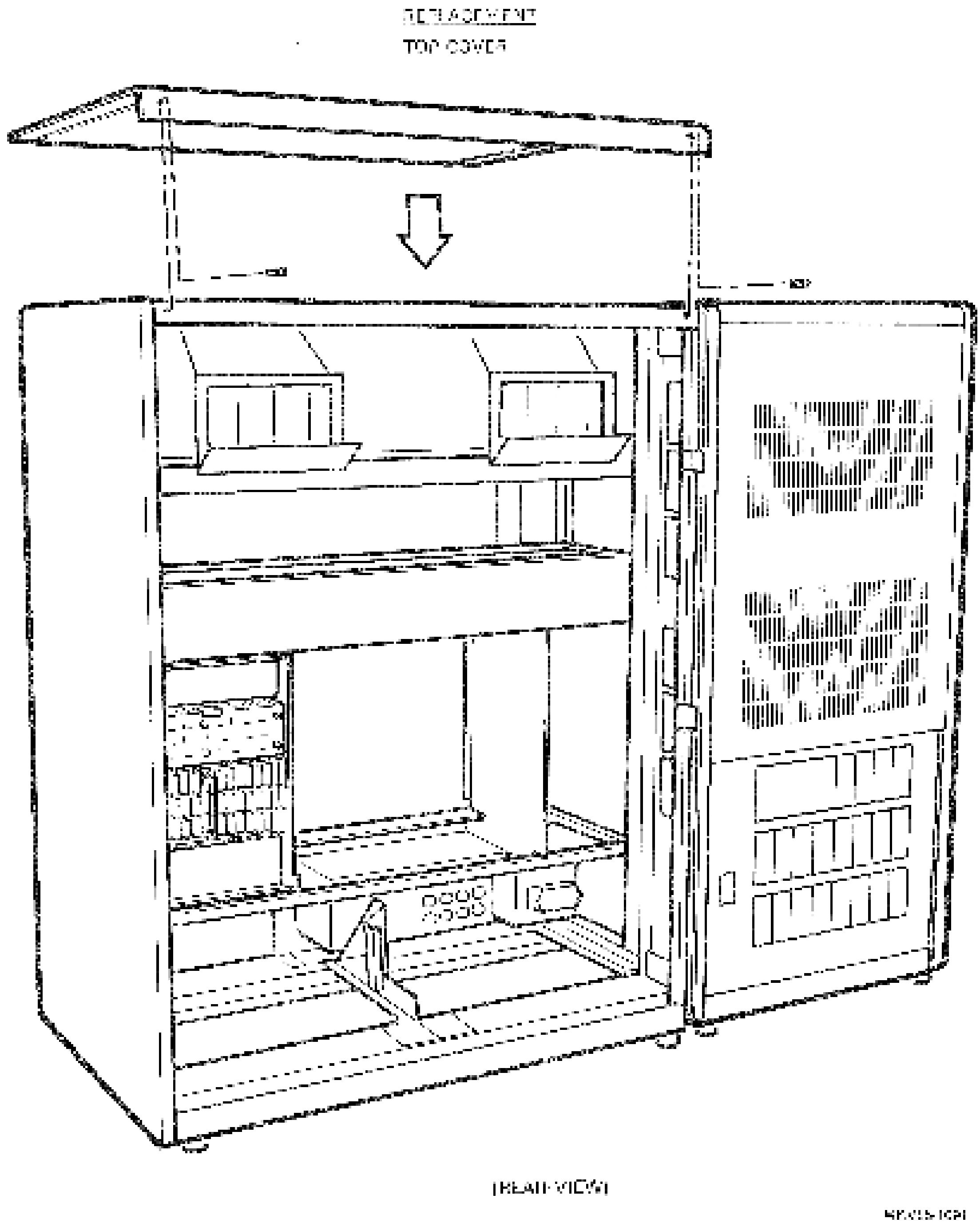


Figure 2-19 CPU and Front-End Cabinet Mating (Sheet 7 of 9)

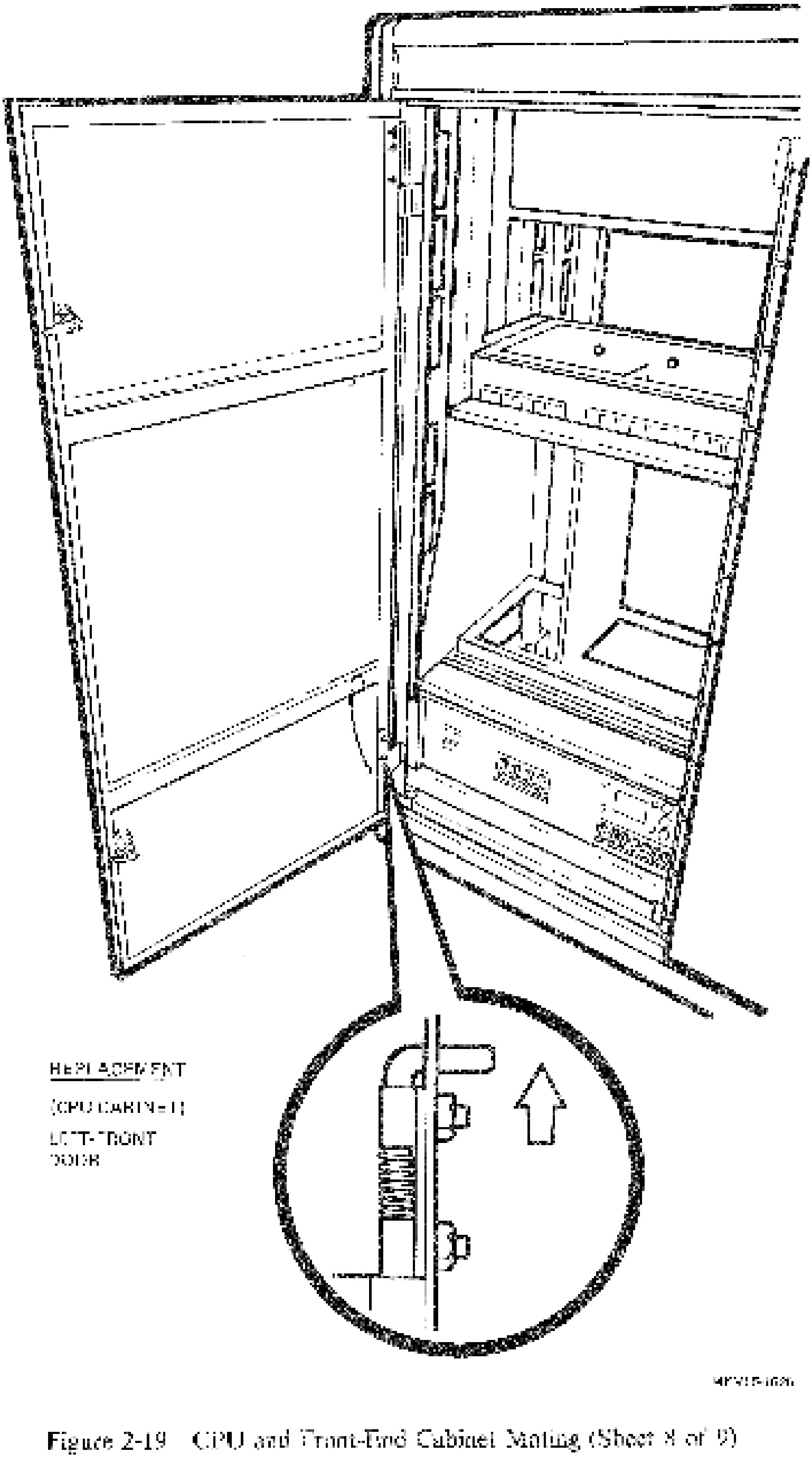


Figure 2-19 CPU and Front-End Cabinet Mating (Sheet 8 of 9)

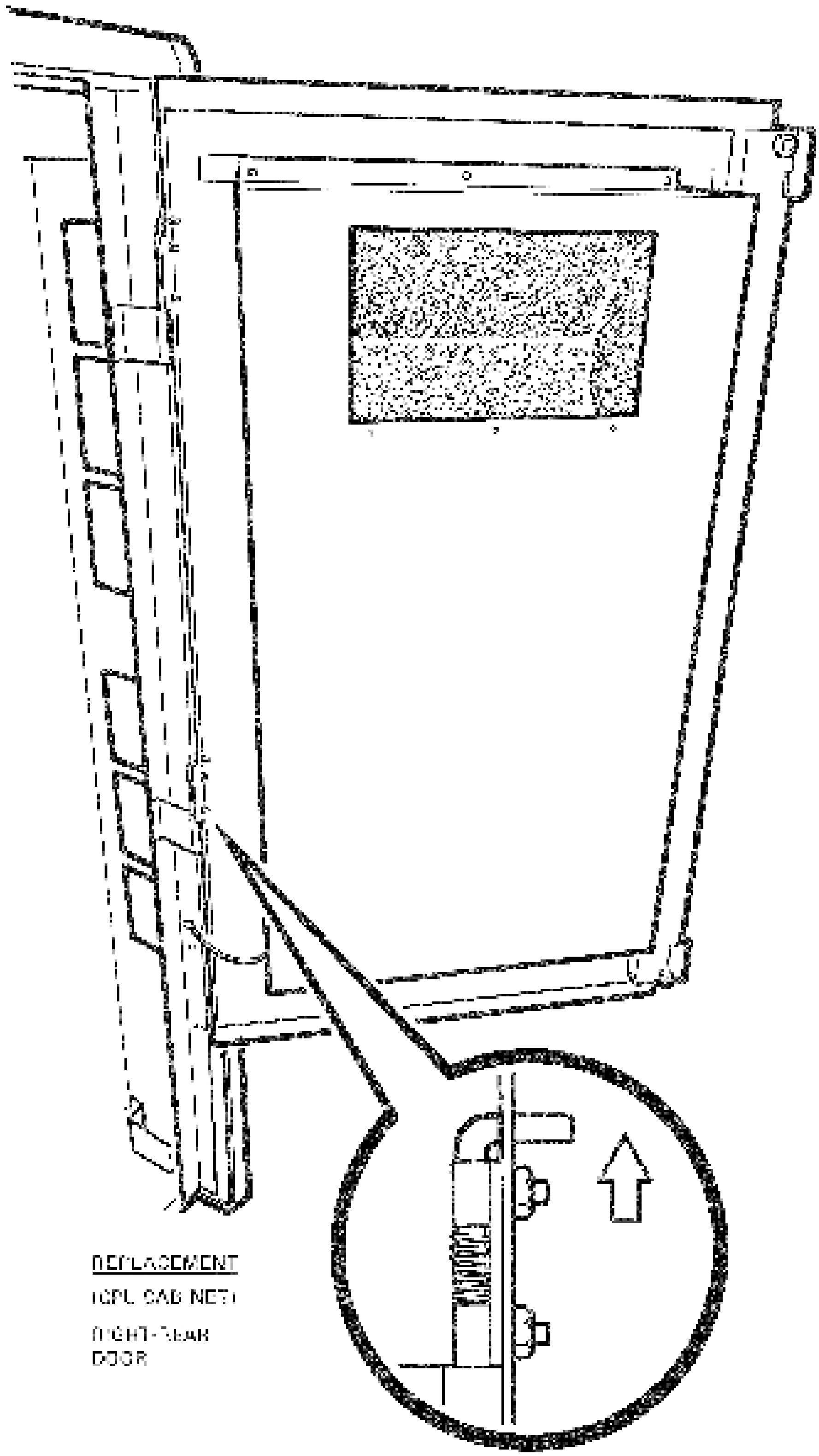


Figure 2-19 CPU and Front-End Cabinet Mating (Sheet 9 of 9)

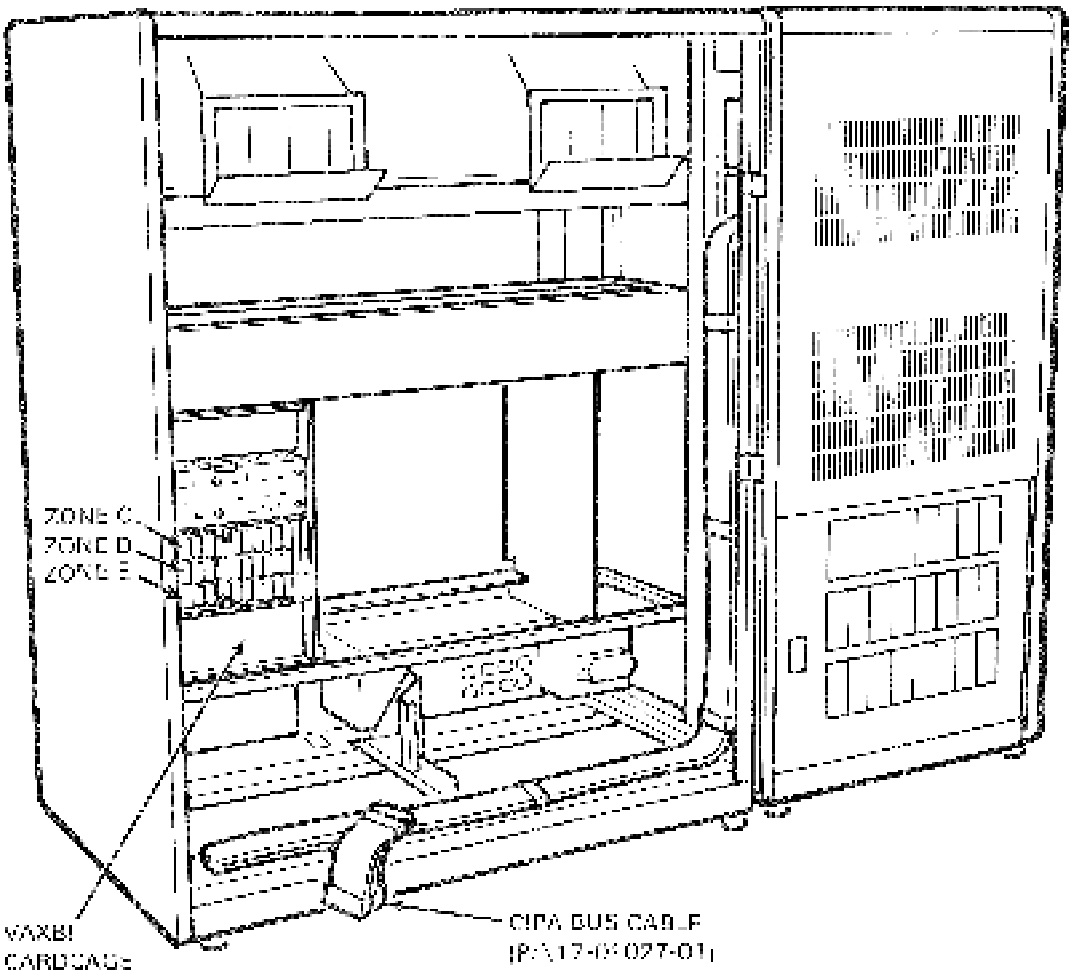


Figure 2-20 CIPA Bus Cable Routing

2.6 MECHANICAL INSTALLATION (Add-On Level)

An existing VAX 8000-Series system can easily be upgraded and configured for operation in a cluster environment. The upgrade process requires that the host system cabinetry contain sufficient space to accommodate a 10.50 inch mounting box.

There are two styles of CIPA mounting boxes. The style can be distinguished by the type of cable assembly, retractor or guidance, which is affixed at the rear of the mounting box (see Figure 2-21). A cable retractor assembly is employed when configuring to either VAX 8200 or VAX 8300 systems, and a cable guidance assembly is employed when configuring to either VAX 8500/8550 or VAX 8700/8800 systems.

Procedures for installing either mounting box are detailed in Sections 2.6.1 and 2.6.2.

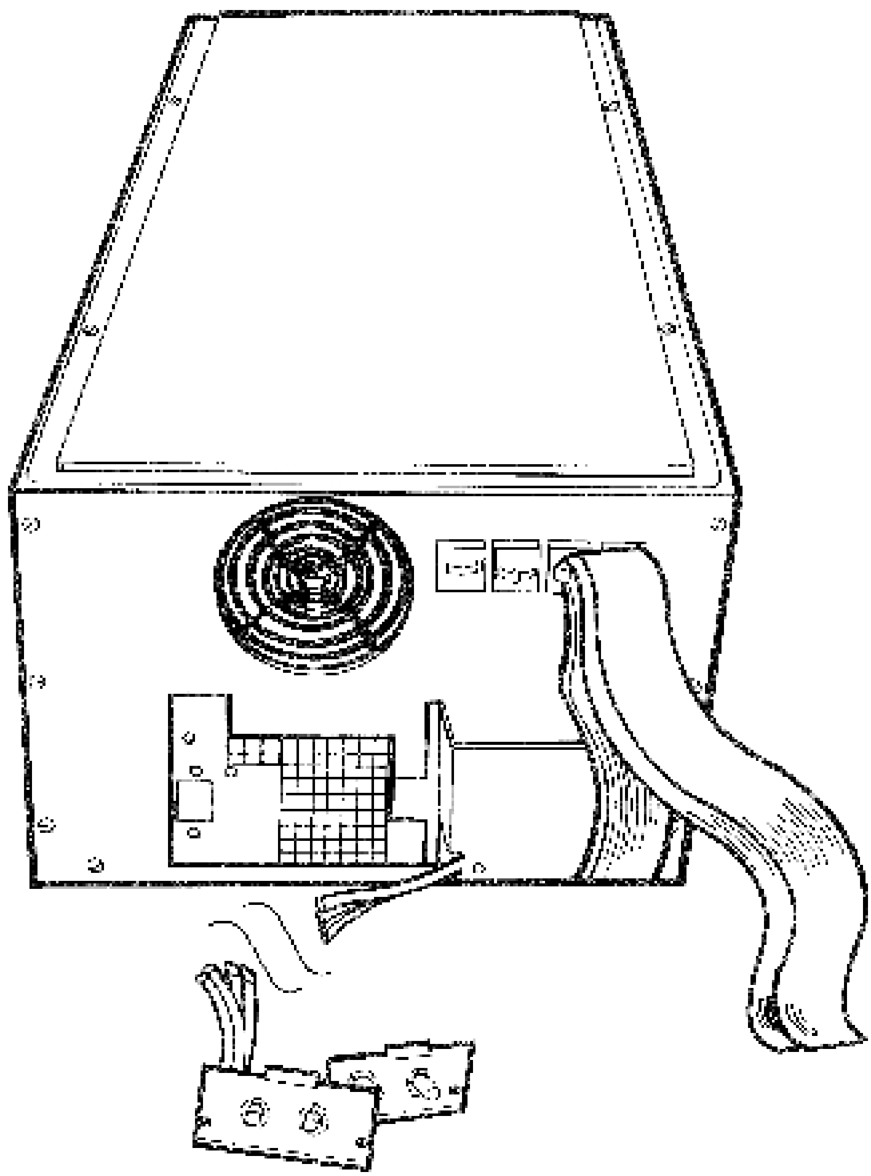


Figure 2-21 CIPA Mounting Box Differences (Sheet 1 of 2)

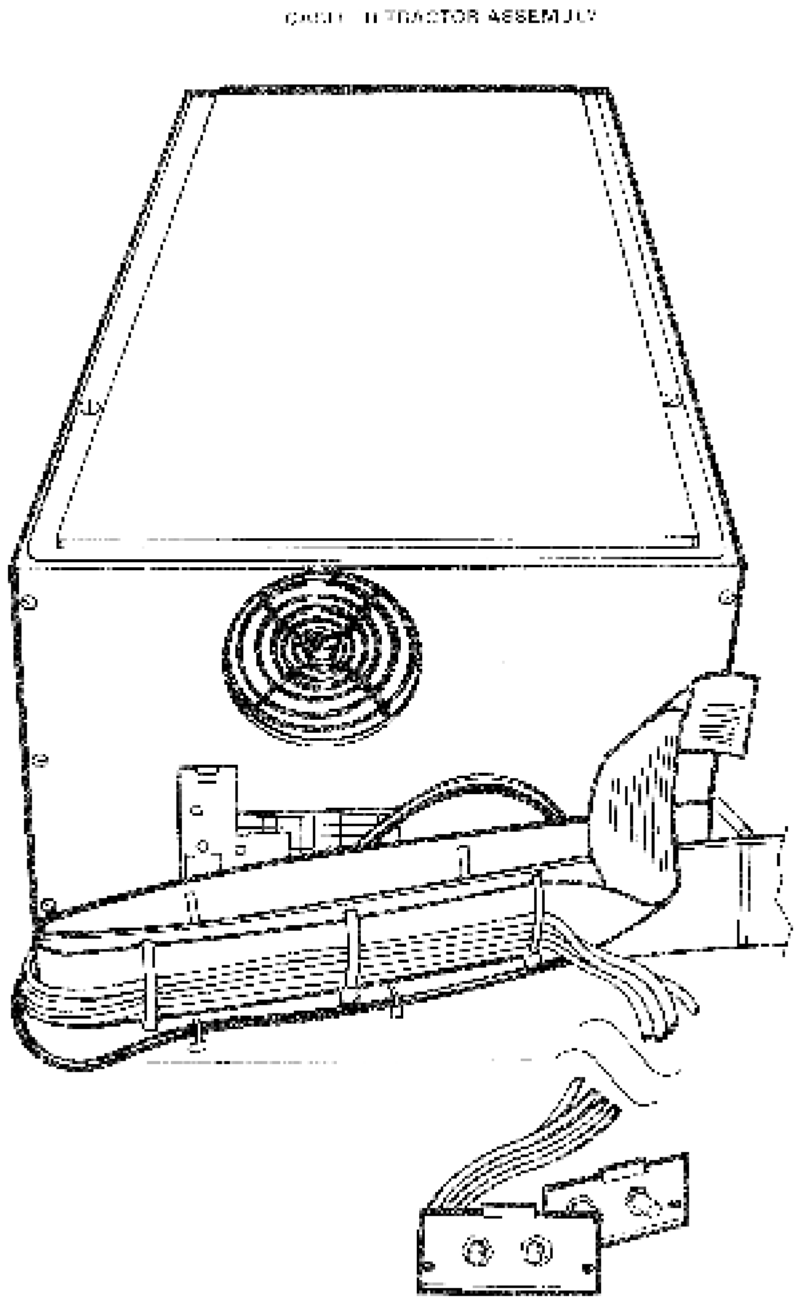


Figure 2-21 CIPA Mounting Box Differences (Sheet 2 of 2)

2.6.1 VAX 8200/8300 Systems

Procedure:

1. Using a 5/32 inch Allen wrench, remove front and rear doors of the CIPA cabinet.

2. Refer to Figure 2-22 while performing the following:

   a. Attach the chassis slide-track containing a ground strap to the front-right vertical rail (front to rear) by inserting a 10-32 screw into the bar nut and tightening.

   b. Attach the free end of the ground wire to the vertical rail with a 10-32 screw and 10-32 nut and tighten.

   c. Attach the other chassis slide-track to the front-left vertical rail (front to rear) by inserting a 10-32 screw into the bar nut and tightening.

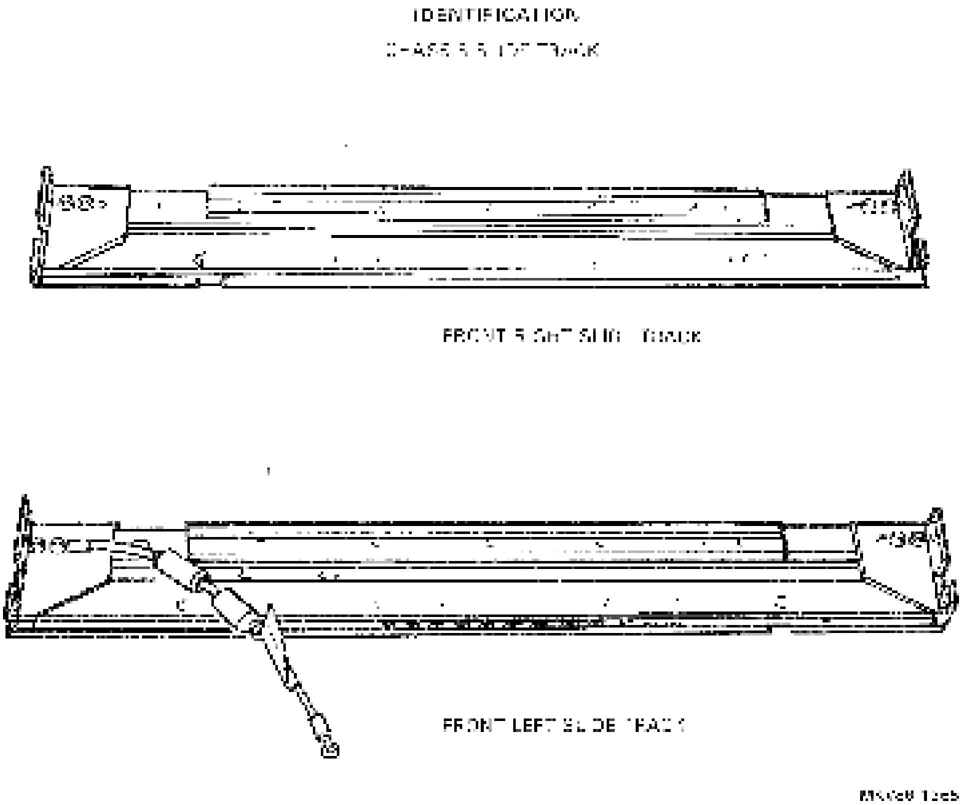


Figure 2-22 Mounting Box Installation Within an H9642 Cabinet (Sheet 1 of 3)

d. Extend each chassis slide-track outward from the cabinet by exerting pressure on the two release mechanisms.

e. Lift the mounting box parallel with the slide-tracks. Insert the 6-32 screws into the slide-tracks and tighten.

CAUTION

Lifting and handing the mounting box requires the assistance of at least two individuals or a mechanical lift device.

3. From the rear-inside of the cabinet, locate the free end of the flat ribbon CIPA bus cable.
4. Route the CIPA bus cable through the RFI shield panel knockout hole and waveguide and then through the RFI shield panel knockout hole of the CPU cabinet.
5. Route the CIPA bus cable up through the interior of the processor cabinet to the bottom of the BA32 mounting box and then outward from the cabinet.
6. Remove the cable strain relief clamp on the BA32 mounting box.
7. Route the CIPA bus cable through the I/O cable slot located in the lower rear of the BA32 mounting box.
8. Place the power controller's local/remote switch in its remote position.
9. Connect the power cord of the mounting box into the power controller's switched receptacle (J8 for unit #1 or J9 for unit #2).
10. Install the coaxial cable set assembly to either the bulkhead I/O panel or to the star coupler.
11. Route, but do NOT connect at this time, the BNCIA-xx coaxial (CI bus) cables. Refer to the *Star Coupler User's Guide*.
12. Perform the electrical installation detailed in Sections 2.7.1 through 2.7.3 of this manual.

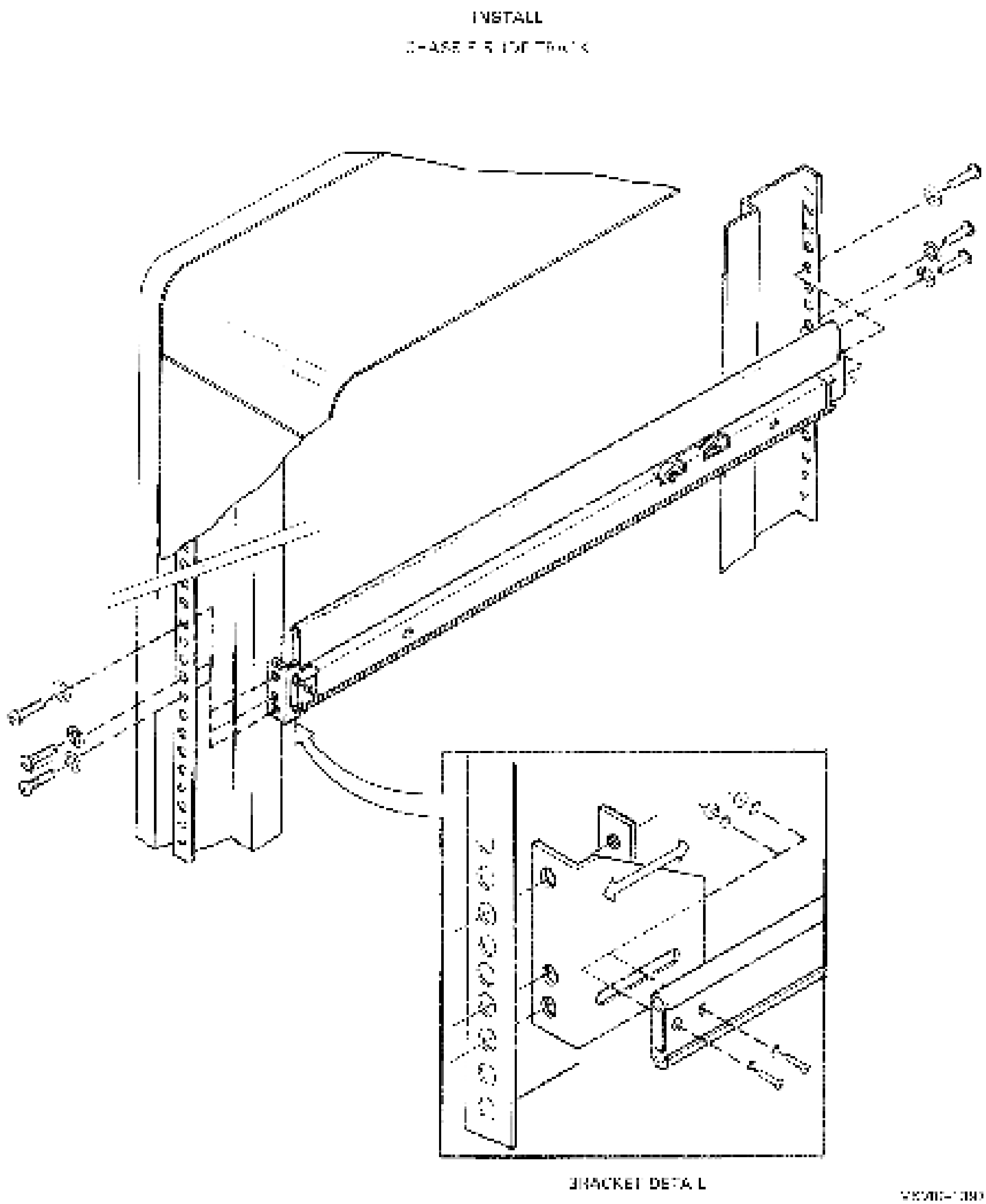


Figure 2-22 Mounting Box Installation Within an H9642 Cabinet (Sheet 3 of 3)

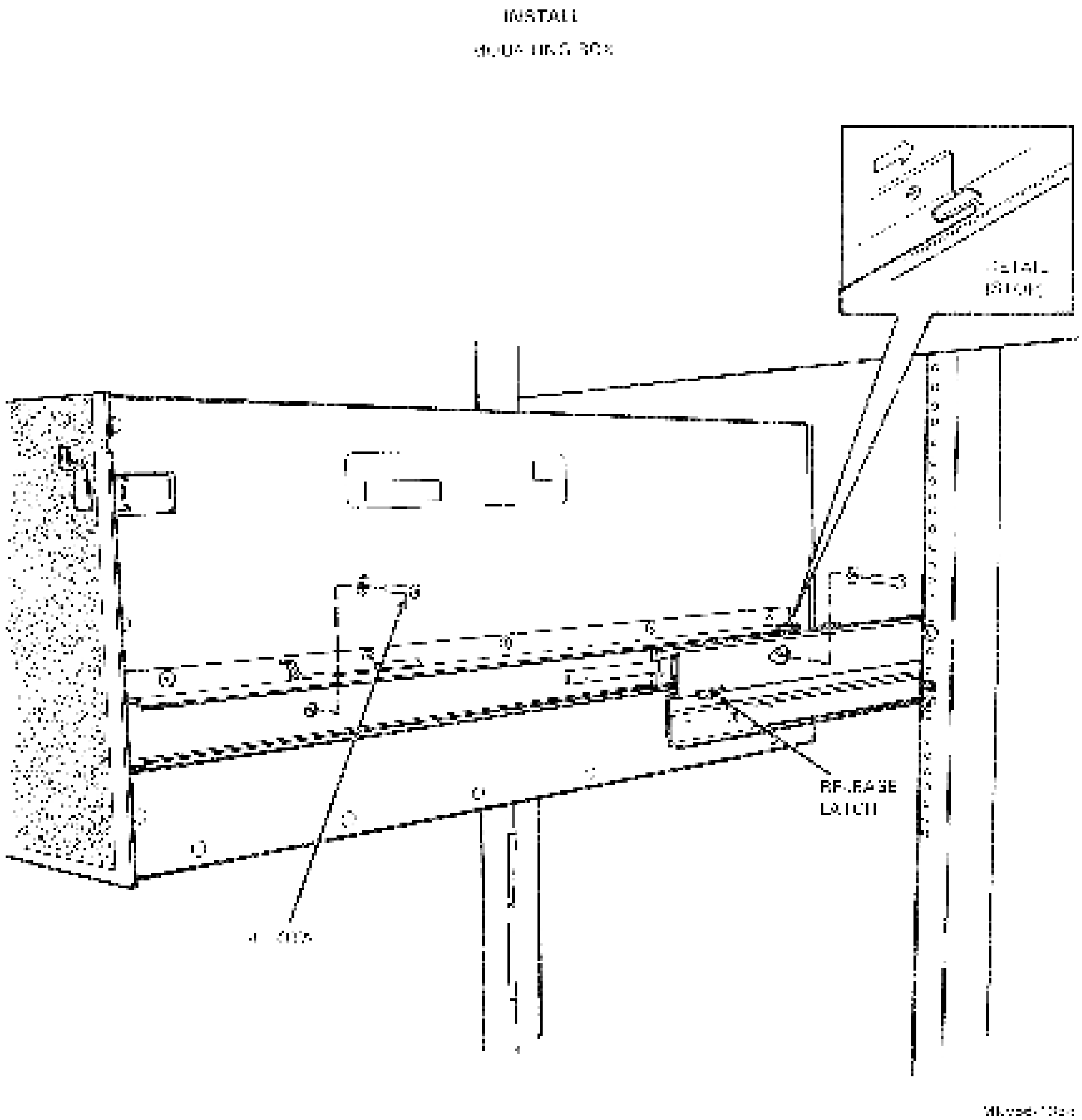


Figure 2-22   Mounting Box Installation Within an H9642 Cabinet (Sheet 3 of 3)

Procedure:

1. Remove front and rear doors of the front-end cabinet or expansion cabinet by removing the single hex screw securing the door's ground strap to the cabinet frame, pulling the release mechanism on each door hinge, and then lifting the door off its hinges.

2. Refer to Figure 2-23 while performing the following:

   a. Attach the chassis slide track containing a ground strap to the front-right vertical rail (front to rear) by inserting a 1/4-20 screw into the 1/4-20 isolation mount and tightening.

   b. Attach the free end of the ground wire to the vertical rail with a 10-32 screw and 10-32 nut, and tighten.

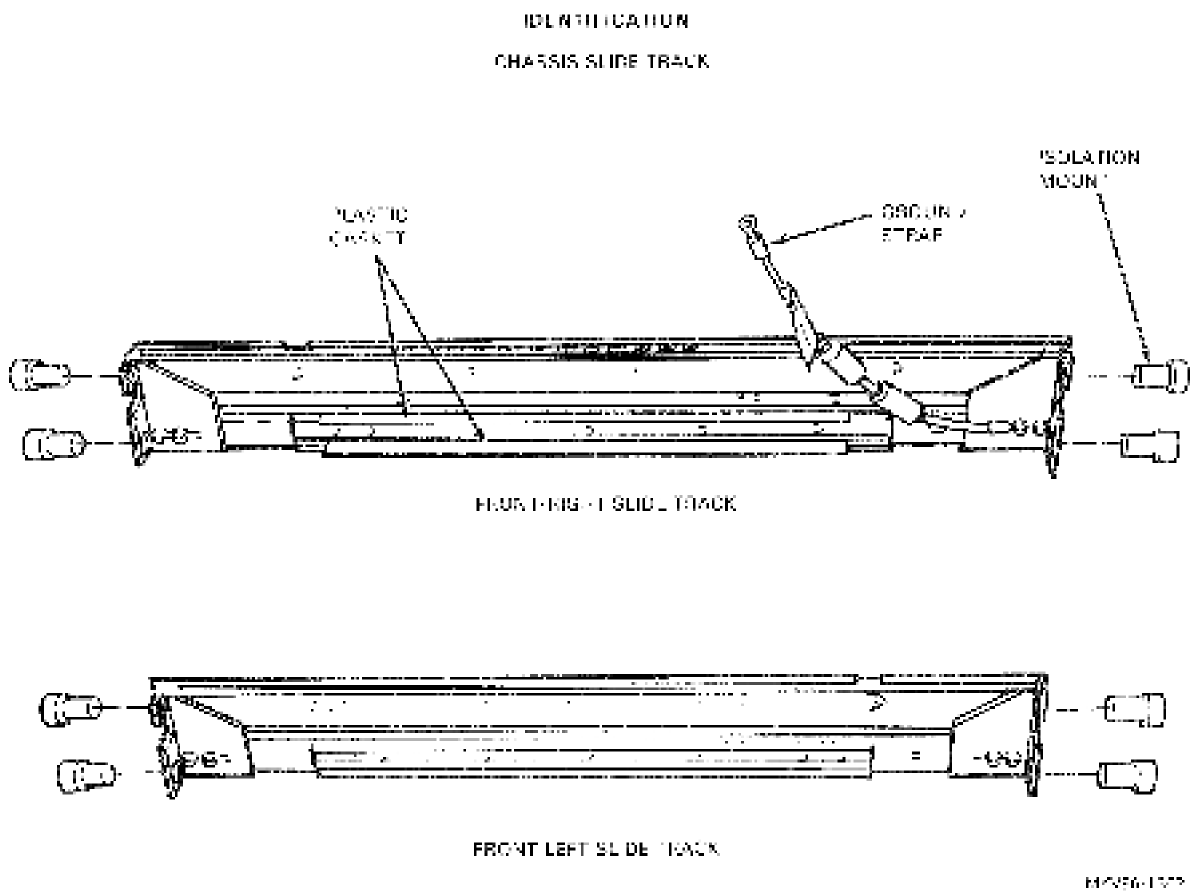


Figure 2-23 Mounting Box Installation Within an H9652 Cabinet (Sheet 1 of 3)

c. Attach the other chassis slide-track onto the front-left vertical rail (front to rear) by inserting a 1/4-20 screw into the 1/4-20 isolation mount and tightening it.

NOTE
To comply with Section 2.2.3 of this manual, visually inspect for the presence of:

1) A U-shaped gasket on both slide stiffeners of each chassis slide-track.

2) A two inch U-shaped gasket at the front right vertical rail of the cabinet and equally centered at the mounting box latching mechanism.

3) Two small rubber bumper feet affixed to the cabinet's front vertical rails in the vicinity of the chassis slide-track.

d. Extend each chassis slide-track outward from the cabinet by exerting pressure on the two release mechanisms.

e. Lift the mounting box parallel with slide-tracks, insert the 6-32 screws into the slide-tracks, and tighten.

CAUTION
Lifting and handling the mounting box requires the assistance of at least two individuals or a mechanical lift device.

3. Route the CIPA bus cables from the cabinet to the VAXBI cardcage located in the processor cabinet.

NOTE
Refer to the appropriate *Installation Guide* (EK-8500I-IN or EK-8800I-IN) for additional intra-cabinet cabling.

4. Connect the power cord of the mounting box into the power controller's switched receptacle.

5. Install the coaxial cable set assembly to the bulkhead I/O panel.

6. Route, but do NOT connect at this time, the BNCIA-xx coaxial (CI bus) cables. Refer to the *Star Coupler User's Guide.*

7. Perform the electrical installation detailed in Sections 2.7.1 through 2.7.3 of this manual.

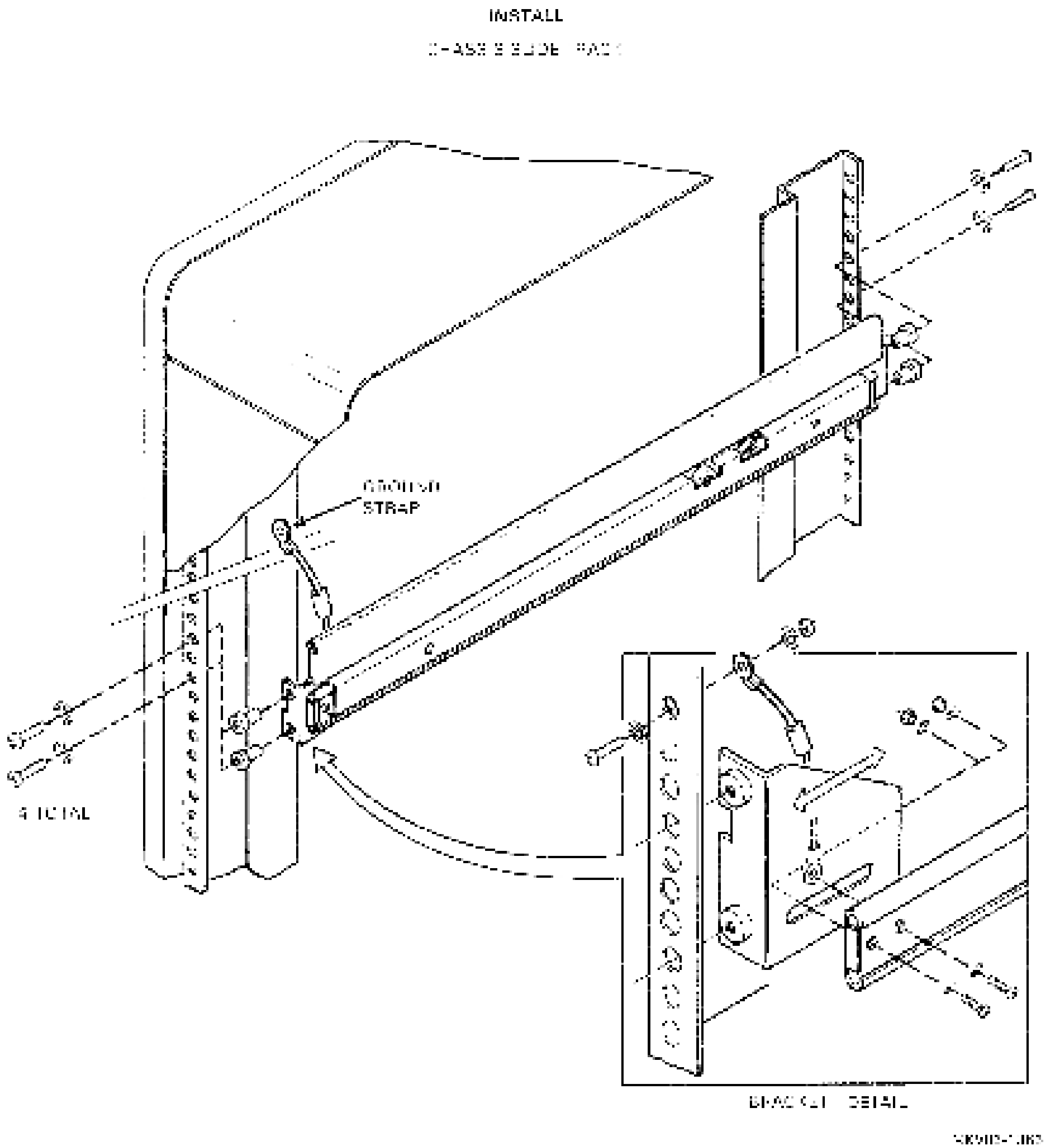


Figure 2-23 Mounting Box Installation Within an H9652 Cabinet (Sheet 2 of 3)

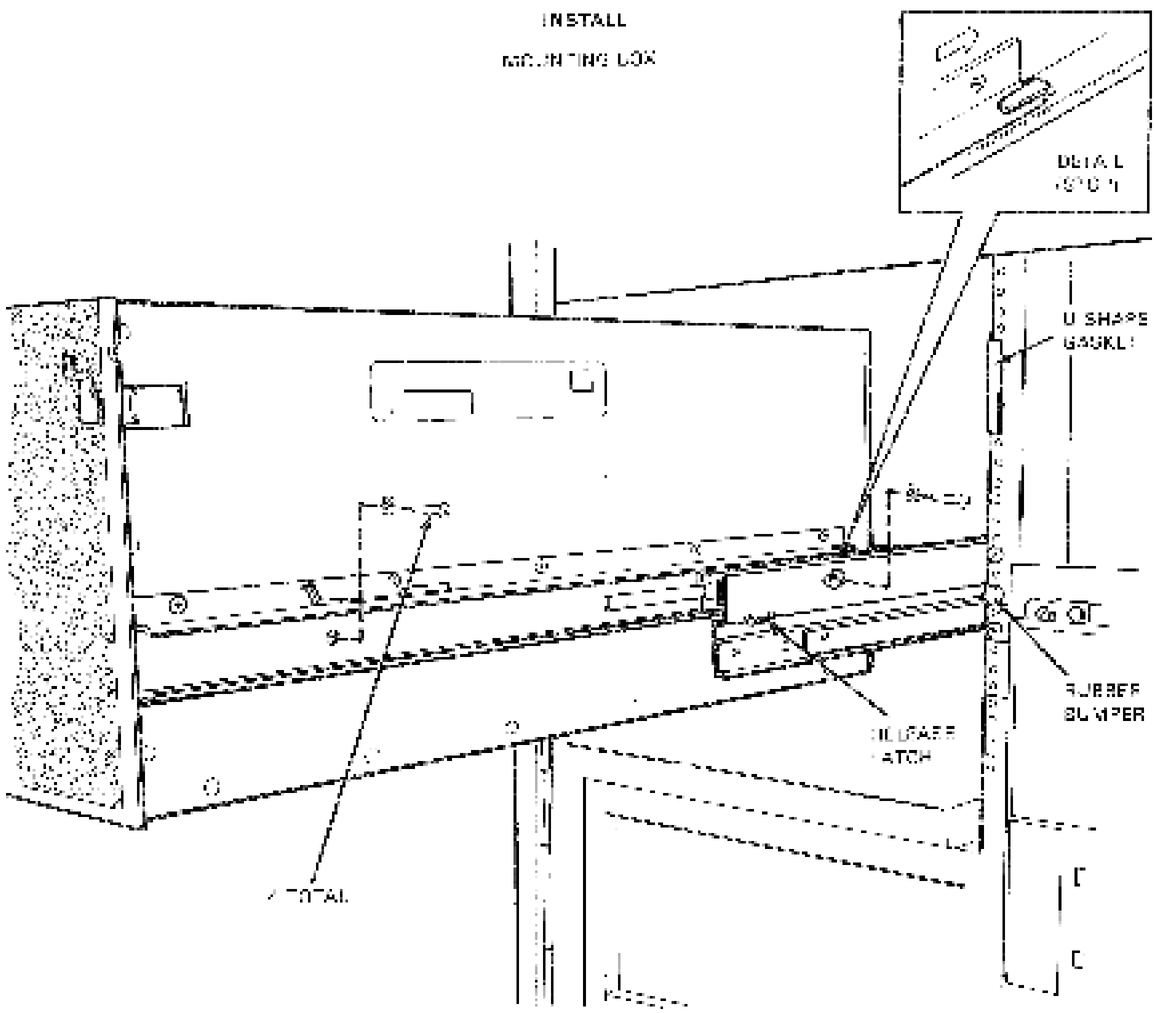


Figure 2-23 Mounting Box Installation Within an H9652 Cabinet (Sheet 3 of 3)

2.7 ELECTRICAL INSTALLATION AND CONFIGURATION

2.7.1 T1017 and T1018 Module Installation

CAUTION
Use a static discharge system (Velostat™ Kit P/N 29-11762-00) when handling the T1017 and T1018 modules to prevent damage due to electrostatic discharge.

NOTE
Proceed directly to Step 3 if the VAXBI cardcage already contains the T1017 and T1018 modules and the BCI cables.

® Velostat™ is a trade mark of the Minnesota Mining and Manufacturing Co.

Procedure:

1. Carefully insert the T1017 and T1018 modules into any two unoccupied but adjacent module slots within the VAXBI cardcage (see Figure 2-24).

CAUTION

When installing the modules, the T1017 must be positioned to the left of the T1018 module.

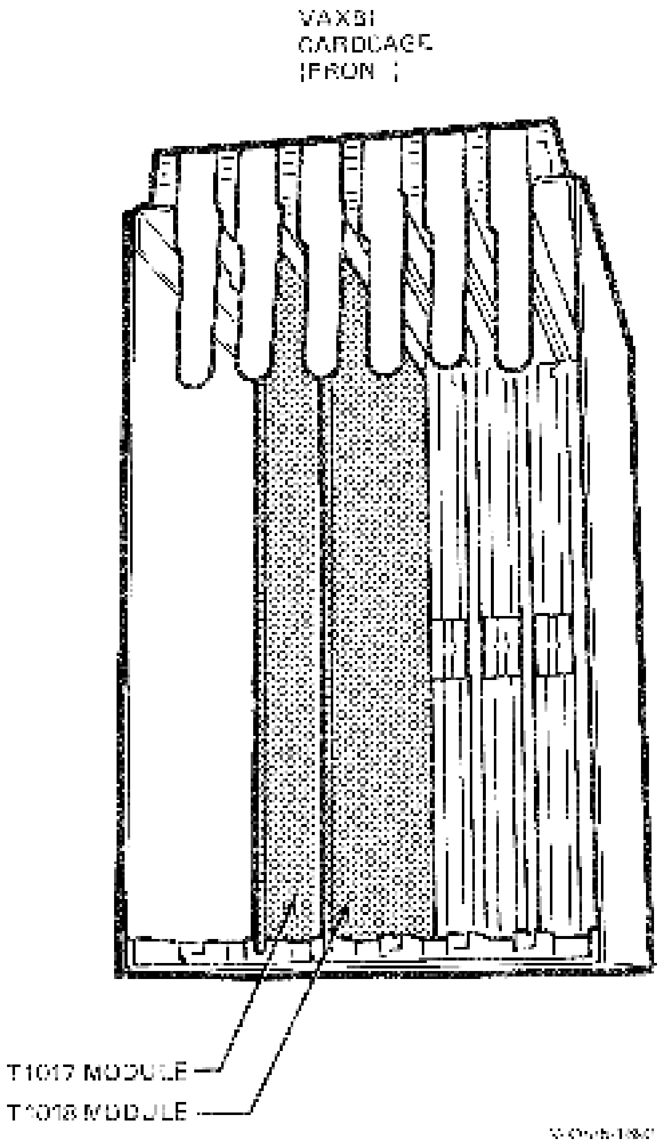


Figure 2-24 VAXBI Backplane – Module Installation

2. Refer to Figure 2-25 while carefully connecting the BCI cables to the VAXBI cardcage connectors, as follows:

   a. Attach a 3 inch cable (P/N 17-01029-02) at zone C between the innermost connectors of the T1017 and T1018 modules.

   b. Attach a 3 inch cable (P/N 17-01029-02) at zone D between the innermost connectors of the T1017 and T1018 modules.

   c. Attach a 3.75 inch cable (P/N 17-01029-01) at zone C between the outermost connectors of the T1017 and T1018 modules.

   d. Attach a 3.75 inch cable (P/N 17-01029-01) at zone D between the outermost connectors of the T1017 and T1018 modules

3. Refer to Figure 2-26 while carefully connecting the CIPA bus cable to the VAXBI cardcage connectors, as follows:

CAUTION
Orientate the marker stripe on the cable assembly toward zone A of the VAXBI cardcage.

   a. Attach connector P6 of the CIPA bus cable (P/N 17-01027-01) to the side 2 connector of the T1017 module at zone E.

   b. Attach connector P5 of the CIPA bus cable (P/N 17-01027-01) to the side 1 connector of the T1017 module at zone E.

   c. Attach connector P4 of the CIPA bus cable (P/N 17-01027-01) to the side 2 connector of the T1018 module at zone E.

   d. Attach connector P3 of the CIPA bus cable (P/N 17-01027-01) to the side 1 connector of the T1018 module at zone E.

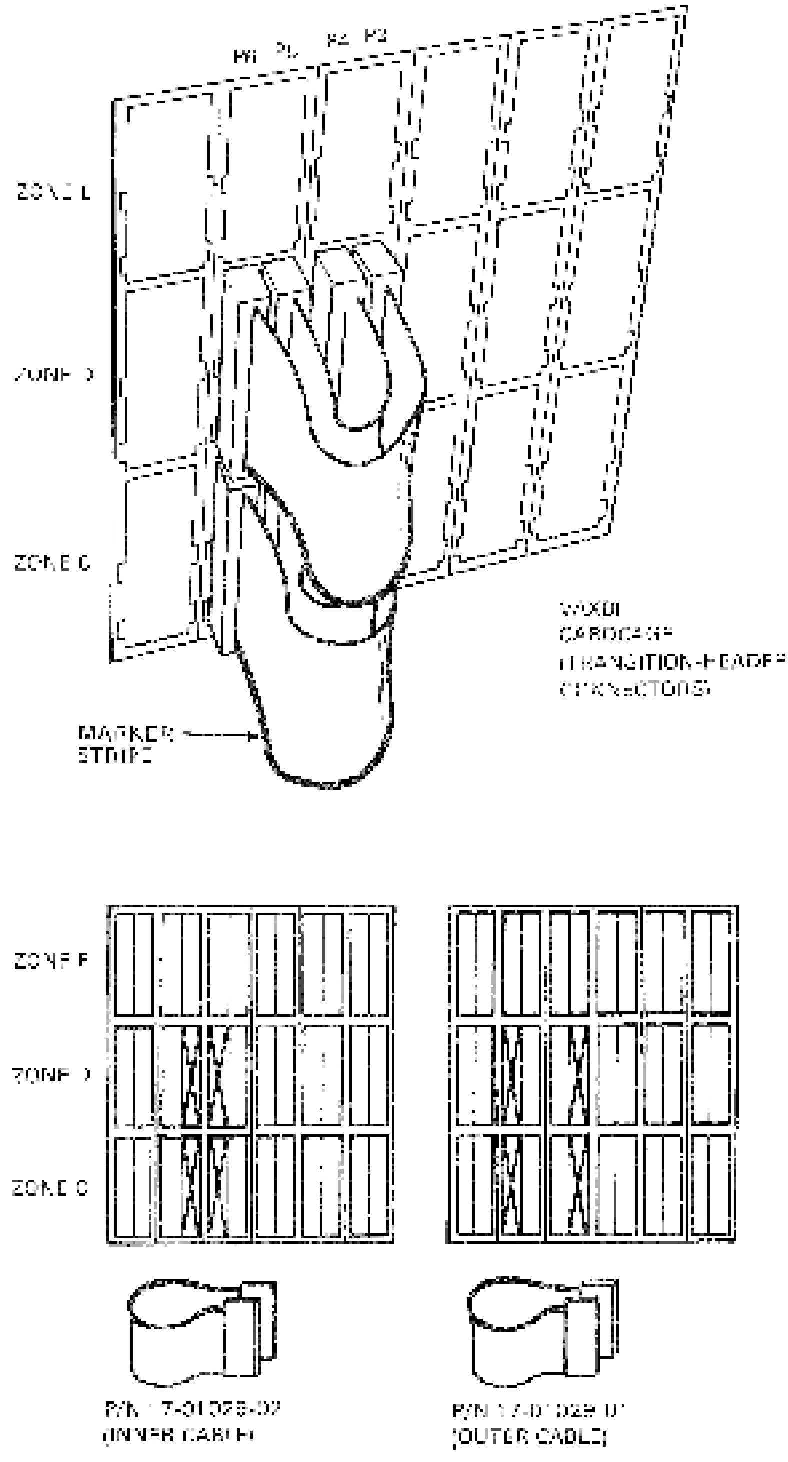
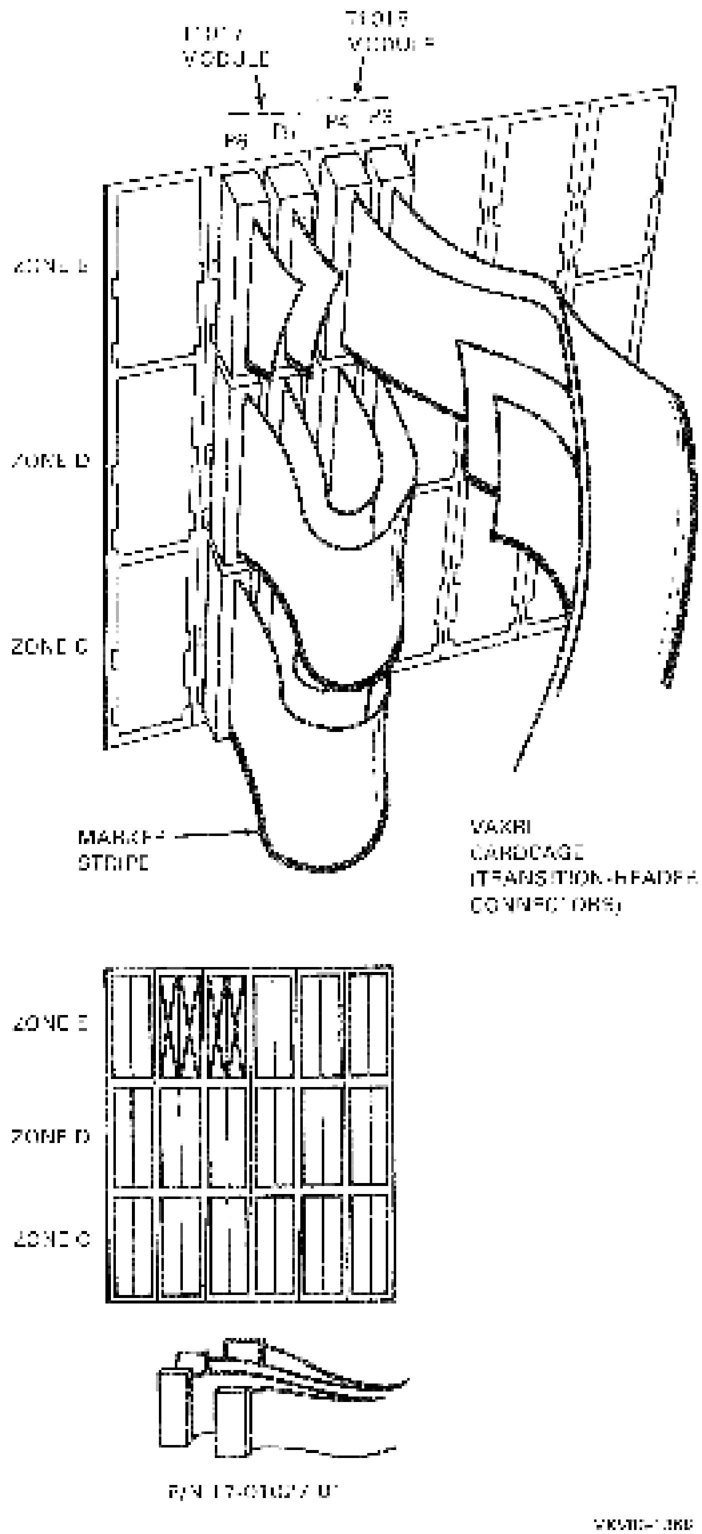
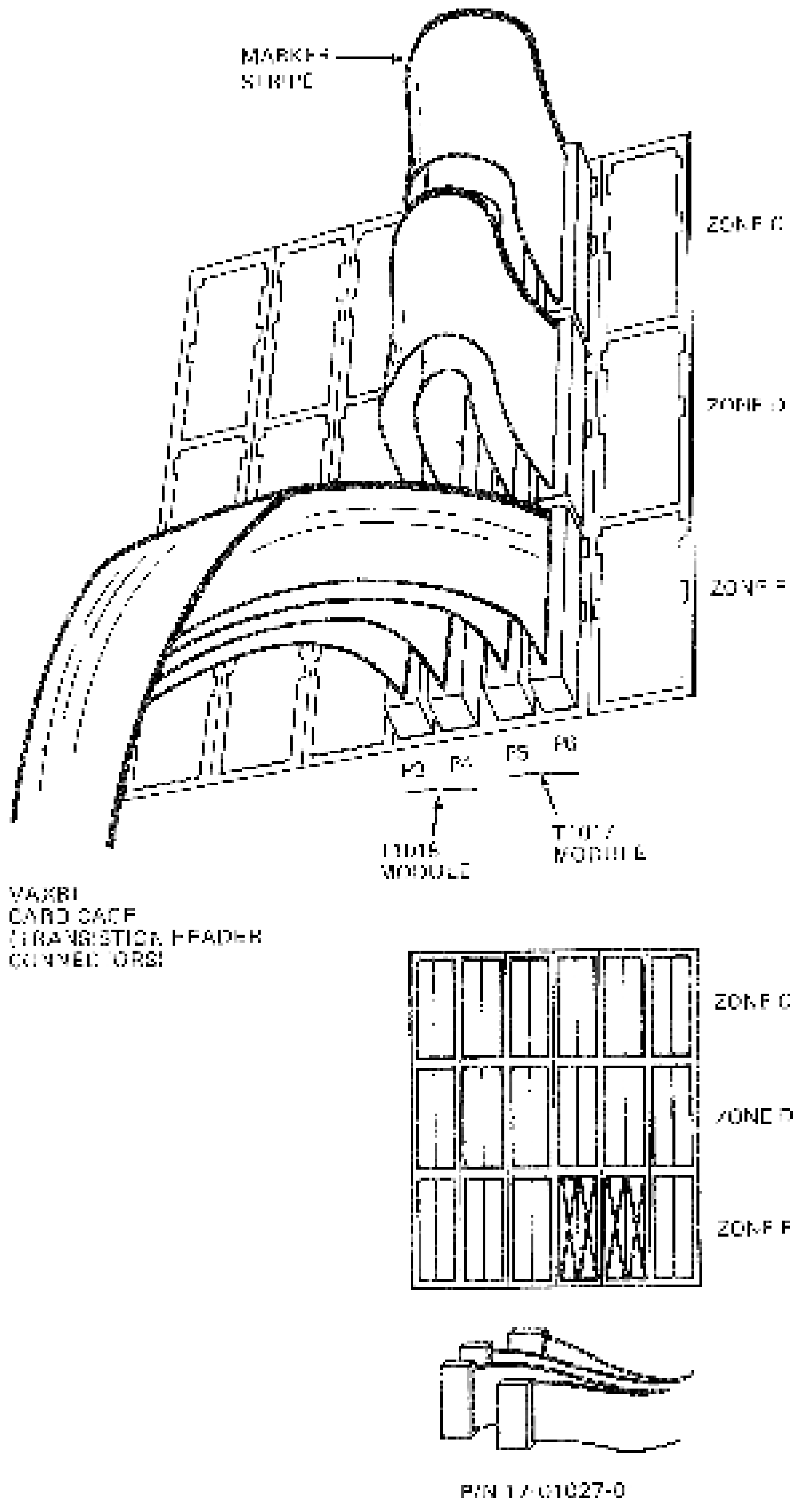


Figure 2-25 VAXBI Backplane -- BCI Cable Connections



a. VAX 8200 or VAX 8300 Systems

Figure 2-26 VAXBI Backplane – CIPA Bus Cable Connections (Sheet 1 of 2)



b. VAX 8500/8550/8700/8800 Systems

Figure 2-26 VAXBI Backplane - CIPA Bus Cable Connections (Sheet 2 of 2)

2.7.2 CIPA Backplane Jumpers Verification

Eight jumpers (W1-W8) are used on the CIPA backplane to control certain operating parameters on the data path (L0400) and link interface (L0100) modules. These jumpers are configured at the factory for normal operation, which is jumpers out.

Procedure:

1. Refer to Figure 2-27 while performing the following:

   a. Remove the shipping screws that secure the CIPA mounting box to the cabinet frame.

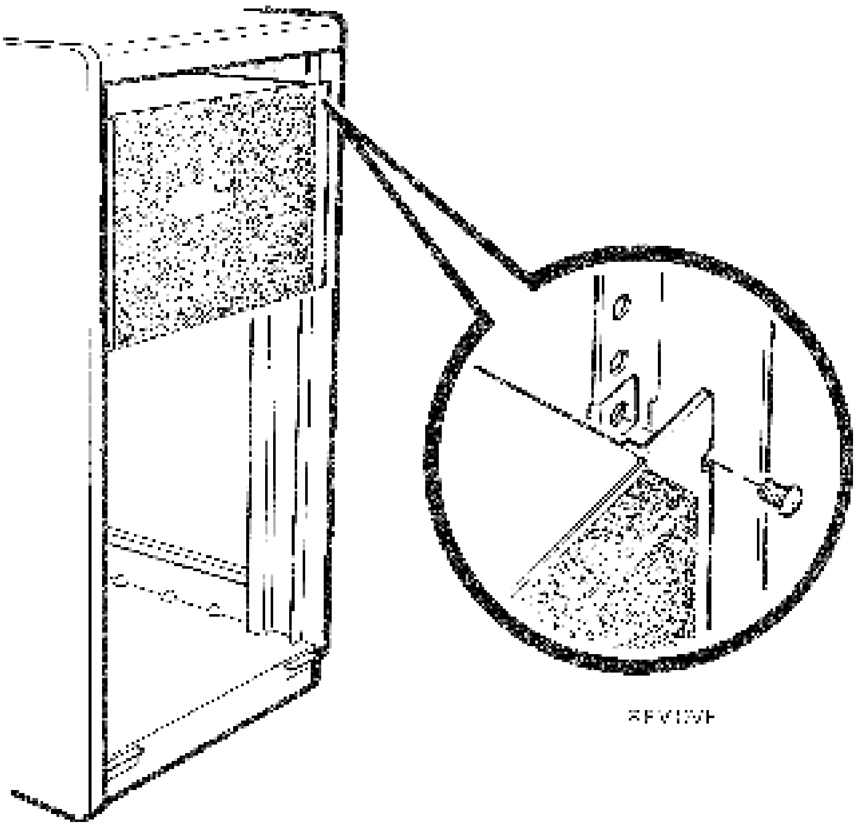


Figure 2-27 CIPA Mounting Box Access (Sheet 1 of 3)

b. Release the safety latch located on the front of the mounting box at the upper-right corner and slide the mounting box, on its rails, part way out of the cabinet.

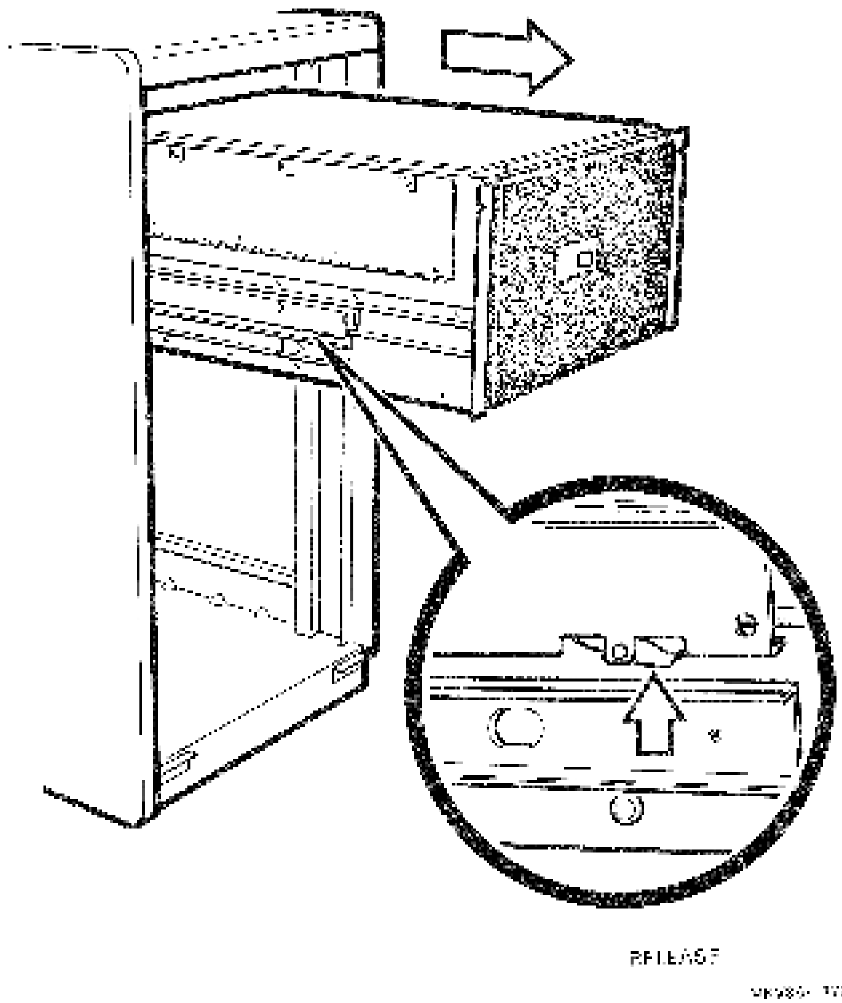


Figure 2-27 CIPA Mounting Box – Access (Sheet 2 of 3)

c. Release the latches on the sides of the slide rails and pull the mounting box out to the full extension of the rails.

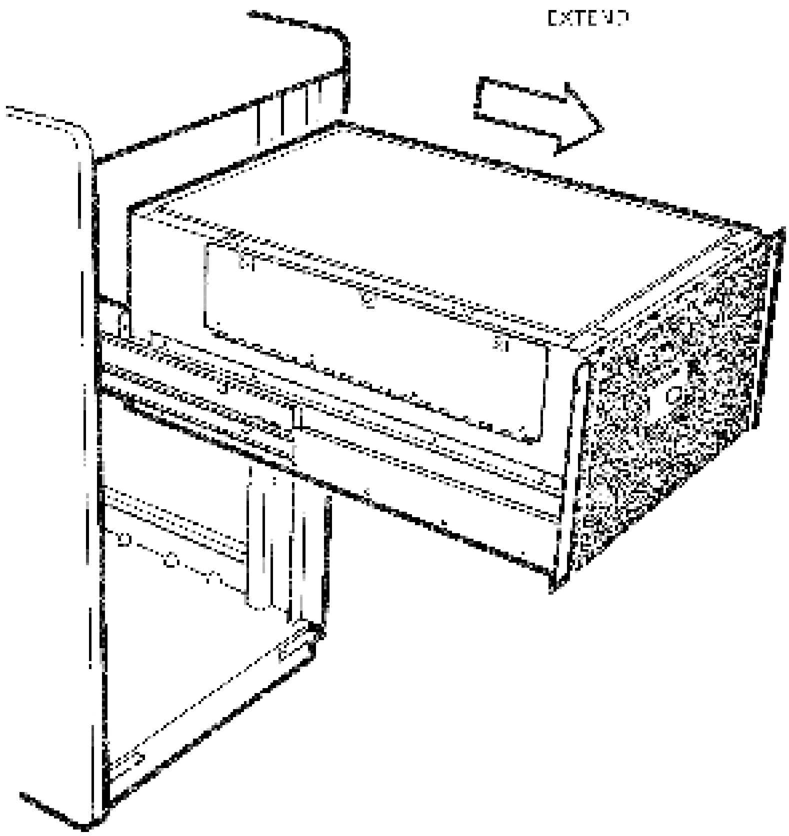


Figure 2-27 CIPA Mounting Box – Access (Sheet 3 of 3)

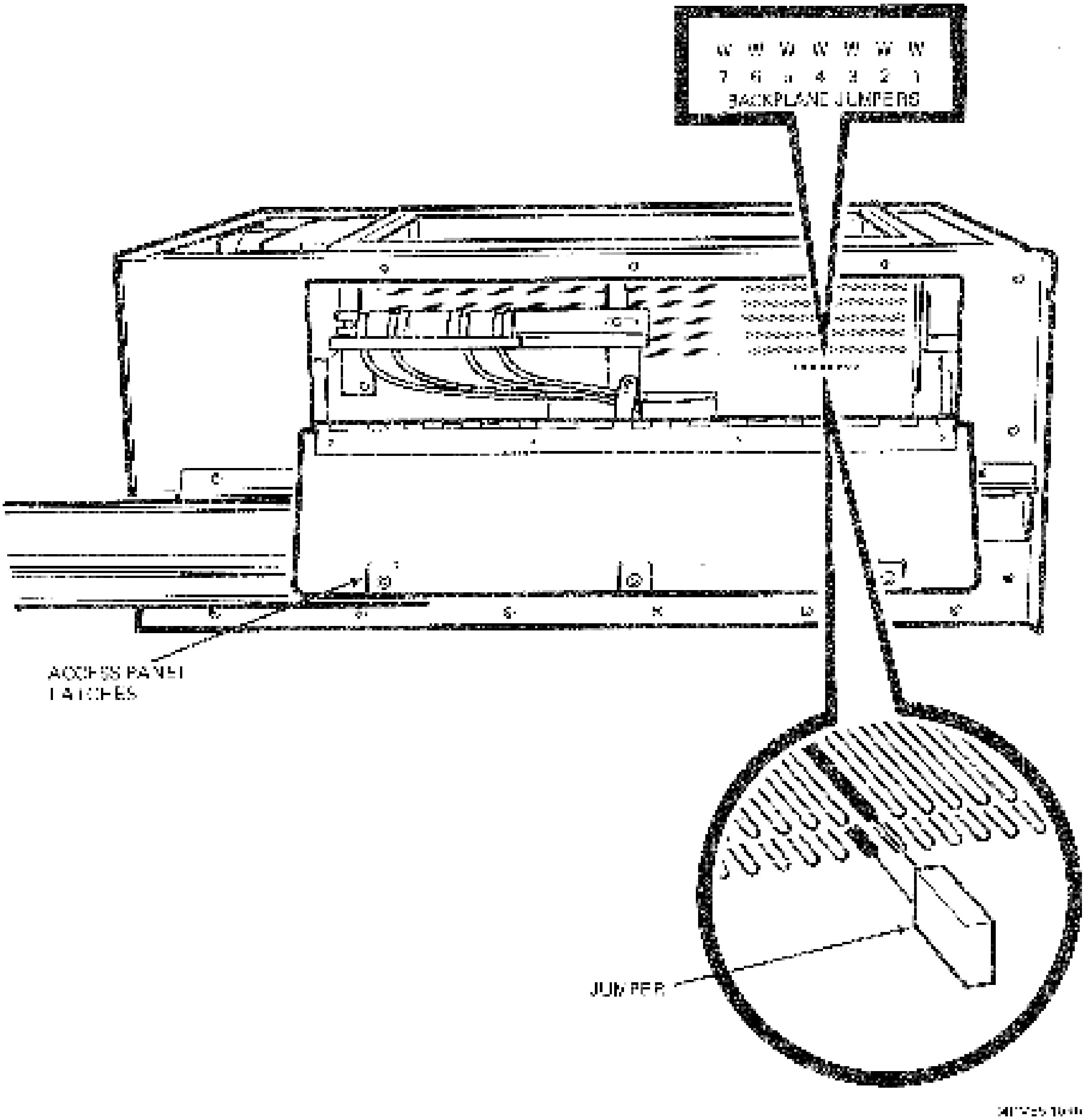


Figure 2-28 CIPA Mounting Box – Backplane Jumpers

2. Unlatch and open the CIPA backplane access cover located on the left of the CIPA mounting box for access to the jumpers (see Figure 2-28).

NOTE

For additional information on the CIPA backplane jumpers, refer to Appendix H.

2.7.3 Node Address Switch Verification

Two dual-inline switchpacks on the CI link interface module (L0108) provide the host system with a unique CI-node address within the VAXcluster. This address is typically a number from 0 to 15. To assign a CI-node address, the switches on each of the switch packs (S1 and S2) must be set to the binary value of the assigned number.

Procedure:

1. Remove the top access cover of the CIPA mounting box by removing the cover fasteners (see Figure 2-29).

2. Loosen the two screws that secure the CIPA cardcage to the chassis.

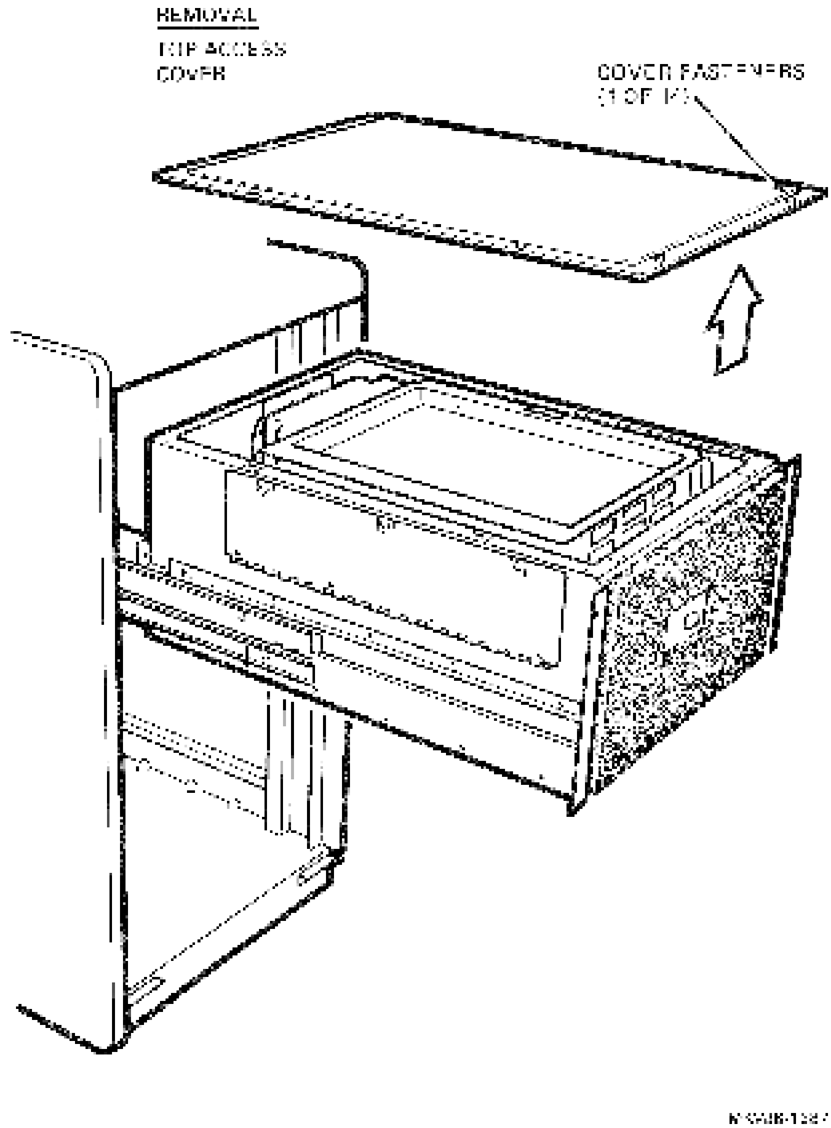


Figure 2-29 CIPA Mounting Box – Module Access (Sheet 1 of 4)

3. Raise the CIPA cardcage to its upright position (see Figure 2-29) by lifting the cardcage all the way up until the safety latch button locks the cardcage in its upright position.

4. Determine the CI-node address to be assigned.

   a. For CIBCI installations that create a new VAXcluster, select a CI-node address within the range of the number of CI bus ports being installed.

   b. For CIBCI installations that add a CI node to an existing VAXcluster, choose a node address not currently assigned.

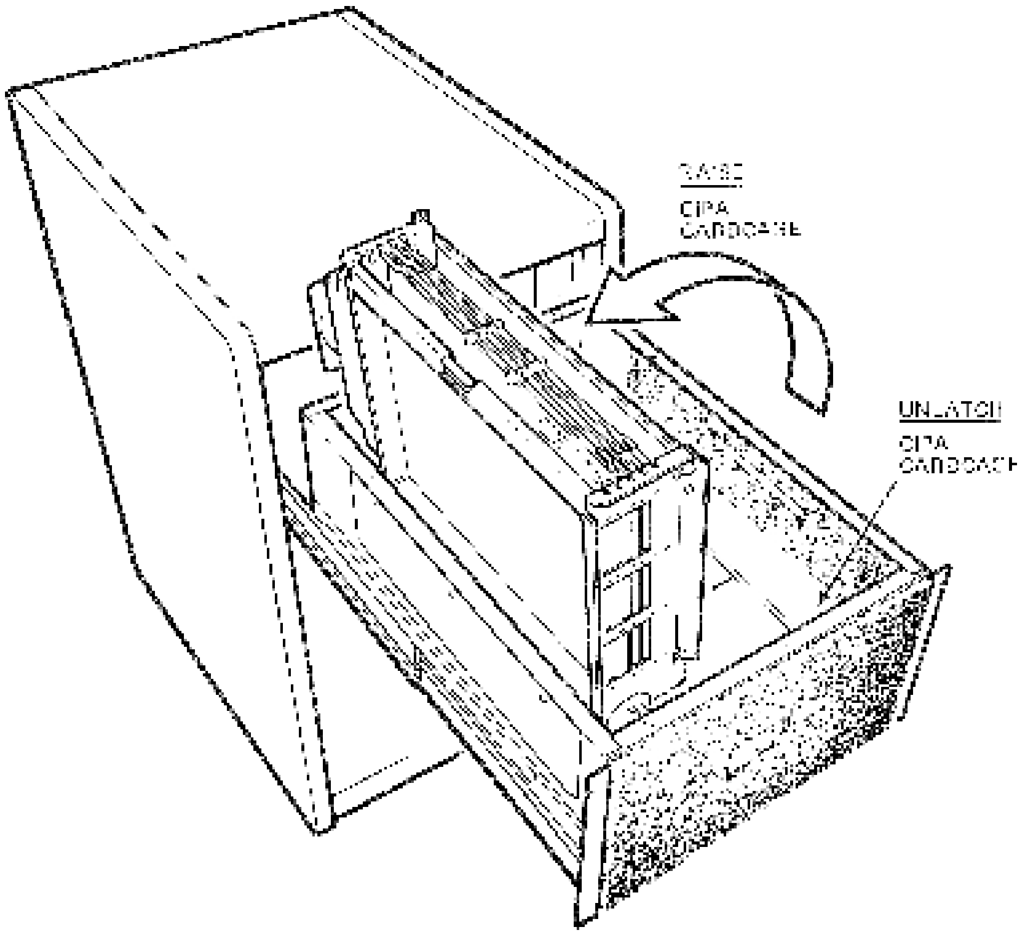


Figure 2-29 CIPA Mounting Box - Module Access (Sheet 2 of 4)

5. Configure the CI-node address switches by setting S1 and S2 to the selected address. The on position of each switch represents a logical zero and the off position a logical one (see Table 2-11.

6. Lower the cardcage by holding the cardcage with one hand and pulling the safety latch button with the other hand.

7. Tighten the two screws to secure the CIPA cardcage to the chassis.

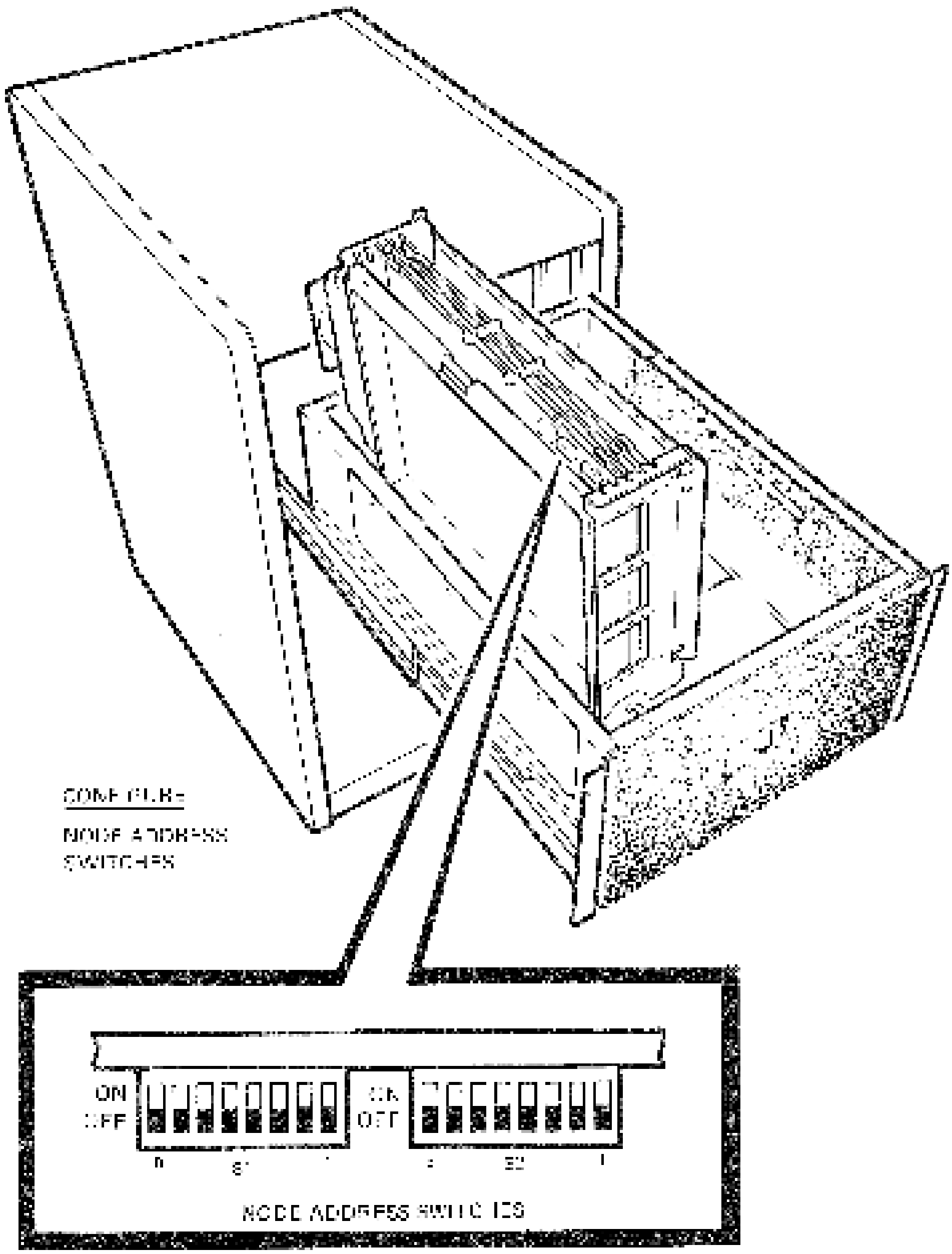


Figure 2-29 CIPA Mounting Box – Module Access (Sheet 3 of 4)

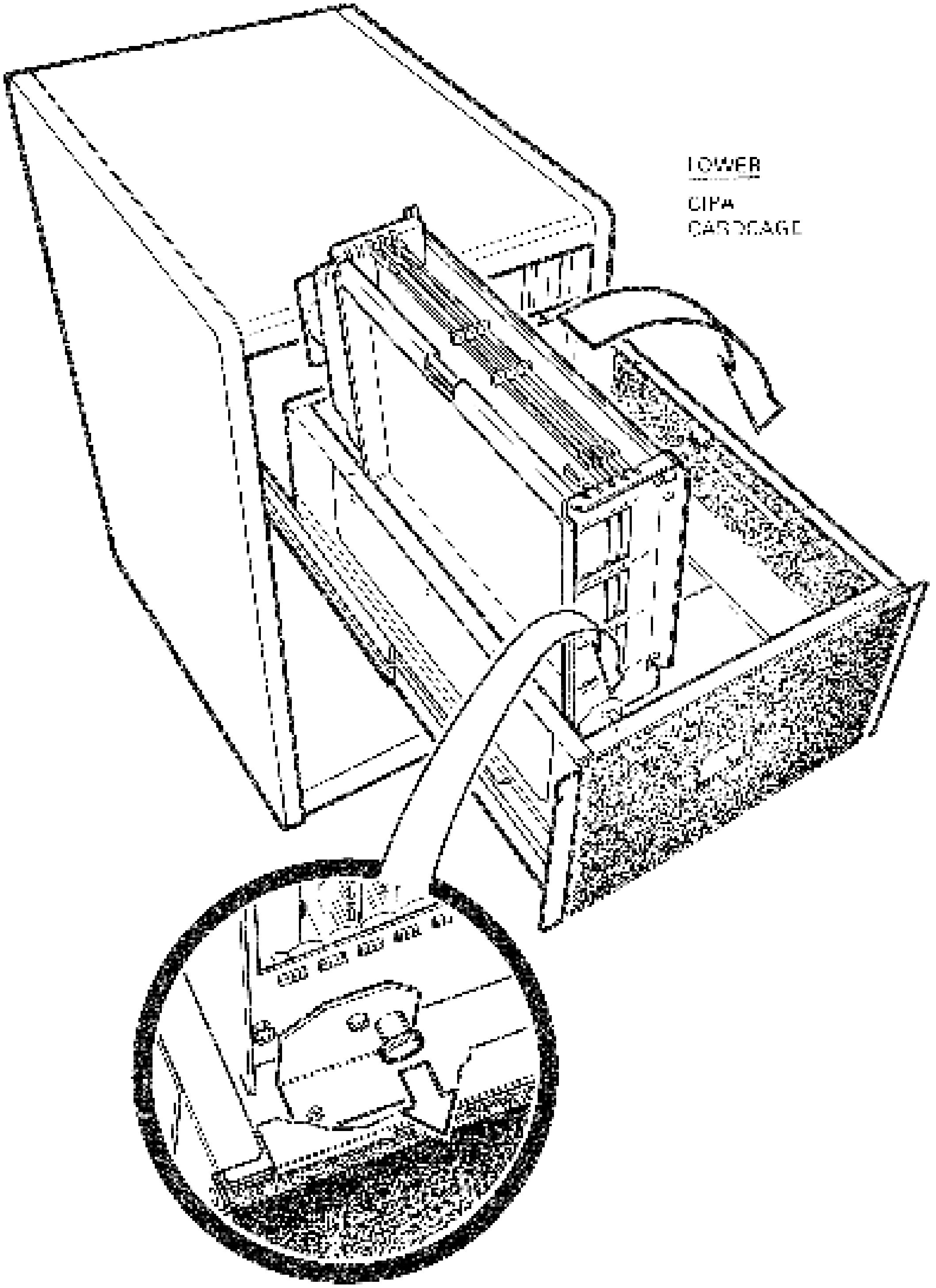


Figure 2-29 CIPA Mounting Box - Module Access (Sheet 4 of 4)

Table 2-[illegible] Node Address Switch Settings (S1 and S2 Switch Packs)

Node Number	Switch Number 4	3	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

NOTE:
On — logical 0
Off — logical 1
Switch positions 5 through 8 MUST be set to the ON position.

8. Locate the correct node address identification label. This label is part of a set (PN 3619264-17) shipped with the star coupler hardware.

NOTE
Refer to the *Star Coupler User's Guide* (EK-SC004-UG or EK-SC008-UG) for additional information.

9. Remove the paper backing from the address label. Place it on the outside of the CPU cabinet's back door (in a visible location).

10. Replace the top access cover. Then, lower and retract the CIPA mounting box and replace the cabinet front and rear doors.

NOTE
Do NOT install the BNCIA cables between the CIBCI and the star coupler hardware at this time.

11. Proceed to Chapter 3 for acceptance testing.

CHAPTER 3
ACCEPTANCE VERIFICATION

3.1 INTRODUCTION
Chapter 3 contains information on acceptance verification, including:

Power Setup and Verification – Applying ac power to the system and verifying proper dc power in the CIPA mounting box.

Diagnostic Verification – Verifying the functionality of the CIBCI hardware by running diagnostic tests with the system in a stand-alone environment.

Maintenance Verification – Facilitating VAXcluster maintenance by describing the tools that are required and/or provided for individual nodes or options within a VAXcluster system.

3.2 POWER SETUP AND VERIFICATION
Each CIBCI option is shipped with internal and external BNCIA-xx cabling (CIPA bus or buses and ac power cables) completed within the CIPA mounting box or CIPA cabinet. Verify this cabling, effect the necessary inter/intracabinet cabling, and complete the system verification in the sequence given in the following two sections.

CAUTION
Ensure that the front panel keylock switch of the VAX 8200 or VAX 8700 is in the off position before system modules and cables are installed.

NOTE
It is assumed here that the cabinet doors of the CPU(s) and the CIPA are still off as part of the cabinet mating procedure discussed in Sections 2.5.1 and 2.5.2.

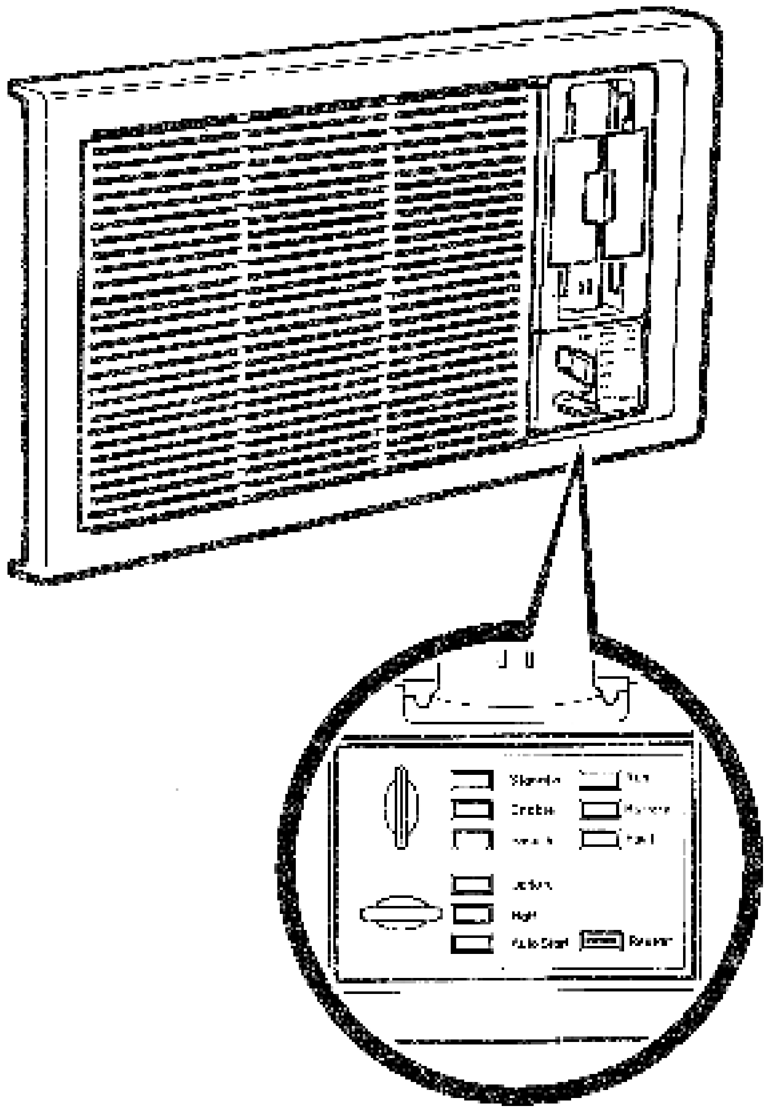
3.2.1 VAX 8200/8300 Systems

Procedure:

1. Switch the main circuit breakers on all ac power controllers to their OFF positions.
2. Place the remote/local switches on all ac power controllers in their remote positions.
3. Connect the ac power cable from the CIPA cabinet's ac power controller to an external ac power source.
4. Switch the main circuit breakers on all ac power controllers to their ON positions.

5. Insert a plastic switch-key into switch S2 (the lower switch) on the control panel and turn the key clockwise to the halt enable (HALT EN) position (number 2 if you have the international control panel), as shown in Figure 3-1.

6. Insert a second plastic switch-key into switch S1 (the upper switch) on the control panel and turn the key clockwise to the enabled position (the symbol "|" if you have the international control panel), as shown in Figure 3-1.



MKV85-1543

Figure 3-1 Control Panel Switches and LEDs

7. Verify the CIPA mounting box dc voltages by checking the LEDs on the top of the power supply. All LEDs should be lit (on).

8. Turn off electrical power to the system by placing the plastic switch-key located in switch S1 on the control panel to its most counterclockwise position.

9. Carefully lower the CIPA cardcage cover into the CIPA mounting box and secure it to the chassis.

10. Replace the top cover on the CIPA mounting box and slide the box back into the cabinet.

11. Replace the front door on the H9642 CIPA cabinet and check that the door makes positive contact with each RFI gasket spring on ALL edges.

NOTE
Do NOT replace the rear door on the H9642 CIPA cabinet at this time.

12. Replace all covers, doors, and panels on the VAX 8200/8300 system cabinets.

13. Turn the plastic switch-key located in switch S1 (the upper switch) on the control panel clockwise to the enabled position (the symbol ¶ ¶ if you have the international control panel), as shown in Figure 3-1.

14. Observe that the red status fault indicator located on the control panel is on, indicating that the CPU is performing its self-test routines.

    If the CPU successfully passes its self-test, you will see the following display on the console terminal:

```
#ABCDEFGHIJKLMNP
```

or

```
#ABCDEFGHIJKLMNP
```

    If the CPU fails its self-test, consult the *VAX 8200 Owner's Manual* or *VAX 8300 Owner's Manual*.

15. Observe the successful completion of system-wide hardware initialization by verifying that the red status fault indicator is turned off, and that the terminal contains a hexadecimal and periods display.

    If the red status fault indicator remains on, and one or more of the hexadecimal digits in the display has a minus sign before it, an adapter hardware self-test failure was detected. Consult the *VAX 8200 Owner's Manual* or *VAX 8300 Owner's Manual*.

Procedure:

1. Switch the main circuit breakers on all ac power controllers to their OFF positions.

2. Place the remote/local switch on all ac power controllers in the remote position.

3. Switch the main circuit breaker on all ac power controllers to their ON positions.

4. Place the power on/off switch of the PC380 console system in its ON position.

5. Observe that a DIGITAL logo is displayed on the console video terminal screen. This display indicates that the PC380 console system is performing its internal self-test routines.

NOTE

If the PC380 console system successfully passes its self-tests, you will observe the console prompt (>>>) on the screen. Otherwise, an image of the failure problem will be displayed. In the case of a failure, consult the *PC380 Console User's Guide*.

6. Apply power to the system by entering the POWER ON console command language (CCL) command. Use the PC380 console system keyboard as follows:

```
>>> POWER ON
```

7. Verify the CIPA mounting box dc voltages by checking the LEDs on the top of the power supply. All LEDs should be lit (on).

8. Disconnect power to the system by entering the POWER OFF console command language (CCL) command. Use the PC380 console system keyboard as follows:

```
>>> POWER OFF
```

9. Carefully lower the CIPA cardcage cover into the CIPA mounting box and secure it to the chassis.

10. Replace the top cover on the CIPA mounting box and slide the box back into the cabinet.

11. Replace the front door on the H9652 expander cabinet and check that the door makes positive contact with each RFI gasket spring on ALL edges.

NOTE

Do NOT replace the rear door on the H9652 expander cabinet at this time.

12. Replace all covers, doors, and panels on the system cabinetry.

13. Reapply power to the system by entering the POWER ON console command language (CCL) command. Use the PC380 console system keyboard as follows:

```
>>> POWER ON
```

3.3 DIAGNOSTIC VERIFICATION

To determine if the CIBCI adapter hardware is functioning properly, eight level 3 diagnostic programs must be executed. These diagnostic programs, along with the their appropriate diagnostic supervisor program, are contained on separate RX50 floppy diskettes.

Six of the eight level 3 diagnostic programs are executed with the system operating in a stand-alone environment (not connected to a VAXcluster and not running under the VMS operating system). This is referred to as repair level testing.

Two of the eight level 3 diagnostic programs are then executed in order to test the functionality of the hardware. This is referred to as functional level testing. (See Tables 3-1 and 3-2.)

Table 3-1 List of the CIBCI Diagnostic Programs

Program Designation	Program Title
EVCKA	CIBCI repair level diagnostic 1
EVCKB	CIBCI repair level diagnostic 2
EVCKC	CIBCI repair level diagnostic 3
EVCKD	CIBCI repair level diagnostic 4
EVCKE	CIBCI repair level diagnostic 5
EVCKF	CIBCI repair level diagnostic 6
EVGAA	CI functional diagnostic 1
EVGAB	CI functional diagnostic 2

Table 3-2 Summary of the Diagnostic Testing Hierarchy

Diagnostic Category	Diagnostic Program Level	Testing Function
Repair	Level 3	Tests the [illegible] operation of the CIBCI adapter.
Functional	Level 3	Tests the functional hardware of the CIBCI adapter.
Exerciser	Level 4	Tests the communications between CI nodes.
		Detects a failing CI node.
		Verifies the repair of a failing CI node.

3.3.1 VAX 8200/8300 Systems

3.3.1.1 Preliminary Setup Before running the diagnostics, make the following CI bus loopback connections on the CI bulkhead connector panel located at the back of the CIPA cabinet (see Figure 3-2).

Procedure:

1. Using one of the attenuator pads (P/N 12-19907-01) and two of the modularity cables (P/N 70-18530-00) supplied in the CIxxx control distribution (CD) kit (A2-W0865-10), connect transmit A (J22) to receive A (J24).

2. Perform the same connection for path B using the other attenuator and two modularity cables from the CIxxx control distribution (CD) kit, part number A2-W0865-10. Connect transmit B (J21) to receive B (J23).

NOTE
For more information on CI bus termination, refer to Appendix A.

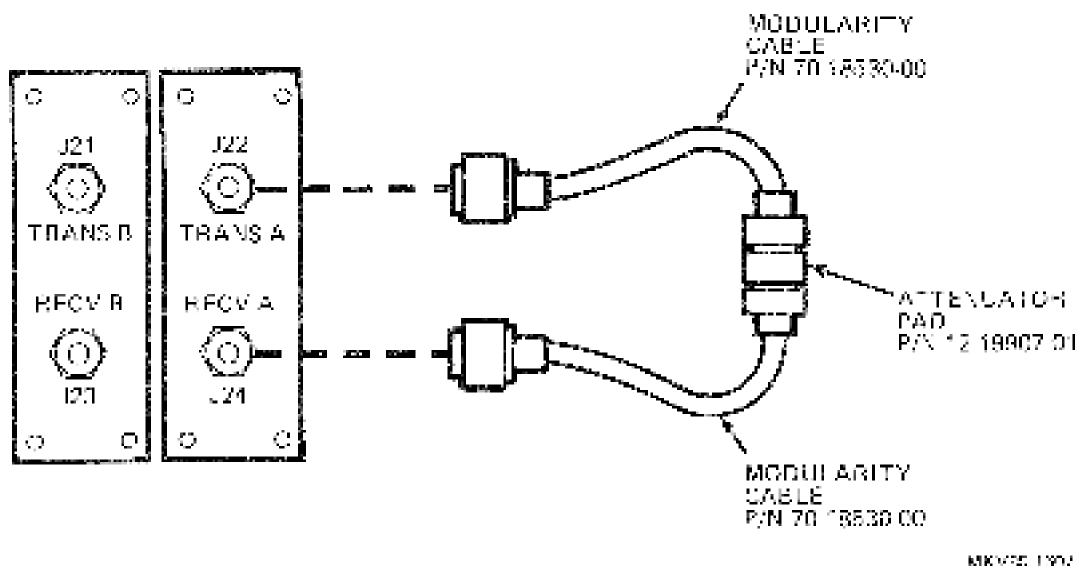


Figure 3-2 Diagnostic Loopback Cable Connections

3.3.1.2 Loading the Diagnostic Supervisor Program

Procedure:

1. Insert the RX50 diskette containing file ERSAA.EXE into the console RX50 disk drive unit 0.

2. Load the diagnostic supervisor program into physical memory by entering the following CCL command at the console terminal:

```
>>> B
XX
DIAGNOSTIC SUPERVISOR
```

3. Identify the CIBCI adapter and its node configuration parameters to the diagnostic supervisor program, as follows:

```
DS> ATTACH CIBCI HUB PAA0 6 4 0
```

4. Select the CIBCI adapter as the unit under test, as follows:

```
DS> SELECT PAA0
```

5. Show the unit selected, as follows:

```
DS> SHOW SELECT
```

3.3.1.3 **Repair Level Testing** A minimum of five (5) successful passes of each diagnostic program must be completed to satisfy acceptance testing requirements. Examples 3-1 through 3-6 provide trace printouts for diagnostics EVCKA through EVCKF, respectively.

NOTE
Help files are available under the diagnostic supervisor for all diagnostic programs including the supervisor program itself.

Procedure:

1. Remove the RX50 diskette from the console RX50 disk drive unit 0.

2. Insert the RX50 diskette containing files EVCKA.EXE through EVCKF.EXE into the console RX50 disk drive unit 0.

3. Load the EVCKA diagnostic program, as follows:

```
DS> LOAD EVCKA (first repair level diagnostic)
```

4. Set the desired diagnostic supervisor control flags to enable printing of the number and title of each test before it is executed and to halt on a detected error. Set the flags as follows:

```
DS> SET FLAGS TRACE, HALT
```

5. Start the diagnostic program, as follows:

```
DS> START/PASS:5
```

6. Repeat steps 3 through 5 to load and execute the remaining repair level diagnostics (EVCKB through EVCKF).

```
DS> LOAD EVCKA
DS> SET FLAGS TRACE, HALT
DS> SET EVENT FLAGS 4
DS> START/PASS:5

..Program: CIBCI - EVCKA Repair level, revision 1.0, 28 tests,
  at 00:24:40.75.
  Testing: _PAA0

SET EVENT FLAG 4 FOR REV LEVEL OF BIIC IN TEST 3

  Test  1:    ERROR INTERRUPT CONTROL TEST
  Test  2:    DEVICE TYPE REGISTER TEST
  Test  3:    DC AND DIBCI  SELF TEST
  REVISION LEVEL OF BIIC CHIP ON CIBCI IS: 0

  Test  4:    CNFGR - L WRITE ACCESS TEST
  Test  5:    CNFGR - L READ ACCESS TEST
  Test  6:    R/W TEST OF DIAG BIT IN CNFGR
  Test  7:    CNFGR - L READ ACCESS TEST - AFTER DISABLING UCSREN IN BCICR
  Test  8:    CNFGR - L READ ACCESS TEST - AFTER DISABLING STS IN BCICR
  Test  9:    PORT DATA REGISTER - R/W TEST - SOURCE IS BI
  Test 10:    R/W TEST OF BUFFERED COMMAND ADDRESS REGISTER (BCAR)
  Test 11:    R/W TEST OF BCMR
  Test 12:    R/W TEST OF DMA REGISTER
  Test 13:    RECEIVED COMMAND DATA PATH TEST
  Test 14:    R/W TEST OF CNFGR,BCAR AND BCMR TAKEN ALTOGETHER
  Test 15:    SIZE OF TRANSFER TEST
  Test 16:    DMA FILE - R/W COUNTER TEST
  Test 17:    DMA FILE - COUNTER SEQUENCE TEST
  Test 18:    R/W TEST OF BCAR AND BCMR USING THE MASTER SEQUENCER
  Test 19:    BICA ADDRESS REGISTER TEST
  Test 20:    STOP TEST
  Test 21:    PORT DATA REGISTER - CIPA DATA PATH TEST
  Test 22:    WITH DIAG BIT CLEAR, R/W TEST OF BCAR
  Test 23:    WITH DIAG BIT CLEAR, R/W TEST OF DMA REGISTER
  Test 24:    CIPACD REGISTER READ TEST (CIPA BUS READ TEST)
  Test 25:    L READ ACCESS TEST OF LS AFTER DISABLING UCSREN IN BCI CONTROL REG
  Test 26:    NUACK TEST FOR NODE ADDRESS 200
  Test 27:    L READ ACCESS TEST OF LS AFTER DISABLING STS IN THE BICSR
  Test 28:    USER INTERRUPT CONTROL TEST

..End of run, 0 errors detected, pass count is 1,
  time is 15-JUL 1985 00:24:52.74
DS>
```

Example 3-1   Trace Printout for Repair Diagnostic EVCKA

```
DS> LOAD EVCKB
DS> SET FLAGS TRACE, HALT
DS> START/PASS:5

..Program: CIBC] - EVCKB Repair level, revision 1.0, 27 tests,
  at 00:25:50.39.
  Testing: _PAA0

  Test   1:     BUSIB/ID IN DATA PATHS TEST
  Test   2:     PMCSR ACCESS TEST
  Test   3:     PMCSR - BIT READ/WRITE TEST
  Test   4:     INITIALIZE TEST
  Test   5:     MADR/BUS MD DATA PATHS TEST
  Test   6:     LOCAL STORE DUAL ADDRESS TEST
  Test   7:     LOCAL STORE READ/WRITE RAM TEST
  Test   8:     LOCAL STORE DYNAMIC MEMORY TEST
  Test   9:     INTERLOCKED READ/WRITE TEST
  Test  10:     VCDT - READ/WRITE RAM TEST
  Test  11:     VCDT DUAL ADDRESS TEST
  Test  12:     VCDT DYNAMIC MEMORY TEST
  Test  13:     CONTROL STORE - DUAL ADDRESS TEST
  Test  14:     CONTROL STORE - READ/WRITE RAM TEST
  Test  15:     CONTROL STORE RAM DYNAMIC MEMORY TEST
  Test  16:     CONTROL STORE ROM INSERTION TEST
  Test  17:     REGISTER DUAL ADDRESS TEST
  Test  18:     BUSIB SOURCE=LIT DEST=LS[LIT]
  Test  19:     BUSIB SOURCE EQUALS ALU
  Test  20:     BUSIB DESTINATION IS VCDT[LIT]
  Test  21:     BUSIB SOURCE EQUALS LS[LIT]
  Test  22:     BUSIB SOURCE EQUALS VCDT[LIT]
  Test  23:     BUSIB DESTINATION EQUALS LS[INDEX]
  Test  24:     INDEX REGISTER SA0/SA1 CHECK
  Test  25:     BUSIB SOURCE LS[INDEX]
  Test  26:     BUSIB DESTINATION EQUALS LS[XLATE]
  Test  27:     BUSIB SOURCE EQUALS LS[XLATE]
..End of run, 0 errors detected, pass count is 1,
  time is 15-JUL-1985 03:29:37.92
DS>
```

Example 3-2 Trace Printout for Repair Diagnostic EVCKB

```
DS> LOAD EVCKC
DS> SET FLAGS TRACE, HALT
DS> START/PASS:5

 .Program: CIBCI - EVCKC Repair level, revision 1.0, 33 tests,
  at 00:30:00.96.
  Testing: _PAA0

  Test  1:    2911 SEQUENCER JUMP TEST
  Test  2:    CONTROL STORE PARITY ERROR TEST
  Test  3:    "2901" RAM DUAL ADDRESS TEST
  Test  4:    "2901" RAM/Q STUCK BIT TEST
  Test  5:    "2901" RAM/Q REGISTER SHIFT
  Test  6:    "2901" ALU FUNCTION TEST
  Test  7:    "2901" CONDITION CODE Z BRANCH TEST.
  Test  8:    "2901" CONDITION CODE N BRANCH TEST.
  Test  9:    "2901" CONDITION CODE V BRANCH TEST.
  Test 10:    "2901" CONDITION CODE C BRANCH TEST.
  Test 11:    2911 SEQUENCER UPC+1 TEST
  Test 12:    2911 SEQUENCER JSR TEST
  Test 13:    POP!! MICROSTCK
  Test 14:    BUS IB<00> BRANCH TEST
  Test 15:    BUS IB<08> BRANCH TEST
  Test 16:    BUS IB<12> BRANCH TEST
  Test 17:    BUS IB<15> BRANCH TEST
  Test 18:    BUS IB<20> BRANCH TEST
  Test 19:    BUS IB<21> BRANCH TEST
  Test 20:    BUS IB<24> BRANCH TEST
  Test 21:    BUS IB<31> BRANCH TEST
  Test 22:    BUS IB<10> <09> BRANCH TEST
  Test 23:    BUS IB<14> <13> BRANCH TEST
  Test 24:    BUS IB<26> <22> BRANCH TEST
  Test 25:    BUS IB<26> <25> BRANCH TEST
  Test 26:    BUS IB<19> <18> <17> <16> BRANCH TEST
  Test 27:    MAINTENANCE TIMER DISABLE BRANCH TEST
  Test 28:    TICK BRANCH TEST
  Test 29:    REGISTER WRITTEN BRANCH T1
  Test 30:    REGISTER WRITTEN BRANCH T2
  Test 31:    XBDR - PORT INITIATED WRITE TEST
  Test 32:    BICA CMMD ADDR REG - PORT INITIATED WRITE TEST
 .Test 33:    BYTE MASK - PORT INITIATED WRITE TEST
..End of run, 0 errors detected, pass count is 1,
  time is 15-JUL-1985 00:32:56.24
DS>
```

Example 3-3 Trace Printout for Repair Diagnostic EVCKC

```
DS> LOAD EVCKD
DS> SET FLAGS TRACE, HALT
DS> START/PASS:5

..Program: CIBCI - EVCKD Repair level, revision 1.0, 21 tests,
  at 00:33:15.20.
  Testing: _PAA0

  Test  1:    EXTERNAL BUS LONGWORD WRITE TO MEMORY TEST
  Test  2:    LOCAL STORE PARITY ERROR TEST
  Test  3:    DYNAMIC LOCAL STORE MOVING INVERSIONS
  Test  4:    DYNAMIC VCDT MOVING INVERSIONS
  Test  5:    EXTERNAL BUS LONGWORD READ TO MEMORY TEST
  Test  6:    EXTERNAL BUS INTERLOCK READ TO MEMORY TEST
  Test  7:    EXTERNAL BUS INTERLOCK WRITE TO MEMORY TEST
  Test  8:    EXTERNAL BUS LONGWORD WRITE TO NXM TEST
  Test  9:    CORRECTABLE READ DATA TEST FOR VAX-11/750
  TEST IGNORED FOR THIS PROCESSOR

  Test 10:    READ DATA SUBSTITUTE TEST FOR VAX-11/750
  TEST IGNORED FOR THIS PROCESSOR

  Test 11:    READ DATA SUBSTITUTE TEST FOR VAX 8700
  TEST IGNORED FOR THIS PROCESSOR

  Test 12:    CORRECTABLE READ DATA TEST FOR VAX 8200
  Test 13:    READ DATA SUBSTITUTE TEST FOR VAX 8200
  Test 14:    EXTERNAL BUS EXTENDED WRITES TEST
  Test 15:    EXTERNAL BUS EXTENDED READS TEST
  Test 16:    EXTERNAL BUS MASK REGISTER TEST
  Test 17:    INTERRUPT TEST
  Test 18:    MTE DURING INTERRUPT TEST
  Test 19:    CIPA BUS PARITY ERROR (CBPE) TEST
  Test 20:    SUSPEND AND EXECUTE TEST
  Test 21:    PACKET BUFFER OUT/IN REG LOOPBACK TEST
..End of run, 0 errors detected, pass count is 1,
  time is 15-JUL-1985 00:34:43.62
DS>
```

Example 3-4 Trace Printout for Repair Diagnostic EVCKD

```
DS> LOAD EVCKE
DS> SET FLAGS TRACE, HALT
DS> START/PASS:5

..Program: CIBCI - EVCKE Repair level, revision 1.0, 13 tests,
  at 00:34:59.93.
  Testing: _PAA0

  Test  1:    PACKET BUFFER SELECT TEST
  Test  2:    OUTPUT PARITY ERROR TEST GENERATED BY PBTR
  Test  3:    TRANSMIT BUFFER "A" PATH/ADDR CHECK
  Test  4:    TRANSMIT BUFFER "B" PATH/ADDR CHECK
  Test  5:    RECEIVE BUFFER "A" PATH/ADDR CHECK
  Test  6:    RECEIVE BUFFER "B" PATH/ADDR CHECK
  Test  7:    TRANSMIT BUFFER "A" SA1/SA0
  Test  8:    TRANSMIT BUFFER "B" SA1/SA0
  Test  9:    RECEIVE BUFFER "A" SA1/SA0
  Test  10:   RECEIVE BUFFER "B" SA1/SA0
  Test  11:   FORCE RECEIVE BUFFER PARITY ERROR
  Test  12:   RECEIVE BUFFER "A" OVERFLOW TEST
  Test  13:   RECEIVE BUFFER "B" OVERFLOW TEST
..End of run, 0 errors detected, pass count is 1,
  time is 15-JUL-1985 00:36:29.06
DS>
```

Example 3-5 Trace Printout for Repair Diagnostic EVCKE

```
DS> LOAD EVCKF
DS> SET FLAGS TRACE, HALT
DS> START/PASS:5

..Program: CIBCI - EVCKF Repair level, revision 1.0, 14 tests,
  at 00:38:31.19.
  Testing: _PAA0

  Test  1:    INTERNAL MAINTENANCE LOOP TEST
  Test  2:    INTERNAL MT LOOPBACK WHILE LOADING XMIT BUFFER TEST
  Test  3:    INTERNAL MT LOOP TEST WITH ONE RCV BUF AVAILABLE
  Test  4:    INTERNAL MT LOOP TEST WITH NO RCV BUF'S AVAILABLE
  Test  5:    INTERNAL MAINT LP WITH SWAP NODE ADDRESS
  Test  6:    TRANSMIT BUFFER PARITY ERROR TEST
  Test  7:    ALTERNATING PACKET BUFFER UNLOAD TEST
  Test  8:    ARBITRATION TEST N+1+1
  Test  9:    EXTERNAL MAINT. LOOP PATH "A"
  Test  10:   EXTERNAL MAINT. LOOP PATH "B"
  Test  11:   EXT. MAINT. LOOP "RECEIVERS DISABLED"
  Test  12:   EXT. MAINT. LOOP "ABORTING TRANSMISSION"
  Test  13:   "ACKNOWLEDGE TIMEOUT" TEST
  Test  14:   EXTERNAL BUS LONGWORD WRITE TO ITSELF (LOCAL STORE)
..End of run, 0 errors detected, pass count is 1,
  time is 15-JUL-1985 00:40:26.82
DS>
```

Example 3-6 Trace Printout for Repair Diagnostic EVCKF

3.3.1.4 CI Bus Cable Testing – After successfully completing five passes of each of the six repair level diagnostics, remove the attenuator pads and modularity cables from the CI bulkhead connector panels (J21-J24) and perform the following steps:

Procedure:

1. Verify that this CIBCI port has a unique node address within the VAXcluster before connecting any cables.
2. Locate the set of four CI bus coaxial cables (BNCIA-xxx) and connect one end of each cable to the appropriate CI bulkhead connector panels.

NOTE

The coaxial CI bus cables may be connected or removed from the CI bulkhead connector panels without powering down either the system or the CIPA cabinet. DO NOT unroll or route the CI bus cables at this time.

3. Connect the two attenuator pads to the free ends of the coaxial CI bus cables. Be sure to connect transmit A to receive A, and transmit B to receive B.
4. Run five passes of the external loop section of the diagnostic program EVCKF to test the CI bus cables, as follows:

```
DS> RUN EVCKF/SECTION:EXTM_LOOP/PASS:5
```

3.3.1.5 Functional Level Testing – With the CI bus cables and attenuator pads providing signal loop-back, load and run the CI functional diagnostics EVGAA and EVGAB. A minimum of five passes of each diagnostic must be completed to satisfy acceptance testing requirements. Examples 3-7 and 3-8 show trace printouts for diagnostics EVGAA and EVGAB, respectively.

Procedure:

1. Ensure that the RX50 diskette containing files EVGAA.EXE and EVGAB.EXE is installed in the console RX50 disk drive unit 0.
2. Load the EVGAA diagnostic program, as follows:

   `DS> LOAD EVGAA` (first functional diagnostic)

3. Set event flags 1 and 2 to reload the CI microcode and output the port queue entries. This is always required after running the repair level diagnostics. Set the event flags as follows:

```
DS> SET EVENT FLAG 1, 2
```

4. Set the desired diagnostic supervisor control flags to enable printing of the number and title of each test before it is executed and to halt on a detected error, as follows:

```
DS> SET FLAGS TRACE, HALT
```

```
DS> LOAD EVGAA
DS> SET EVENT FLAGS 1, 2
DS> SET FLAGS TRACE, HALT
DS> START/PASS:5

..Program: EVGAA - CI FUNCTIONAL PART I, revision 2.5, 17 tests,
  at 00:48:21.95.
  Testing: _PAA0

  EVENT FLAG 1:
  DIAGNOSTIC WILL LOAD CI RAM UCODE
  FROM THE DEFAULT LOAD PATH.

  EVENT FLAG 2:
  OUTPUT THE PORT QUEUE ENTRIES.

  EVENT FLAG 3:
  INVOKES THE REQUEST ID LOOP FUNCTION.

  ROM REVISION = 3          WCS REVISION = 4

  Test  1:    CLUSTER CONFIGURATION

                      CLUSTER CONFIGURATION -- PATH A
                      *******************************

  NOTE.
  YOU CANNOT DIFFERENTIATE BETWEEN A CI780 AND CI750 REMOTELY.

  NODE #   DEVICE TYPE    ROM/WCS REV.    PORT FUNCTIONALITY    PATH TYPE
  ------   -----------    ------------    ------------------    ---------
    0         CI7X0           3    4          FFFFFF00(X)       DUAL PATH

                      CLUSTER CONFIGURATION -- PATH B
                      *******************************

  NODE #   DEVICE TYPE    ROM/WCS REV.    PORT FUNCTIONALITY    PATH TYPE
  ------   -----------    ------------    ------------------    ---------
    0         CI7X0           3    4          FFFFFF00(X)       DUAL PATH

  Test  2:    SETCKT TEST WITH VARIOUS MASKS AND M_VALUES
  Test  3:    SETCKT TEST FOR EACH VALID PORT
  Test  4:    SETCKT TEST FOR NVALID PORT
  Test  5:    REQID TEST
  Test  6:    REQID TEST WITH 6 PACKETS ON DGFQ
  Test  7:    DATAGRAM DISCARD TEST
  Test  8:    RESPONSE QUEUE AVAILABLE INTERRUPT TEST
  Test  9:    SEND DATAGRAM -SNDDG- TEST
  Test  10:   SNDMSG TEST WITH NOVIRTUAL CIRCUIT TEST
  Test  11:   SEND MESSAGE TEST, CROSSING PAGE BOUNDARY
  Test  12:   MESSAGE LENGTH TEST
  Test  13:   PACKET SIZE VIOLATION TEST
  Test  14:   SEND LOOPBACK -SNDLB- TEST
  Test  15:   SNDLB TEST, FULL BUFFER PATH A
  Test  16:   SNDLB TEST, FULL BUFFER PATH B
  Test  17:   SNDLB TEST, BOTH PATHS
..End of run, 0 errors detected, pass count is 1,
  time is 15-JUL-1985 00:50:40.36
DS>
```

Example 3-7 Trace Printout for Functional Diagnostic EVGAA

6. Start the EVGAA diagnostic program, as follows:

```
DS> START/PASS:5
```

7. After five successful passes, remove the console floppy diskette, insert the diagnostic floppy diskette, clear event flags 1 and 2, and then load and run the EVGAB diagnostic program (second functional diagnostic) for five successful passes.

8. Turn off the electrical power to the system by placing the VAX 8200 control panel keylock switch (upper switch) in the OFF position.

```
DS> LOAD EVGAB
DS> CLEAR EVENT FLAG 1, 2
DS> SET FLAGS TRACE, HALT
DS> START/PASS:5

..Program: EVGAB - CI FUNCTIONAL PART II, revision 2.5, 12 tests,
  at 00:50:54.31.
  Testing: _PAA0

  ROM REVISION = 3        WCS REVISION = 4

  Test  1:    SEND DATA TEST, WITH OFFSET COMBINATIONS
  Test  2:    REQUEST DATA TEST, WITH OFFSET COMBINATIONS
  Test  3:    INVALIDATE TRANSLATION CACHE TEST
  Test  4:    SNDMDAT TEST, ENABLED/MAINTENANCE STATE
  Test  5:    SNDMDAT TEST, ENABLED STATE
  Test  6:    REQMDAT TEST, ENABLED/MAINT STATE
  Test  7:    REQMDAT TEST, ENABLED STATE
  Test  8:    SEND RESET TEST IN ENABLED STATE
  Test  9:    QUEUE CONTENTION TEST
  Test  10:   BUFFER READ ACCESS TEST
  Test  11:   BUFFER WRITE ACCESS TEST
  Test  12:   WRITE TO GLOBAL BUFFER TEST
..End of run, 0 errors detected, pass count is 1,
  time is 15-JUL-1985 00:52:29.92
DS>
```

Example 3-8 Trace Printout for Functional Diagnostic EVGAB

9. Replace the rear door on the CIPA cabinet and check that the door makes positive contact with each RFI gasket spring on all edges.

10. Disconnect the attenuator pads from the ends of the CI bus cables in preparation for routing and connecting the cables to the star coupler.

NOTE

For information on connecting the coaxial CI bus cables to the star coupler, refer to the *SC004 or SC008 Star Coupler User's Guide*.

3.3.2 VAX 8500/8550/8700/8800 Systems

3.3.2.1 Preliminary Setup Before running the diagnostics, make the following CI bus loopback connections on the CI bulkhead connector panel located at the back of the system cabinet (see Figure 3-3).

Procedure:

1. Using one of the attenuator pads (P/N 12-19907-01) and two of the modularity cables (P/N 70-18530-00) supplied in the CIxxx control distribution (CD) kit (P/N A2-W0865-10), connect transmit A (J22) to receive A (J24).

2. Perform the same connection for path B using the other attenuator and two modularity cables from the generic CIxxx control distribution (CD) Kit, part number A2-W0865-10. Connect transmit B (J21) to receive B (J23).

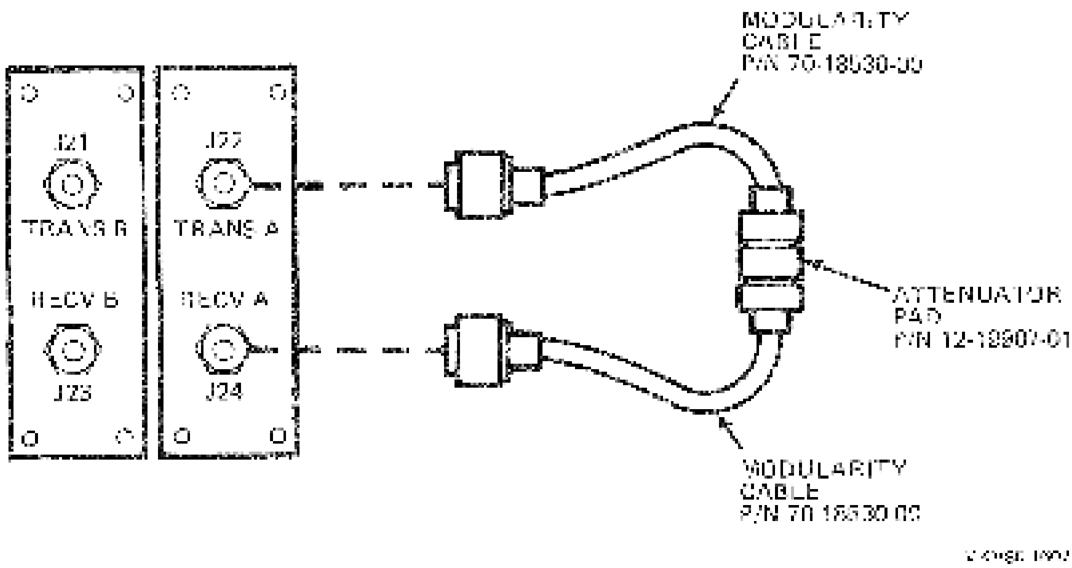


Figure 3-3 Diagnostic Loopback Cable Connections

NOTE
For more information on CI bus termination, refer to Appendix A.

3.3.2.2 Loading the Diagnostic Supervisor Program –

Procedure:

1. Insert the RX50 diskette containing file EZSAA.EXE into the PC380 console system's RX50 disk drive unit 0.

2. Load the diagnostic supervisor program resident on the PC380 console system disk into physical memory by entering the following CCL command. Use the PC380 console system keyboard as follows:

```
>>> DIAG30
%%
DIAGNOSTIC SUPERVISOR . . . ;DS>
```

3. Attach the CPU to its memory and NBI adapters, as follows:

```
DS> ATTACH KA88A HUB KA0 YES
DS> ATTACH KA88A HUB KA1 NO
DS> ATTACH MS88A HUB MS0
DS> ATTACH NB1A HUB NBIA0 0
DS> ATTACH NB1B NBIA0 NBIB0 0 0
DS> ATTACH NB1B NBIA0 NBIB1 1 0
DS> ATTACH NB1A HUB NBIA1 1
DS> ATTACH NB1B NBIA1 NBIB0 0 0
DS> ATTACH NB1B NBIA1 NBIB1 1 0
```

4. Identify the CIBCI adapter and its node configuration parameters to the diagnostic supervisor program, as follows:

```
DS> ATTACH CIBCI NBIB0 PAA0 6 4 0
```

5. Select the CIBCI adapter as the unit under test, as follows:

```
DS> SELECT PAA0
```

6. Show the unit selected, as follows:

```
DS> SHOW SELECT
```

3.3.2.3 **Repair Level Testing** – Use the following procedure to load and run the repair level diagnostics in sequence from 1 to 6. A minimum of five (5) successful passes of each diagnostic program must be completed to satisfy acceptance testing requirements. Examples 3-9 through 3-14 provide trace printouts for diagnostics EVCKA through EVCKF, respectively.

NOTE

HELP files are available under the diagnostic supervisor for all of the diagnostic programs.

Procedure:

1. Remove the RX50 diskette from the PC380 console system's RX50 disk drive unit 0.

2. Insert the RX50 diskette containing files EVCKA.EXE through EVCKF.EXE into the PC380 console system's RX50 disk drive unit 0.

3. Load the EVCKA diagnostic program, as follows:

   DS> LOAD EVCKA (first repair level diagnostic)

```
DS> LOAD EVCKA
DS> SET FLAGS TRACE, HALT
DS> SET EVENT FLAG 4
DS> START/PASS:5

..Program: SIBCI - EVCKA Repair level, revision 1.0, 28 tests,
  at 00:24:40.75.
  Testing: _PAA0

SET EVENT FLAG 4 FOR REV LEVEL OF SIIC IN TEST 3

  Test  1:   ERROR INTERRUPT CONTROL TEST
  Test  2:   DEVICE TYPE REGISTER TEST
  Test  3:   BC AND SIBCI SELF TEST
  REVISION LEVEL OF SIIC CHIP ON SIBCI IS: 0

  Test  4:   CNFGR - L WRITE ACCESS TEST
  Test  5:   CNFGR - L READ ACCESS TEST
  Test  6:   R/W TEST OF DIAG BIT IN CNFGR
  Test  7:   CNFGR - L READ ACCESS TEST - AFTER DISABLING UCSREN IN BCICR
  Test  8:   CNFGR - L READ ACCESS TEST - AFTER DISABLING STS IN BCICR
  Test  9:   PORT DATA REGISTER - R/W TEST - SOURCE IS B1
  Test 10:   R/W TEST OF BUFFERED COMMAND ADDRESS REGISTER (BCAR)
  Test 11:   R/W TEST OF BCMR
  Test 12:   R/W TEST OF DMA REGISTER
  Test 13:   RECEIVED COMMAND DATA PATH TEST
  Test 14:   R/W TEST OF CNFGR,BCAR AND BCMR TAKEN ALTOGETHER
  Test 15:   SIZE OF TRANSFER TEST
  Test 16:   DMA FILE - R/W COUNTER TEST
  Test 17:   DMA FILE - COUNTER SEQUENCE TEST
  Test 18:   R/W TEST OF BCAR AND BCMR USING THE MASTER SEQUENCER
  Test 19:   HICA ADDRESS REGISTER TEST
  Test 20:   STOP TEST
  Test 21:   PORT DATA REGISTER - CIPA DATA PATH TEST
  Test 22:   WITH DIAG BIT CLEAR, R/W TEST OF BCAR
  Test 23:   WITH DIAG BIT CLEAR, R/W TEST OF DMA REGISTER
  Test 24:   CIPAPD REGISTER READ TEST (CIPA BUX READ TEST)
  Test 25:   L READ ACCESS TEST OF LS AFTER DISABLING UCSREN IN SCI CONTROL REG
  Test 26:   NUACK TEST FOR NODE ADDRESS 200
  Test 27:   L READ ACCESS TEST OF LS AFTER DISABLING STS IN THE BICSR
  Test 28:   USER INTERRUPT CONTROL TEST

..End of run, 0 errors detected, pass count is 1,
  time is 15-JUL-1985 00:24:52.74
```

Example 3-9 Trace Printout for Repair Diagnostic EVCKA

4. Set the desired diagnostic supervisor control flags to enable printing of the number and title of each test before it is executed and to halt on a detected error, as follows:

```
DS> SET FLAGS TRACE, HALT
DS> SET EVENT FLAG 4
```

5. Start the diagnostic program, as follows

```
DS> START/PASS:5
```

6. Repeat steps 3 through 5 to load and execute the remaining repair level diagnostics (EVCKB through EVCKF).

```
DS> LOAD EVCKB
DS> SET FLAGS TRACE, HALT
DS> START/PASS:5

..Program: CIDC? - EVCKB Repair level, revision 1.0, 27 tests,
  at 00:25:58.39.
  Testing: _PAA0

  Test  1:    BUSIB/IB IN DATA PATHS TEST
  Test  2:    PMCSR ADDRESS TEST
  Test  3:    PMCSR - BIT READ/WRITE TEST
  Test  4:    INITIALIZE TEST
  Test  5:    MADR/BUS MD DATA PATHS TEST
  Test  6:    LOCAL STORE DUAL ADDRESS TEST
  Test  7:    LOCAL STORE READ/WRITE RAM TEST
  Test  8:    LOCAL STORE DYNAMIC MEMORY TEST
  Test  9:    INTERLOCKED READ/WRITE TEST
  Test 10:    VCDT - READ/WRITE RAM TEST
  Test 11:    VCDT DUAL ADDRESS TEST
  Test 12:    VCDT DYNAMIC MEMORY TEST
  Test 13:    CONTROL STORE - DUAL ADDRESS TEST
  Test 14:    CONTROL STORE - READ/WRITE RAM TEST
  Test 15:    CONTROL STORE RAM DYNAMIC MEMORY TEST
  Test 16:    CONTROL STORE ROM INSERTION TEST
  Test 17:    REGISTER DUAL ADDRESS TEST
  Test 18:    BUSIB SOURCE=LIT DEST=LS[LIT]
  Test 19:    BUSIA SOURCE EQUALS AL
  Test 20:    BUSIB DESTINATION IS VCDT[LIT]
  Test 21:    BUSIA SOURCE EQUALS LS[LIT]
  Test 22:    BUSIB SOURCE EQUALS VCDT[LIT]
  Test 23:    BUSIB DESTINATION EQUALS LS[INDEX]
  Test 24:    INDEX REGISTER SA0/SA1 CHECK
  Test 25:    BUSIB SOURCE LS[INDEX]
  Test 26:    BUSIB DESTINATION EQUALS LS[XLATE]
  Test 27:    BUSIB SOURCE EQUALS LS[XLATE]
..End of run, 0 errors detected, pass count is 1,
  time is 15-JUL-1985 00:29:37.82
DS>
```

Example 3-10 Trace Printout for Repair Diagnostic EVCKB

```
DS> LOAD EVCKC
DS> SET FLAGS TRACE, HALT
DS> START/PASS:5

..Program: DIBCI - EVCKC Repair level, revision 1.0, 33 tests,
  at 00:30:00.56.
  Testing: _PAA0

  Test  1:    2911 SEQUENCER JUMP TEST
  Test  2:    CONTROL STORE PARITY ERROR TEST
  Test  3:    "2901" RAM DUAL ADDRESS TEST
  Test  4:    "2901" RAM/Q STUCK BIT TEST
  Test  5:    "2901" RAM/Q REGISTER SHIFT
  Test  6:    "2901" ALU FUNCTION TEST
  Test  7:    "2901" CONDITION CODE Z BRANCH TEST.
  Test  8:    "2901" CONDITION CODE N BRANCH TEST.
  Test  9:    "2901" CONDITION CODE V BRANCH TEST.
  Test 10:    "2901" CONDITION CODE C BRANCH TEST.
  Test 11:    2911 SEQUENCER UPC+1 TEST
  Test 12:    2911 SEQUENCER JSR TEST
  Test 13:    POP!! MICROSTCK
  Test 14:    BUS [B<00> BRANCH TEST
  Test 15:    BUS [B<08> BRANCH TEST
  Test 16:    BUS [B<12> BRANCH TEST
  Test 17:    BUS [B<15> BRANCH TEST
  Test 18:    BUS [B<20> BRANCH TEST
  Test 19:    BUS [B<21> BRANCH TEST
  Test 20:    BUS [B<24> BRANCH TEST
  Test 21:    BUS [B<31> BRANCH TEST
  Test 22:    BUS [B<10> <09> BRANCH TEST
  Test 23:    BUS [B<14> <13> BRANCH TEST
  Test 24:    BUS [B<26> <22> BRANCH TEST
  Test 25:    BUS [B<26> <25> BRANCH TEST
  Test 26:    BUS [B<19> <18> <17> <16> BRANCH TEST
  Test 27:    MAINTENANCE TIMER DISABLE BRANCH TEST
  Test 28:    TICK BRANCH TEST
  Test 29:    REGISTER WRITTEN BRANCH T1
  Test 30:    REGISTER WRITTEN BRANCH T2
  Test 31:    XBOR - PORT INITIATED WRITE TEST
  Test 32:    BICA CNMD ADDR REG - PORT INITIATED WRITE TEST
  Test 33:    BYTE MASK - PORT INITIATED WRITE TEST
..End of run, 0 errors detected, pass count is 1,
  time is 15-JUL-1985 00:32:56.24
DS>
```

Example 3-11 Trace Printout for Repair Diagnostic EVCKC

```
DS> LOAD EVCKD
DS> SET FLAGS TRACE, HALT
DS> START/PASS:5

..Program: CIBCI - EVCKD Repair level, revision 1.0, 21 tests,
  at 00:33:15 20.
  Testing: _PAA0

  Test  1:     EXTERNAL BUS LONGWORD WRITE TO MEMORY TEST
  Test  2:     LOCAL STORE PARITY ERROR TEST
  Test  3:     DYNAMIC LOCAL STORE MOVING INVERSIONS
  Test  4:     DYNAMIC VCDT MOVING INVERSIONS
  Test  5:     EXTERNAL BUS LONGWORD READ TO MEMORY TEST
  Test  6:     EXTERNAL BUS INTERLOCK READ TO MEMORY TEST
  Test  7:     EXTERNAL BUS INTERLOCK WRITE TO MEMORY TEST
  Test  8:     EXTERNAL BUS LONGWORD WRITE TO NXM TEST
  Test  9:     CORRECTABLE READ DATA TEST FOR VAX-11/750
  TEST IGNORED FOR THIS PROCESSOR

  Test  10:    READ DATA SUBSTITUTE TEST FOR VAX-11/750
  TEST IGNORED FOR THIS PROCESSOR

  Test  11:    READ DATA SUBSTITUTE TEST FOR VAX 8700
  Test  12:    CORRECTABLE READ DATA TEST FOR VAX 8200
  TEST IGNORED FOR THIS PROCESSOR

  Test  11:    READ DATA SUBSTITUTE TEST FOR VAX 8200
  TEST IGNORED FOR THIS PROCESSOR

  Test  14:    EXTERNAL BUS EXTENDED WRITES TEST
  Test  15:    EXTERNAL BUS EXTENDED READS TEST
  Test  16:    EXTERNAL BUS MASK REGISTER TEST
  Test  17:    INTERRUPT TEST
  Test  18:    NTE DURING INTERRUPT TEST
  Test  19:    CIPA BUS PARITY ERROR (CBPE) TEST
  Test  20:    SUSPEND AND EXECUTE TEST
  Test  21:    PACKET BUFFER OUT/IN REG LOOPBACK TEST
..End of run, 0 errors detected, pass count is 1,
  time is 15-JUL-1985 00:34:43.62
DS>
```

Example 3-12 Trace Printout for Repair Diagnostic EVCKD

```
DS> LOAD EVCKE
DS> SET FLAGS TRACE, HALT
DS> START/PASS:5

..Program: CIBCI - EVCKE Repair level, revision 1.0, 13 tests,
  at 00:34:59.89.
  Testing: _PAA0

  Test  1:    PACKET BUFFER SELECT TEST
  Test  2:    OUTPUT PARITY ERROR TEST GENERATED BY PBIR
  Test  3:    TRANSMIT BUFFER "A" PATH/ADDR CHECK
  Test  4:    TRANSMIT BUFFER "B" PATH/ADDR CHECK
  Test  5:    RECEIVE BUFFER "A" PATH/ADDR CHECK
  Test  6:    RECEIVE BUFFER "B" PATH/ADDR CHECK
  Test  7:    TRANSMIT BUFFER "A" SA1/SA0
  Test  8:    TRANSMIT BUFFER "B" SA1/SA0
  Test  9:    RECEIVE BUFFER "A" SA1/SA0
  Test  10:   RECEIVE BUFFER "B" SA1/SA0
  Test  11:   FORCE RECEIVE BUFFER PARITY ERROR
  Test  12:   RECEIVE BUFFER "A" OVERFLOW TEST
  Test  13:   RECEIVE BUFFER "B" OVERFLOW TEST
..End of run, 0 errors detected, pass count is 1,
  time is 15-JUL-1985 00:36:29.06
DS>
```

Example 3-13 Trace Printout for Repair Diagnostic EVCKE

```
DS> LOAD EVCKF
DS> SET FLAGS TRACE, HALT
DS> START/PASS:5

..Program: CIBCI - EVCKF Repair level, revision 1.0, 14 tests,
  at 00:39:31.19.
  Testing: _PAA0

  Test  1:    INTERNAL MAINTENANCE LOOP TEST
  Test  2:    INTERNAL MT LOOPBACK WHILE LOADING XMIT BUFFER TEST
  Test  3:    INTERNAL MT LOOP TEST WITH ONE RCV BUF AVAILABLE
  Test  4:    INTERNAL MT LOOP TEST WITH NO REV BUF'S AVAILABLE
  Test  5:    INTERNAL MAINT LP WITH SWAP NODE ADDRESS
  Test  6:    TRANSMIT BUFFER PARITY ERROR TEST
  Test  7:    ALTERNATING PACKET BUFFER UNLOAD TEST
  Test  8:    ARBITRATION TEST N+I+1
  Test  9:    EXTERNAL MAINT. LOOP PATH "A"
  Test  10:   EXTERNAL MAINT. LOOP PATH "B"
  Test  11:   EXT. MAINT. LOOP "RECEIVERS DISABLED"
  Test  12:   EXT. MAINT. LOOP "ABORTNG TRANSMISSION"
  Test  13:   "ACKNOWLEDGE TIMEOUT" TEST
  Test  14:   EXTERNAL BUS LONGWORD WRITE TO ITSELF (LOCAL STORE)
..End of run, 0 errors detected, pass count is 1,
  time is 15-JUL-1985 00:40:26.82
DS>
```

Example 3-14 Trace Printout for Repair Diagnostic EVCKF

3.3.2.4 CI Bus Cable Testing – After successfully completing five (5) passes of each of the six repair level diagnostics, remove the attenuator pads and modularity cables from the CI bulkhead connector panels (J21-J24) and perform the following steps.

Procedure:

1. Verify that this CIBCI port has a unique node address within the VAXcluster before connecting any cables.

2. Locate the set of four CI bus coaxial cables (BNCIA-XX) and connect one end of each cable to the appropriate CI bulkhead connector panel.

   NOTE
   The coaxial CI bus cables may be connected or removed from the CI bulkhead connector panels without powering down either the system or the CIPA cabinet. DO NOT unroll or route the CI bus cables at this time.

3. Connect the two attenuator pads to the free ends of the coaxial CI bus cables. Be sure to connect transmit A to receive A, and transmit B to receive B.

4. Run five passes of the external loop section of diagnostic program EVCKF to test the CI bus cables, as follows:

   ```
   DS> RUN EVCKF/SECTION:EXTM_LOOP/PASS:5
   ```

3.3.2.5 Functional Level Testing – With the CI bus cables and attenuator pads providing signal loopback, load and run the CI functional diagnostics EVGAA and EVGAB. A minimum of five passes of each diagnostic must be completed to satisfy acceptance testing requirements. Examples 3-15 and 3-16 show trace printouts for diagnostics EVGAA and EVGAB, respectively.

Procedure:

1. Insert the RX50 diskette containing EVGAA.EXE and EVGAB.EXE into the PC380 console system's RX50 disk drive unit 0.

2. Load the EVGAA diagnostic program, as follows:

   `DS> LOAD EVGAA` (first functional diagnostic)

3. Set event flags 1 and 2 to reload the CI microcode and output the port queue entries. This is always required after running the repair level diagnostics. Set the flags as follows:

   ```
   DS> SET EVENT FLAGS 1, 2
   ```

4. Remove the diagnostic floppy diskette and insert the console floppy diskette that contains the CI780.BIN file.

5. Set the desired diagnostic supervisor control flags to enable printing of the number and title of each test before it is executed and to halt on a detected error. Set the flags as follows:

   ```
   DS> SET FLAGS TRACE, HALT
   ```

6. Start the EVGAA diagnostic program as follows.

```
DS> START/PASS:5
```

7. After five successful passes, remove the console floppy diskette, insert the diagnostic floppy diskette, clear event flags 1 and 2, and then load and run the EVGAB diagnostic program (second functional diagnostic) for five successful passes.

8. Disconnect power to the system by entering the POWER OFF console command language (CCL) command using the PC380 console system keyboard, as follows:

```
>>> POWER OFF
```

9. Replace the rear door on the CIPA cabinet and check that the door makes positive contact with each RFI gasket spring on ALL edges.

10. Disconnect the attenuator pads from the ends of the CI bus cables in preparation for routing and connecting the cables to the star coupler.

NOTE

For information on connecting the coaxial CI bus cables to the star coupler, refer to the *SC004 or SC008 Star Coupler User's Guide*.

```
DS> LOAD EVGAA
DS> SET EVENT FLAGS 1, 2
DS> SET FLAGS TRACE, HALT
DS> START/PASS:5

..Program: EVGAA - CI FUNCTIONAL PART 1, revision 2.5, 17 tests,
  at 00:48:21.95.
  Testing: _PAA0

  EVENT FLAG 1:
  DIAGNOSTIC WILL LOAD CI RAM UCODE
  FROM THE DEFAULT LOAD PATH.

  EVENT FLAG 2.
  OUTPUT THE PORT QUEUE ENTRIES.

  EVENT FLAG 3:
  INVOKES THE REQUEST ID LOOP FUNCTION.

  ROM REVISION = 3          WCS REVISION = 4

  Test  1:    CLUSTER CONFIGURATION

                       CLUSTER CONFIGURATION -- PATH A
                       *******************************

  NOTE:
  YOU CANNOT DIFFERENTIATE BETWEEN A CI780 AND CI750 REMOTELY.

  NODE #   DEVICE TYPE    ROM/WCS REV.    PORT FUNCTIONALITY    PATH TYPE
  ------   -----------    ------------    ------------------    ---------
    0         CI7X0          3    4          FFFFFF00(X)        DUAL PATH

                       CLUSTER CONFIGURATION -- PATH B
                       *******************************

  NODE #   DEVICE TYPE    ROM/WCS REV.    PORT FUNCTIONALITY    PATH TYPE
  ------   -----------    ------------    ------------------    ---------
    0         CI7X0          3    4          FFFFFF00(X)        DUAL PATH

  Test  2:    SETCKT TEST WITH VARIOUS MASKS AND M_VALUES
  Test  3:    SETCKT TEST FOR EACH VALID PORT
  Test  4:    SETCKT TEST FOR NVALID PORT
  Test  5:    REQID TEST
  Test  6:    REQID TEST WITH 6 PACKETS ON DCFQ
  Test  7:    DATAGRAM DISCARD TEST
  Test  8:    RESPONSE QUEUE AVAILABLE INTERRUPT TEST
  Test  9:    SEND DATAGRAM -SNDDG- TEST
  Test  10:   SNDMSG TEST WITH NOVIRTUAL CIRCUIT TEST
  Test  11:   SEND MESSAGE TEST, CROSSING PAGE BOUNDARY
  Test  12:   MESSAGE LENGTH TEST
  Test  13:   PACKET SIZE VIOLATION TEST
  Test  14:   SEND LOOPBACK -SNDLB- TEST
  Test  15:   SNDLB TEST, FULL BUFFER PATH A
  Test  16:   SNDLB TEST, FULL BUFFER PATH B
  Test  17:   SNDLB TEST, BOTH PATHS
..End of run, 0 errors detected, pass count is 1,
  time is 15-JUL-1985 00:50:40.36
DS>
```

Example 3-15   Trace Printout for Functional Diagnostic EVGAA

```
DS> LOAD EVGAB
DS> CLEAR EVENT FLAG 1, 2
DS> SET FLAGS TRACE, HALT
DS> START/PASS:5

..Program: EVGAB - CI FUNCTIONAL PART II, revision 2.5, 12 tests,
  at 00:50:54.31.
  Testing: _PAA0

  ROM REVISION = 3        WCS REVISION = 4

  Test  1:    SEND DATA TEST, WITH OFFSET COMBINATIONS
  Test  2:    REQUEST DATA TEST, WITH OFFSET COMBINATIONS
  Test  3:    INVALIDATE TRANSLATION CACHE TEST
  Test  4:    SNDMDAT TEST, ENABLED/MAINTENANCE STATE
  Test  5:    SNDMDAT TEST, ENABLED STATE
  Test  6:    REQMDAT TEST, ENABLED/MAINT STATE
  Test  7:    REQMDAT TEST, ENABLED STATE
  Test  8:    SEND RESET TEST IN ENABLED STATE
  Test  9:    QUEUE CONTENTION TEST
  Test 10:    BUFFER READ ACCESS TEST
  Test 11:    BUFFER WRITE ACCESS TEST
  Test 12:    WRITE TO GLOBAL BUFFER TEST
..End of run, 0 errors detected, pass count is 1,
  time is 15-JUL-1985 00:52:29.92
DS>
```

Example 3-16   Trace Printout for Functional Diagnostic EVGAB

3.4 MAINTENANCE VERIFICATION

A number of software system tools are required in order to facilitate VAXcluster maintenance. These software tools allow isolation of a potential failure to an individual node or option within a VAXcluster system. Refer to Table 3-3.

Table 3-3 Summary of the Functions of VAXcluster System Maintenance and Management Tools

Tool	Function
CI Exerciser	A Level 2R multipurpose exerciser that provides local CI interface functional tests as well as a means to determine the ability of VAXcluster nodes to reliably communicate using the CI bus.
VAXsim Utility	A VAX system integrity monitor utility program monitors and filters errors as they are logged by the VMS operating system. It provides the user with a warning mechanism that quickly identifies an option that is either failed or has degraded operationally.[1]
Show Cluster Utility	Allows the display of a large variety of information relevant to the configuration and operation of the VAXcluster of which the host system is a member.[2]
Set Host/HSC	Allows a user on a host VMS system to effectively become an HSC50 terminal. The user may then issue any standard HSC50 commands and view or control the HSC50 from a terminal connected directly to one of the HSC50 terminal ports.[3]

[1] For more information, consult the *VAX System Integrity Monitor Manual*.
[2] For more information, consult the *VAX/VMS SHOW Cluster Utility Manual*.
[3] For more information, consult the *VAX/VMS DCL Dictionary* under Set Host/HSC.

CHAPTER 4
REGISTER SUMMARY

This section presents the interface conventions which allow programmer access to the CIBCI adapter functions. Access to the CIBCI adapter functions is gained via addressable hardware and software registers that are used to control and monitor the operation within the CIBCI adapter itself. Entry to these registers is accomplished through the VAXBI address space area. The addressable hardware and software registers and their bit map format are discussed in Section 4.3.

4.1 VAXBI ADDRESS SPACE

The physical address on the VAXBI is 30 bits long, thereby providing a VAXBI physical address space of one gigabyte. A programmer accesses this physical address space whenever making reference to a CIBCI adapter's hardware or software register.

The VAXBI physical address space consists of two parts: memory space and I/O space. Selection of memory space and I/O space is determined by address bit 29 of a read or write VAXBI bus transaction. The first 512 megabytes (addresses 0000 0000 through 1FFF FFFF hexadecimal) are physical memory space addresses. The last 512 megabytes of the VAXBI physical address space (addresses 2000 0000 through 3FFF FFFF hexadecimal) are I/O space addresses. Figure 4-1 illustrates the physical partitioning of the VAXBI physical address space.

4.1.1 VAXBI I/O Address Space

As shown in Figure 4-2, the 512 megabyte VAXBI I/O address space is organized into several categories: map window, multi broadcast space, and node space. Only the VAXBI node space is used by the CIBCI adapter.

The VAXBI node space (Figure 4-3) is organized into sixteen 8 kilobyte address blocks. The CIBCI adapter hardware is assigned to one of these address blocks. This address block is referred to as the CIBCI adapter node. It is accessed whenever a CIBCI adapter's hardware or software register is referenced.

4.2 CIBCI ADAPTER NODE

4.2.1 Addressing

The address area of the CIBCI adapter node is calculated by taking a base address representing the VAXBI I/O address space (2000 0000 hexadecimal) and adding 8K times the node ID, plus the offset address of the device register. For simplicity, this calculated address is represented by bb+ whenever a reference is made to any of the forthcoming register bit maps. Figure 4-4 illustrates the formal structure of a 30-bit I/O address. Table 4-1 lists the starting addresses of the 16 VAXBI node spaces.

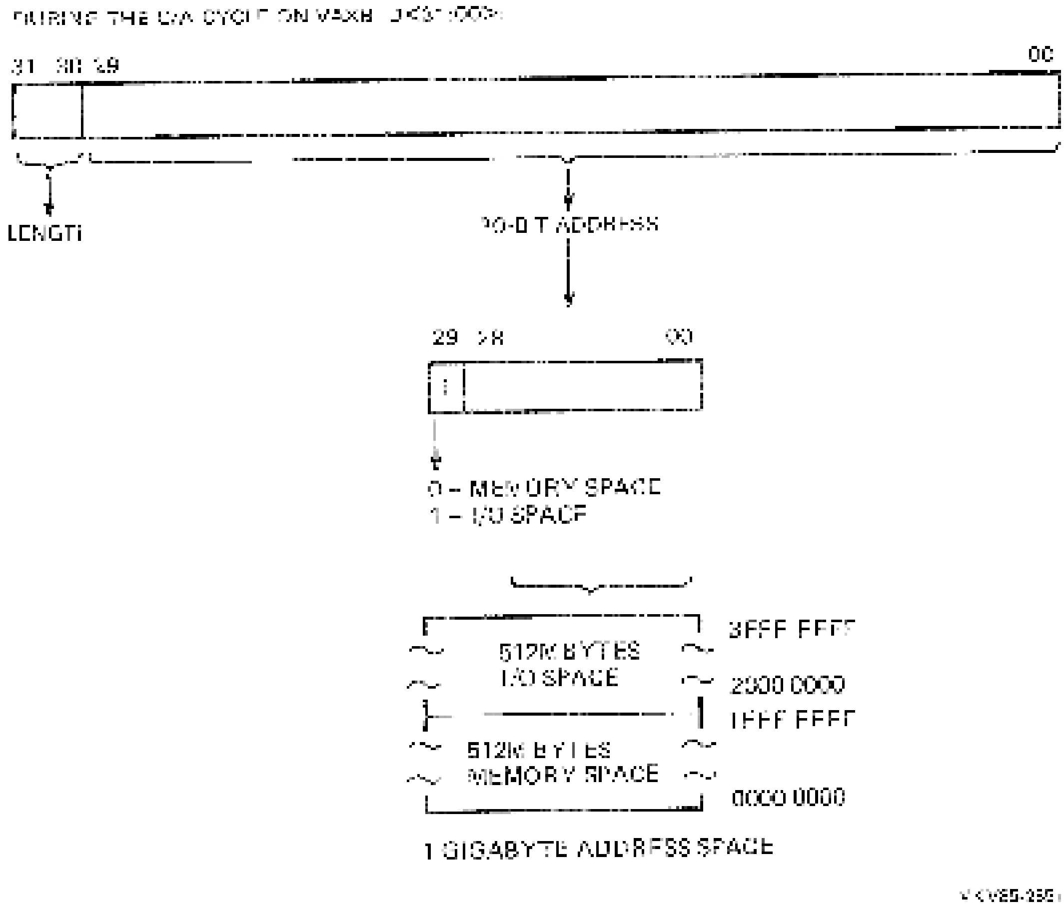


Figure 4-1 VAXBI Physical Address Space

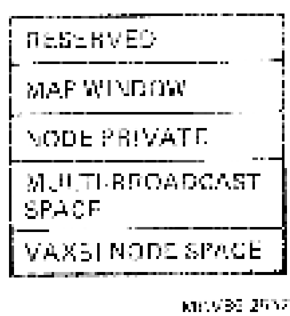


Figure 4-2 VAXBI Physical I/O Address Space

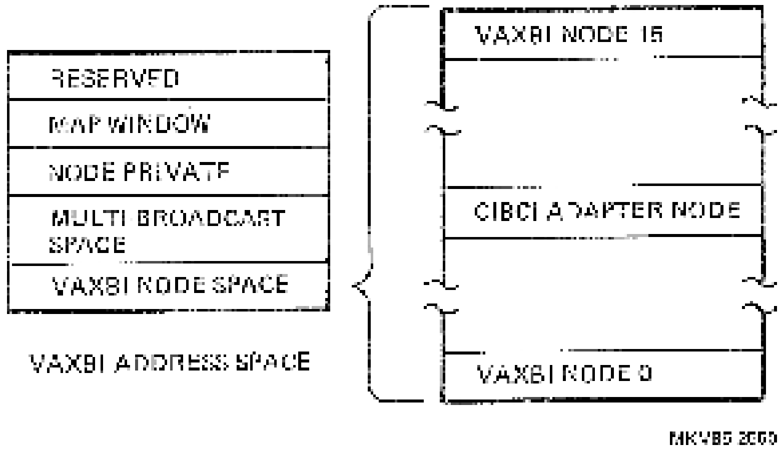


Figure 4-3 VAXBI Node Space

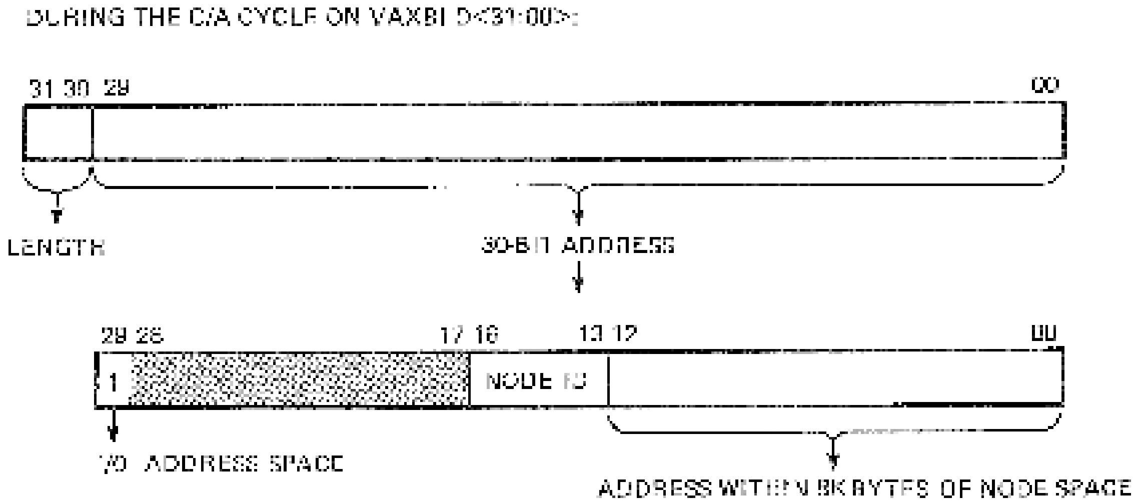


Figure 4-4 30-Bit I/O Address Bit Map

Table [illegible] Node Space Address Assignments

Node ID	BB Address
0	2000 0000
1	2000 2000
2	2000 4000
3	2000 6000
4	2000 8000
5	2000 A000
6	2000 C000
7	2000 E000
8	2001 0000
9	2001 2000
A	2001 4000
B	2001 6000
C	2001 8000
D	2001 A000
E	2001 C000
F	2001 E000

NOTE
Offset address range 0000 to 1FFF hexadecimal.

4.2.2 Partitioning

As shown in Figure 4-5, the CIBCI node space is divided into two segments: VAXBI CSR space and user CSR space. The VAXBI CSR space occupies the first 256 byte locations and is used by the VAXBI protocol and VAXBI control logic of the CIBCI adapter hardware. The user CSR space occupies the remaining locations of the address block. Only a portion of these addresses are used by the CIBCI adapter. Reading or writing to an unused register address produces unpredictable results.

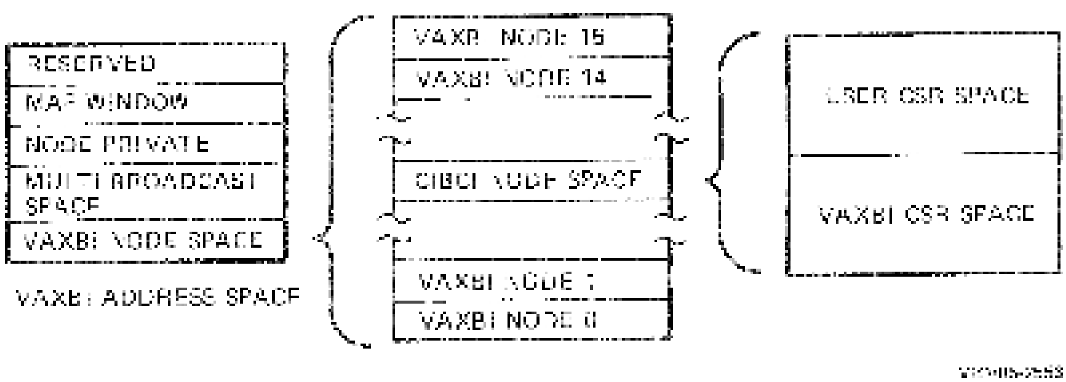


Figure 4-5 CIBCI Address Node Space

4.2.3 Registers

As shown in Figure 4-6, the first 256 bytes of the CIBCI node space are reserved for the VAXBI CSR registers. VAXBI required registers and specific device registers fall into the category of VAXBI CSR registers. The VAXBI required registers are used by all VAXBI nodes including the CI. The specific device registers are special purpose VAXBI registers used to control the VAXBI device window area, and VAXBI data transfer control and interrupt control. The remaining addresses of the CIBCI node space are reserved for user CSR registers. The adapter registers fall into the category of user CSR registers and are used for initializing and controlling the CIBCI adapter hardware. All of these registers are accessed using longword addresses. (See Figure 4-7.)

NOTE

The CIBCI adapter hardware only issues longword or quadword VAXBI bus transactions. It does NOT respond to byte, word, or quadword VAXBI bus transactions.

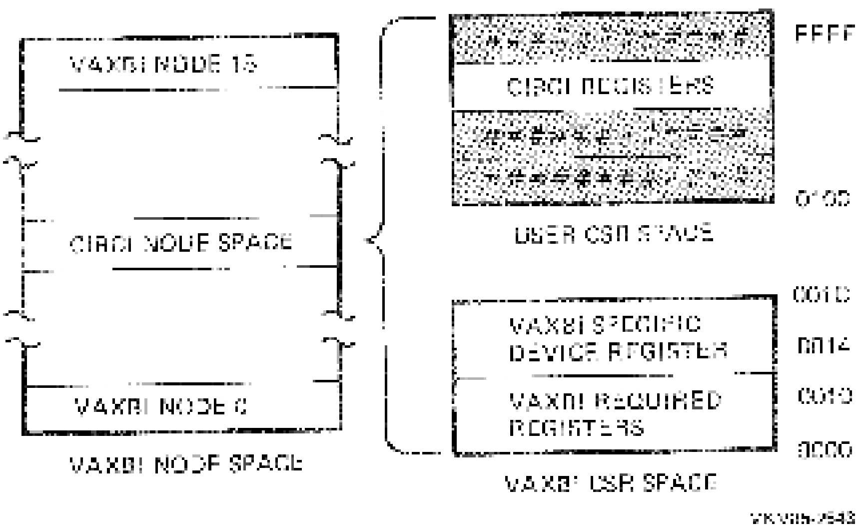


Figure 4-6 CIBCI Adapter Register Address Space

4.3 VAXBI REQUIRED REGISTERS

Figure 4-8 presents a more detailed diagram of the VAXBI required registers.

bb+00	VAXBI REQUIRED REGISTERS
bb+1C	
bb+20	BIIC SPECIFIC DEVICE REGISTERS
bb+EC	
bb+100	CONFIGURATION REGISTER
bb+110	PORT MAINTENANCE CONTROL/STATUS REGISTER
bb+114	MAINTENANCE ADDRESS REGISTER
bb+118	MAINTENANCE DATA REGISTER
bb+124	BICA ADDRESS REGISTER
bb+128	BICA COMMAND/BYTE MASK
bb+12C	DMA REGISTER FILE

MKV85-2555

Figure 4-7 VAXBI Interface Registers and Adapter Registers

bb+00	DEVICE TYPE REGISTER
bb+04	VAXBI CONTROL/STATUS REGISTER
bb+08	BUS ERROR REGISTER
bb+0C	ERROR INTERRUPT CONTROL REGISTER
bb+10	INTERRUPT DESTINATION REGISTER
bb+14	INTER-PROCESSOR INTERRUPT MASK REGISTER

MKV85-2556

Figure 4-8 VAXBI Required Registers

4.3.1 Device Type Register (DTR)

Address Offset — 00 Hexadecimal

The device type register, field bits <15:00>, is used to identify the type of node for use by the VMS operating system's device driver software. The device type assigned to the CIBCI adapter is 10B (hexadecimal) (See Figure 4-9.)

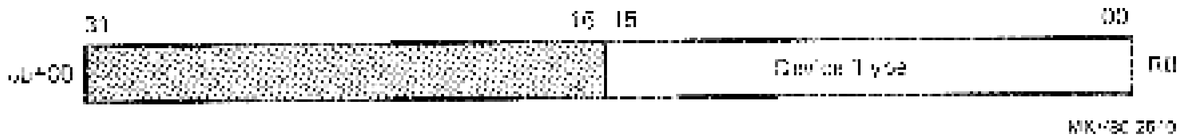


Figure 4-9 Device Type Register Bit Map

4.3.2 VAXBI Control and Status Register (BICSR)

Address Offset = 04 Hexadecimal

The VAXBI control and status register contains control and status information. It also contains the BIIC type and the node ID, and specifies the mode of arbitration. Figure 4-10 illustrates the register format. The bit assignments are described in Table 4-2.

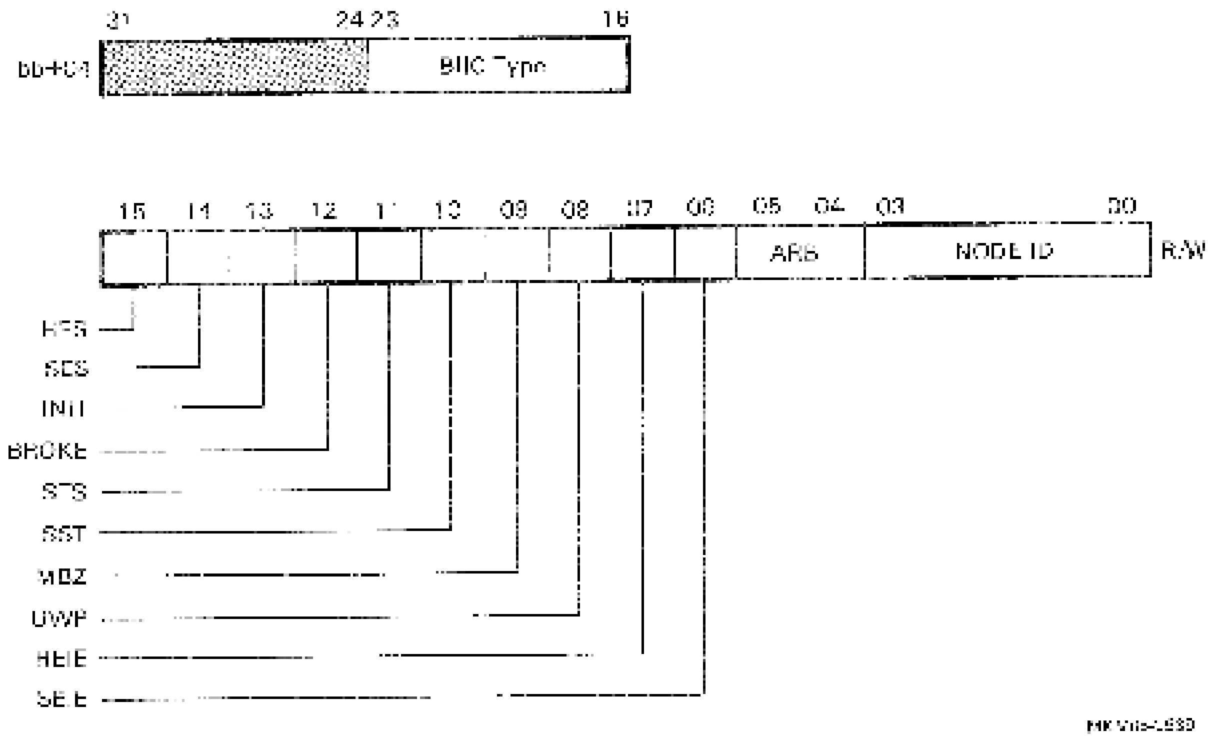


Figure 4-10 VAXBI Control and Status Register Bit Map

Table 4-2 VAXBI Control and Status Register Bits

Bit	Name	Mnemonic	Comments/Function
<23:16>	BIIC type	ITYPE	Indicates the type of BIIC chip used. For example, it should read 0000 0011 (binary) for the fourth implementation.
<15>	Hard error summary	HES	Indicates that one or more of the hard error bits in the bus error register are set.
<14>	Soft error summary	SES	Indicates that one or more of the soft error bits in the bus error register are set.
<13>	Initialize	INIT	The CIBCI has not yet completed its initialization process.
<12>	Broke	BROKE	Indicates that the BIIC chip has not successfully completed its internal self-test routines.
<11>	Self-test status	STS	Indicates the BIIC chip has successfully passed its internal self-test routines.
<10>	Start self-test	NRST	Instructs the BIIC chip to begin its internal self-test routines.
<8>	Unlock write pending	UWP	A successful read lock transaction has been completed and there has not yet been a subsequent write unlock command.
<7>	Hard error interrupt enable	HEIE	Enables an interrupt on error to be generated when a hard error condition (nonrecoverable error) is detected.
<6>	Soft error interrupt enable	SEIE	Enables an interrupt on error to be generated when a soft error condition (recoverable error) is detected.
<5:4>	Arbitration control	ARB	Specifies the mode of arbitration to be used by the CIBCI adapter. VMS sets ARB to zero. ARB Description 0 Round-robin arbitration 1 Fixed-high arbitrate priority 2 Fixed-low arbitrate priority 3 Disable arbitration
<3:0>	Node ID	NODE ID	Specifies the node ID as read from the VAXBI ID plug located on the VAXBI backplane.

4.3.3 Bus Error Register (BER)

Address Offset = 08 Hexadecimal

NOTE

VMS clears all errors by writing a logical 1 to those bits which contain a logical 1.

The bus error register provides bus error status information resulting from VAXBI bus or internal loopback transactions. Figure 4-11 illustrates the register format. The bit assignments are described in Table 4-3.

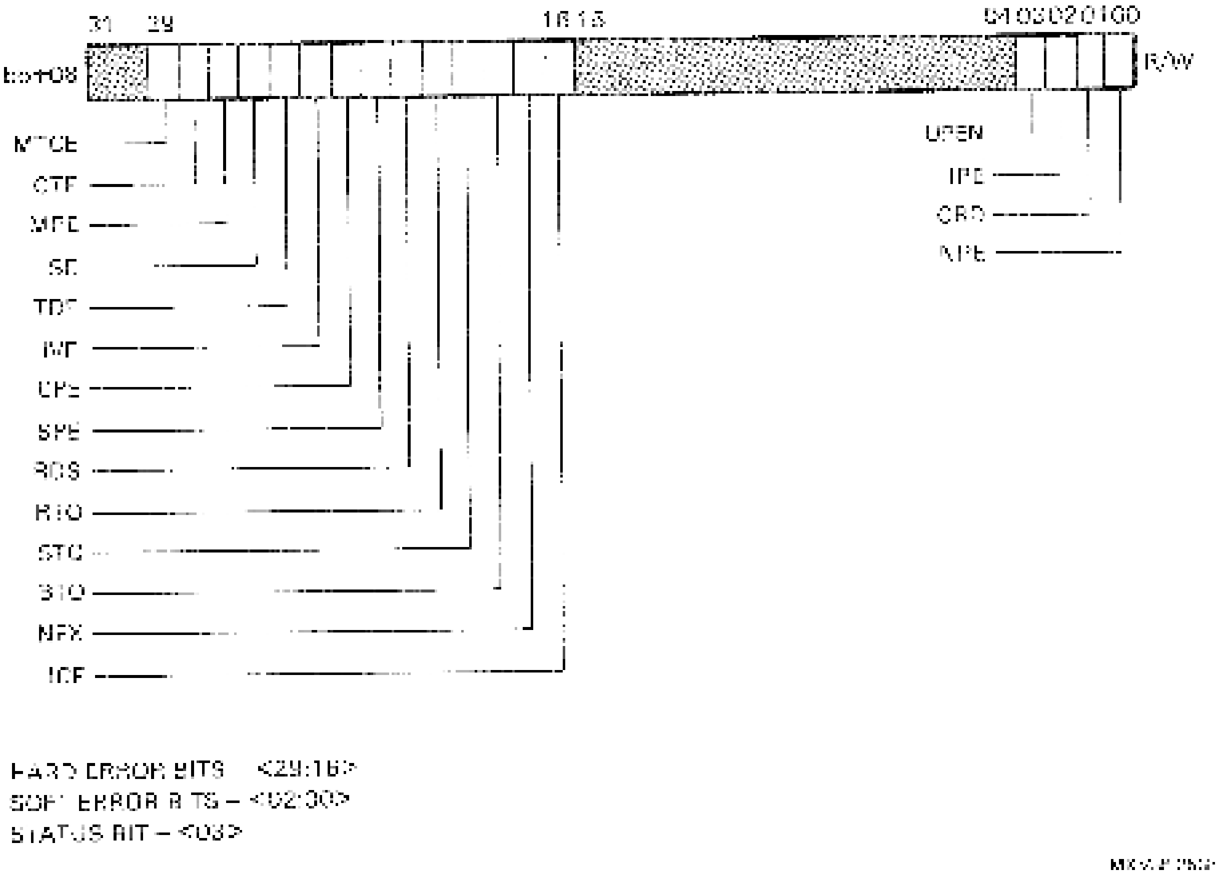


Figure 4-11 Bus Error Register Bit Map

Table 4-3 Bus Error Register Bits

Bit	Name	Mnemonic	Command/Function
<29>	Master transmit check error	MTCE	The BIIC data received from the VAXBI does not match the transmitted data of the VAXBI master.
<28>	Control transmit error	CTE	The CIBCI detected a deasserted state on the VAXBI NO ARB L, VAXBI BSY L, or VAXBI CNF <2:0> L signals in a cycle in which it was attempting to assert the signal.
<27>	Master parity error	MPE	The VAXBI master detected a parity error on the VAXBI bus during an ACK confirmed data transaction cycle (read-type or vector).
<26>	Interlock sequence error	ISE	The BIIC chip successfully executed a write unlock command (UWMCI) when no corresponding read command (IRCI) transaction had been previously issued.
<25>	Transmitter during fault	TDF	A parity error was detected on the VAXBI D<31:0> and VAXBI I<3:0> signal lines during an arbitration ARB cycle which resulted in setting the SPE, MPE, CPE, or IPE bit.
<24>	Identification vector error	IVE	The CIBCI chip did not correctly receive an identification vector confirmation.
<23>	Command parity error	CPE	The BIIC chip detected a parity error in a command/address cycle of either a VAXBI or loopback request transaction.
<22>	Slave parity error	SPE	A parity error was detected on the VAXBI bus during a write-type transaction cycle.
<21>	Read data substitute	RDS	The BIIC chip detected a read data substitute or reserved status code during a read-type command or IDENT command transaction, and detected no parity error during the data transfer cycle that contains the RDS code.
<20>	Retry timeout	RTO	More than 4096 consecutive retry cycles were executed for a given transaction.
<19>	Stall timeout	STO	The stall code was asserted for greater than 127 consecutive cycles.
<18>	Bus timeout	BTO	The BIIC chip was unable to start at least one pending transaction before 4096 cycles have elapsed.

Table 4-3 Bus Error Register Bits (Cont)			
Bit	Name	Mnemonic	Comment/Function
<05>	Nonexistent address	NEX	The BIIC chip received a NO ACK response for a read-type or write-type initiated command.
<04>	Illegal confirmation error	ICE	The BIIC chip received or generated a reserved or illegal confirmation code.
<03>	User parity enable bit	UPEN	Enables the BIIC user port to generate parity for the VAXBI bus.
<02>	Identification parity error	IPE	A parity error was detected on the encoded master ID during an embedded ARB cycle.
<01>	Corrected read data error	CRD	A corrected read data status code was received for a read-type initiated command.
<00>	Null bus parity error	NPE	Odd parity was detected on the VAXBI bus during the second cycle of a two-cycle sequence in which NO ARB L and BSY L were deasserted.

4.3.4 Error Interrupt Control Register (EICR)

Address Offset – 0C Hexadecimal

The error interrupt control register controls the operation of the interrupts initiated by a BIIC detected bus error (which sets a bit in the bus error register) or by setting the force bit in this register. An error interrupt request is the logical OR of all the BER register bits with the force bit and error interrupt enable bits set in the VAXBI control and status register. This register is set up by the software if the STIB bit is set in the VAXBI control and status register.

Figure 4-12 illustrates the register format. The bit assignments are described in Table 4-4.

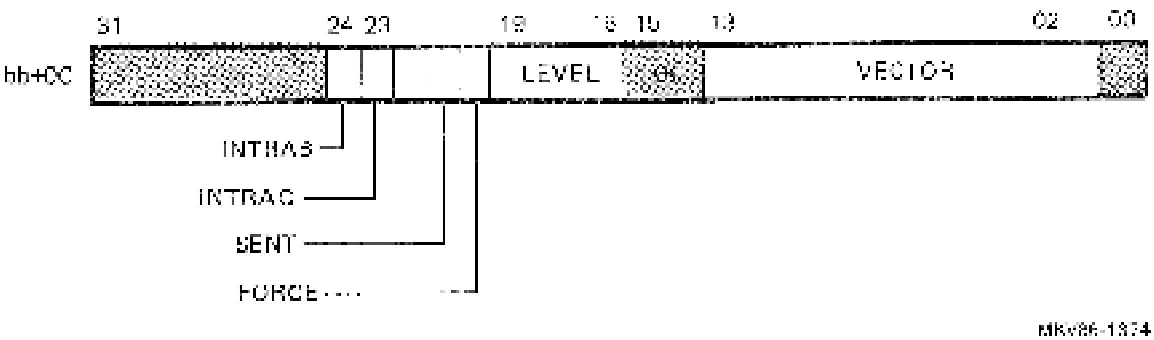


Figure 4-12 Error Interrupt Control Register Bit Map

Table 4-8 Error Interrupt Control Register Bits

Bit	Name	Mnemonic	Comment/Function
<24>	Interrupt abort	INTRAB	A NOACK or ILLEGAL confirmation was received for an initiated INTR command and subsequently the INTR command was aborted.
<25>	Interrupt complete	INTRC	The vector for a VAXBI error interrupt was successfully transmitted or an INTR command sent under the control of this register has been aborted.
<23>	Interrupt sent	INTR SENT	An INTR command has been successfully sent and an IDENT ARB cycle is expected.
<21>	Force interrupt	INTR FORCE	Forces an error interrupt request in the same manner as any bus error register bit except that the request is not qualified by the HEIE and SEIE bits.
<19:16>	Interrupt level	LEVEL	Specifies the level(s) at which INTR commands under control of this register are transmitted over the VAXBI bus. Also, the CIBCI uses this level field to determine whether it will respond to an IDENT command.
<13:2>	Vector	VECTOR	Contains the vector used during interrupt sequences. It is transmitted when the node wins the IDENT ARB cycle of an IDENT transaction that matches the conditions in the error interrupt control register.

4.3.5 Interrupt Destination Register (IDR)

Address Offset — 10 Hexadecimal

The IDR indicates which nodes on the VAXBI are to be targeted by interrupt commands. The destination is sent out during the INTR command and is monitored by all nodes to determine whether to respond. Figure 4-13 illustrates the register format.

NOTE

IDR is loaded by VMS with the decoded ID information of the system interrupt fielding node.

4.3.6 Inter-Processor Interrupt Mask Register (IPIMR)

Address Offset — 14 Hexadecimal

The IPIMR indicates which nodes are permitted to send IPINTRs to this CIBCI node. If a bit in the IPINTR mask field is a one, IPINTRs directed at this node from the corresponding node will cause selection. If the bit is a zero, IPINTRs from that node do not cause selection. Figure 4-14 illustrates the register format.

Figure 4-13 Interrupt Destination Register Bit Map

Figure 4-14 Inter-Processor Interrupt Mask Register Bit Map

NOTE

The CIBCI adapter does not use inter-processor interrupts. Therefore, VMS clears the interrupt mask field so that an inter-processor interrupt is not acknowledged.

4.4 BIIC SPECIFIC DEVICE REGISTERS

Figure 4-15 presents a more detailed diagram of the VAXBI specific device registers.

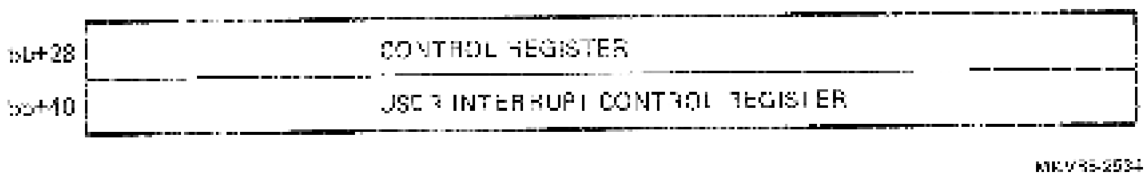


Figure 4-15 VAXBI Specific Device Registers

4.4.3 BCI Control Register (BCICR)

Address Offset = 28 Hexadecimal

The BCI control register enables various functions between the BIIC chip and the user's port to occur. Figure 4-16 illustrates the register format. The bit assignments are described in Table 4-5.

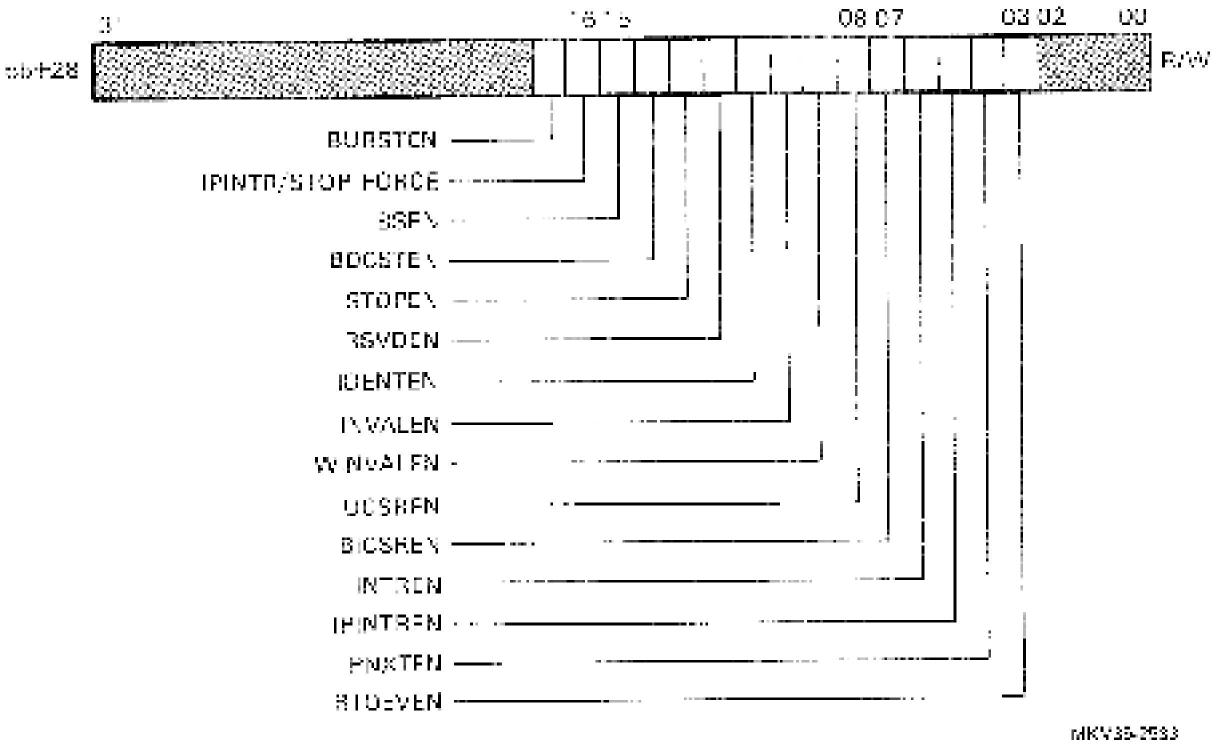


Figure 4-16 BCI Control Register Bit Map

Table 4-5 BCI Control Register Bits

Bit	Name	Mnemonic	Comment/Function
<17>	Burst enable	BURSTEN	Cleared by VMS for normal operation.
<16>	Interprocessor interrupt/stop force	IPINTR/STOP FORCE	Cleared by VMS for normal operation.
<15>	Multicast space enable	MSEN	Cleared by VMS for normal operation.
<14>	Broadcast enable	BDCSTEN	Cleared by VMS for normal operation.

Table 4-5 BCI Control Register Bits (Cont)

Bit	Name	Mnemonic	Condition/Function
<13>	Stop enable	STOPEN	Set by VMS for normal operation to allow assertion of BCI SC0 L and the appropriate SC<02:00> code following the receipt of a STOP command or loopback request directed at the BIIC node.
<12>	Reserved enable	RESVDEN	Cleared by VMS for normal operation.
<11>	Identification enable	IDENTEN	Cleared by VMS for normal operation.
<10>	Invalidate enable	INVALEN	Cleared by VMS for normal operation.
<09>	Write invalidate enable	WINVALEN	Cleared by VMS for normal operation.
<08>	User CSR space enable	UCSREN	Set by VMS for normal operation to allow assertion of BCI SEL L and the appropriate SC<2:0> code when the user CSR space is accessed from the VAXBI.
<07>	BIIC CSR space enable	BICSREN	Cleared by VMS for normal operation.
<06>	Interrupt enable	INTREN	Cleared by VMS for normal operation.
<05>	Interprocessor interrupt enable	IPINTREN	Cleared by VMS for normal operation.
<04>	Pipeline NXT enable	PNXTEN	Cleared by VMS for normal operation.
<03>	RTO EV enable	RTOEVEN	Set by VMS for normal operation to enable the output RTO code in place of the STO event code following the occurrence of a retry timeout.

4.4.2 User Interrupt Control Register (UICR)

Address Offset = 40 Hexadecimal

The UICR controls the operation of interrupts initiated by assertion of the BCI INT<07:04> L lines, or by setting any of the force bits. The operation of the BCI INT<07:04> L lines and the force bits are essentially identical. In the following descriptions, interrupt request indicates BCI INT<07:04> or force bits set. The CI adapter asserts only INT<07:04>, but diagnostics may set the force bits. Figure 4-17 illustrates the register format. The bit assignments are described in Table 4-6.

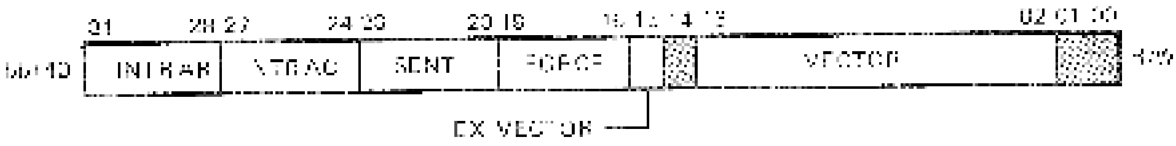


Figure 4-17 User Interrupt Control Register Bit Map

Table 4-46 User Interrupt Control Register Bits

Bit	Name	Mnemonic	Comment / Function
<31:28>	Interrupt abort	INTRAB	A status field corresponding to the four interrupt levels. An interrupt abort bit is set if an INTR command sent under control of this register is aborted because a NOACK or ILLEGAL confirmation code was received.

NOTE: There is one interrupt abort bit for each of the four interrupt levels. VMS clears these bits (if set) after servicing an error interrupt.

Bit	Name	Mnemonic	Comment / Function
<27:24>	Interrupt complete	INTRC	A status field corresponding to the four interrupt levels. An interrupt complete bit is set when the vector for an interrupt has been successfully transmitted, or if an INTR command sent under control of this register is aborted because a NOACK or ILLEGAL confirmation code was received.
<23:20>	Interrupt sent	SENT	Indicates that an INTR command for the corresponding level was successfully sent and an IDENT ARB cycle is expected.
<19:16>	Force interrupt	FORCE	Cleared by VMS for normal operation.
<15>	External vector	EX VECTOR	Cleared by VMS for normal operation.
<13:02>	Vector	VECTOR	Contains the vector during user interrupt sequences. The vector is transmitted when the CIXCD adapter wins an IDENT ARB cycle on an IDENT transaction that matches the conditions in the user interrupt control register.

NOTE: VMS sets this field to the appropriate vector for hard error and soft error interrupts.

4.5 BIIC USER PORT REGISTERS

The BIIC user port registers consist of those registers which reside on the T series modules of the CIBCI adapter option. These registers consist of:

1. One (1) configuration register
2. One (1) address register
3. One (1) byte mask/command register
4. Four (4) register file registers

Figure 4-18 shows the location of BIIC user port registers, part of the adapter registers, in the CIBCI adapter node space.

NOTE

Writing to BIIC user port registers under system operation may erase valid data for the operation in progress.

bb+120	CONFIGURATION REGISTER
bb+124	ADDRESS REGISTER
bb+128	COMMAND/BYTE MASK REGISTER
bb+12C	DIRECT MEMORY ACCESS REGISTER FILE

Figure 4-18 BIIC User Port Registers

4.5.1 BCIA Configuration Register (CNFGR)

Address Offset — 100 Hexadecimal

The CNFGR contains the port status bits, error bits, and adapter code bits for the CIBCI adapter hardware. Figure 4-19 illustrates the register format. Each register bit is described in Table 4-7.

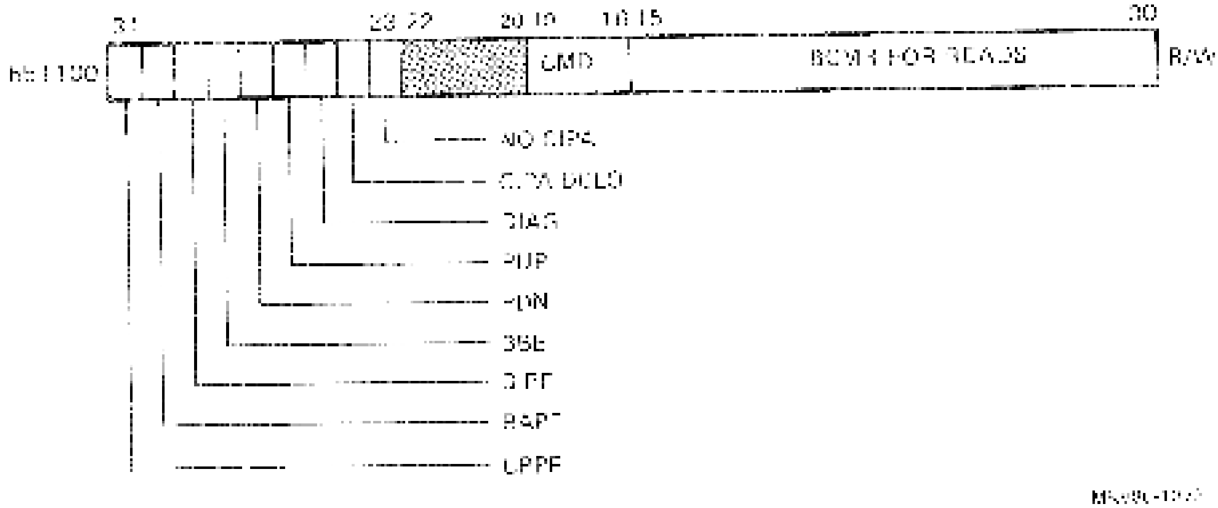


Figure 4-19 BICA Configuration Register Bit Map

Table 4-7 BCIA Configuration Register Bits

Bit	Name	Mnemonic	Comment/Function
<31>	CIPA BUS parity error	CPPE	A parity error was detected on either the CIPA bus, on the BIPA, or in the BIC chip. The interrupt port will assert CIPA BUS ERROR, which sets the MCE bit in the CIPA PSR and causes an interrupt on the VAXBI.
<30>	BIC Adapter parity error	BAPE	A data path parity error (VAXBI address or data) was detected on the BICA data path. This interrupt port will assert CIPA BUS ERROR.
<29>	VAXBI parity error	BIPE	Represents the logical OR of error events that drive CIPA BUS ERROR. This interrupt port will assert CIPA BUS ERROR.
<28>	VAXBI BSY error	BBE	The VAXBI hardware detected a bus busy event error. This interrupt port will assert CIPA BUS ERROR.
<27>	Power down	PDN	The CIBCI adapter hardware is powering down.
<26>	Power up	PUP	The CIBCI adapter hardware is powering up.

Table 4-7 SCA Configuration Register Bits (Cont)

Bit	Name	Mnemonic	Command/Function
<27>	Diagnose	DIAG	When set, enables access to the internal diagnostic mode registers which allows the BICA portion of the CIBCI to be checked out without the CIPA hardware. Cleared by VMS for normal operation.
<26>	CIPA DCLO	CIPA DCLO	The CIPA hardware is connected and powered down, or is disconnected from the BICA. It is used by the diagnostics to decide whether or not it is necessary to set DIAG in order to enter diagnostic maintenance mode.
<25>	No CIPA	NO CIPA	The CIPA hardware interface is either not connected or powered up if CIPA DCLO is cleared.
[illegible]	Maintenance Go	MAINT GO	Initiates a DMA transfer while in diagnostic maintenance mode.
<19:16>	Received command	COMMAND	When read, contains the encoded SBI command converted into a VAXBI command by the CIBCI hardware. <19:16> Command 1 READ MASK 2 WRITE MASK 4 INTLK READ MASK [illegible] INTLK WRITE MASK 8 EXTENDED READ MASK B EXTENDED WRITE MASK
<15:00>	BCMR	BCMR	When read, contains the contents of the BCMR. Writes to bits <15:00> have no effect on the CIBCI adapter hardware.

4.5.2 BICA Address Register (BCAR)

Address Offset = 124 Hexadecimal

The BCAR contains the address to be driven onto the VAXBI bus for CIPA hardware initiated VAXBI transactions. It is loaded under microcode control using the CIPA bus or under macrocode control using the VAXBI protocol. The HI and LO portions of this register are necessary because the CIPA bus is only 16 bits wide and, therefore, can load only 16 bits at a time. Figure 4-20 illustrates the register format. Each register bit is described in Table 4-8.

NOTE

VMS will not normally write into this register. Reading and writing to location bb+124 is treated differently. This difference affects diagnostics and NOT the VMS port driver software.

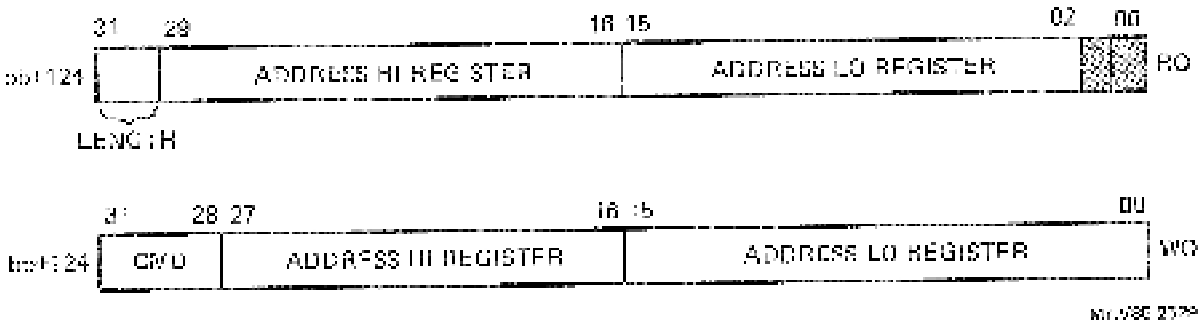


Figure 4-20 BICA Address Register Bit Map

Table 4-8 DCIA Address Register Bits

Bit	Name	Mnemonic	Contents/Function
<31:28>	Command	CMD	Specifies the encoded SBI command which will be converted into a VAXBI command for the DBCI hardware. <31:28> Command 1 READ MASK 2 WRITE MASK 4 INTLK READ MASK 7 INTLK WRITE MASK 8 EXTENDED READ MASK B EXTENDED WRITE MASK
<27:16>	Address HI	ADRS HI	Specifies the high order address bits <29:16> of the DMA transaction over the VAXBI bus.
<15:00>	Address LO	ADRS LO	Specifies the low order address bits <15:02> of the DMA transaction over the VAXBI bus. VAXBI <01:00> of a DMA transaction are always cleared.
<31:30>	Length	SIZE	Contains the length of the VAXBI bus transaction in bytes. <31:30> Length 0 Reserved 1 Longword (4 bytes) 2 Quadword (8 bytes) 3 Octaword (16 bytes)
<29:02>	Address HI/LO	ADRS HI/LO	Contains the physical memory address of the CMD/ADDR VAXBI transfer.

4.5.3 BICA Command/Byte Mask Register (BCMR)

Address Offset — 128 Hexadecimal

The byte mask portion of BCMR specifies which byte(s) of data are written during CIPA hardware initiated DMA write transactions. Figure 4-21 illustrates the register format. Each register bit is described in Table 4-9.

NOTE
Reading and writing to BCMR is treated differently. This difference affects the diagnostics and NOT the VMS port driver software.

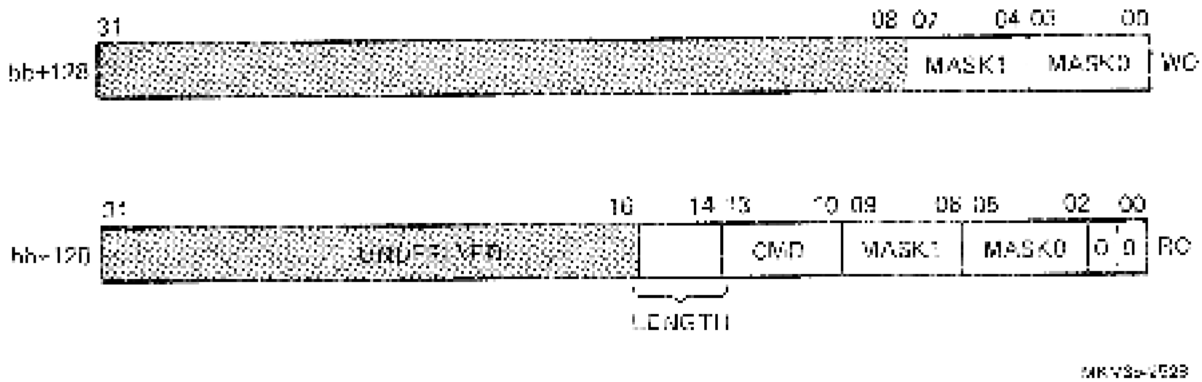


Figure 4-21 BICA Command/Byte Mask Register Bit Map

Table 4-9 BICA Command/Byte Mask Register Bits

Bit	Name	Mnemonic	Comments/Function
<07:04>	Mask 1	MASK1	Specifies which byte(s) are to be written for the second longword of a CIPA quadword DMA write transaction. These bits may be written by the CIPA hardware microcode or by the VAX macrocode via the VAXBI.
<03:00>	Mask 0	MASK0	Specifies which byte(s) are to be written for the first longword of a CIPA DMA write transaction. These bits can be written by the CIPA hardware microcode or by the VAX macrocode via the VAXBI.
<15:14>	Length	[illegible]	Specifies how many bytes of data are transferred over the VAXBI bus. <15:14> Length 0 Reserved 1 Longword (4 bytes) 2 Quadword (8 bytes) 3 Octaword (16 bytes)

Table 4-[illegible] DMA Command/Byte Mask Register Bit Definitions

Bit	Name	Mnemonic	Contents/Function
[illegible]	Command	CMD	Specifies the type of VAXBI transaction.
			<3:0> Bits — Command
			1 — READ
			2 — INTLK READ
			4 — WRITE
			6 — UNLOCK WRITE MASK
			7 — WRITE MASK
<07:04>	Mask 1	MASK1	Same bits written into the mask portion of BCMR bits <07:04>.
<03:00>	Mask 0	MASK0	Same bits written into the mask portion of BCMR bits <03:00>.

4.5.4 DMA Register File (DMAF)

Address Offset = 12C Hexadecimal

The DMAF is a 4 by 32 bit (longword) dual port RAM which holds an octaword's worth of data for a CIPA initiated write. The octaword storage allows overlapped quadword transactions for higher performance. Reading and writing are completely independent operations. As such, it is important to keep track of where the read address pointer and write address pointer are pointing.

The register is divided into high and low so that the 16-bit CIPA bus can write this 32-bit register. Figure 4-22 illustrates the register format.

NOTE
VMS will not normally write into this register.

4.6 CIPA REGISTERS

The CIPA registers consist of those registers which reside on the L-series modules within the CIPA cabinet. Figure 4-23 is a detailed illustration of the CIBCI adapter registers.

NOTE
Both the CIPA bus and CIPA hardware must be in working order to read and write these registers. Except as noted, the CIPA hardware must be in the uninitialized or uninitialized/maintenance state to read or write to these registers.

	31 … 16	15 … 00
bb+12C	FIRST LONGWORD HI	FIRST LONGWORD LO
bb+12C	SECOND LONGWORD HI	SECOND LONGWORD LO
bb+12C	THIRD LONGWORD HI	THIRD LONGWORD LO
bb+12C	FOURTH LONGWORD HI	FOURTH LONGWORD LO

MKV85-2667

Figure 4-22 DMA Register File Bit Map

bb+110	PORT MAINTENANCE CONTROL/STATUS REGISTER
bb+114	MAINTENANCE ADDRESS REGISTER
bb+900	PORT STATUS REGISTER
bb+904	PORT COMMAND QUEUE 0 CONTROL REGISTER
bb+908	PORT QUEUE BLOCK BASE REGISTER
bb+90C	PORT COMMAND QUEUE 1 CONTROL REGISTER
bb+910	PORT COMMAND QUEUE 2 CONTROL REGISTER
bb+914	PORT COMMAND QUEUE 3 CONTROL REGISTER
bb+918	PORT STATUS RELEASE CONTROL REGISTER
bb+91C	PORT ENABLE CONTROL REGISTER
bb+920	PORT DISABLE CONTROL REGISTER
bb+924	PORT INITIALIZE CONTROL REGISTER
bb+928	PORT DATAGRAM FREE QUEUE CONTROL REGISTER
bb+92C	PORT MESSAGE FREE QUEUE CONTROL REGISTER
bb+930	PORT MAINTENANCE TIMER CONTROL REGISTER
bb+934	PORT MAINTENANCE TIMER EXPIRATION CONTROL REGISTER
bb+938	PORT FAILING ADDRESS REGISTER
bb+93C	PORT ERROR STATUS REGISTER
bb+940	PORT PARAMETER REGISTER

MKV85-2558

Figure 4-23 CIPA Adapter Registers

4.6.1 Port Maintenance Control/Status Register (PMCSR)

Address Offset = 110 Hexadecimal

The PMCSR contains CIPA hardware error flags, interrupt, and CI bus initialization control bits. Figure 4-24 illustrates the register format. Each register bit is described in Table 4-10.

NOTE
Bits 14:08 cause the MTE bit in the port status register (PSR) to set, which generates an interrupt on the VAXBI and leaves the port in an uninitialized state.

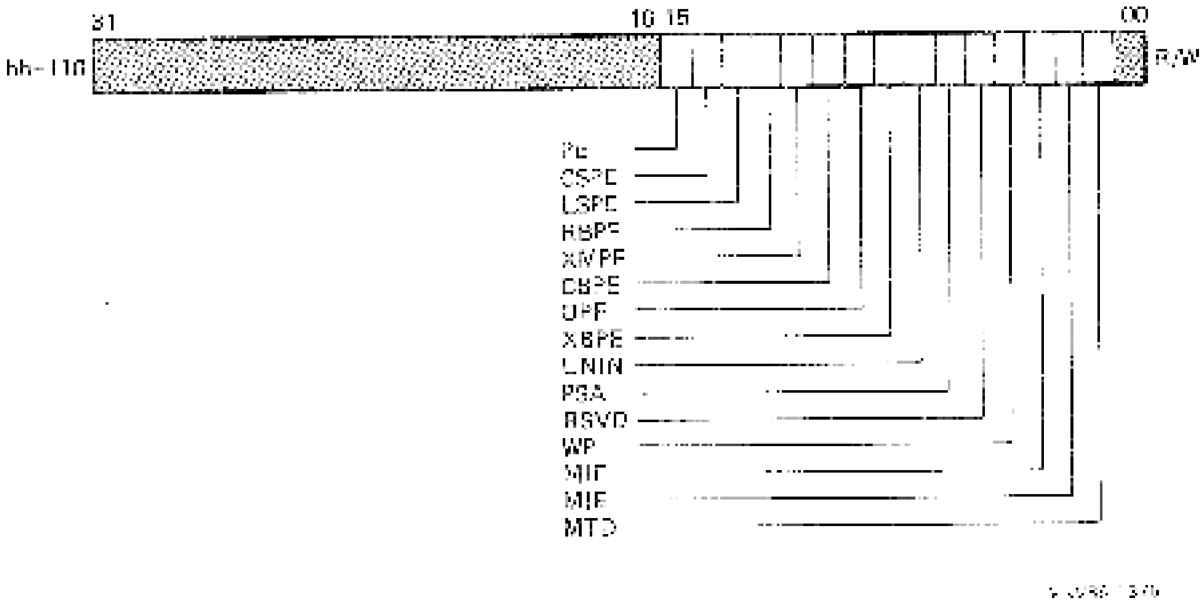


Figure 4-24 Port Maintenance Control/Status Register Bit Map

Table 4-10 Port Control and Status Register Bits

Bit	Name	Mnemonic	Meaning/Function
[illegible]	Parity error	PE	One or more [illegible] bits are set.
[illegible]	Control store parity error	CSPE	A parity error was detected reading PROM or RAM control store.
[illegible]	Local store parity error	LSPE	A parity error was detected while reading CIBCI local store on the [illegible] circuit [illegible].
[illegible]	Receive buffer parity error	RBPE	A parity error was detected while reading the CIBCI packet buffer.
[illegible]	Transmit buffer parity error	XMPE	A parity error was detected while the link module (LM) was unloading the CI transmit buffer.
[illegible]	CIPA bus parity error	CBPE	A parity error was detected while transferring data over the CIPA bus (from BICA to CIPA).
[illegible]	Output parity error	OPE	A parity error was detected on the [illegible] module while transferring data from CIPA to BICA.
[illegible]	Transmit buffer parity error	XBPE	A parity error was detected while the packet buffer and control store module ([illegible]) was accessing the CI bus transmit buffer.
[illegible]	Uninitialized	UNIN	The CIBCI is in the uninitialized state. The microcode is not running and the port will not respond to CI bus commands. It is set by MINIT or [illegible] in the [illegible], or MTE in the PSR.
			It is cleared by writing a 1 to the PIC bit in the port maintenance control register (PMCR), or after a boot timeout. Clearing UNIN starts the microcode.
[illegible]	Programmable starting address	PSA	When set, instructs the microcode to start at the address specified in the MADR, after the PIC bit of the PMCR register is set to a 1 or after a boot timeout.
			When clear, instructs the microcode to start at PROM location zero.
[illegible]	Wrong parity	WP	The parity checker/generator will check/generate even parity rather than odd parity on the [illegible] of the [illegible] module.

Table 4-10 Port Control and Status Register Bits (Cont)

Bit	Name	Mnemonic	Condition/Function
<03>	Maintenance interrupt flag	MIF	Indicates that an interrupt condition has occurred in the CIBCI port. MIF is used with MIE to allow a diagnostic program to operate the port with interrupts disabled. MIF indicates to the program that the PSR has valid data. It is cleared by writing a 1 to the PSRC bit of the port status release control register (PSRCR) or on UNIT.
<04>	Maintenance interrupt enable	MIE	Indicates that the CIBCI port interrupts are enabled.
<05>	Maintenance timer disabled	MTD	When set, disables the boot and sanity timers from causing an interrupt. When clear, the boot and sanity timers are enabled and the port maintenance timer control register (PMTCR) must be periodically written to prevent the CIBCI port from entering an uninitialized/disabled state and generating a sanity timer expiration (SE bit of the PSR) interrupt.
<06>	Maintenance initialize	MIN	This bit is only used by the CIBCI port and is read as zero.

4.6.2 Maintenance Address Register (MADR)

Address Offset = 114 Hexadecimal

VMS uses this register to load CIPA control store microcode. The address loaded into the MADR is used as a pointer address to access control store. Control store is organized as 3K by 48 bits under the control of the CIPA hardware, in the initialized state. In the uninitialized state, the control store appears as 6K worth of addresses each containing 32 bits of data. The MADR is only effective when the CIPA is in the uninitialized state.

Figure 4-25 is a diagram of control store addresses. Each register bit is described in Table 4-11.

NOTE

Register contents are valid only when the port is in the uninitialized state. MADR may be used as the start of the CIPA control store microcode address if the microcode is started with PMCSR PSA bit=1. Also, the control store PROM area is read only with the exception of microcode bit 47 which is the synchronous bit that allows tracking of which PROM locations are used.

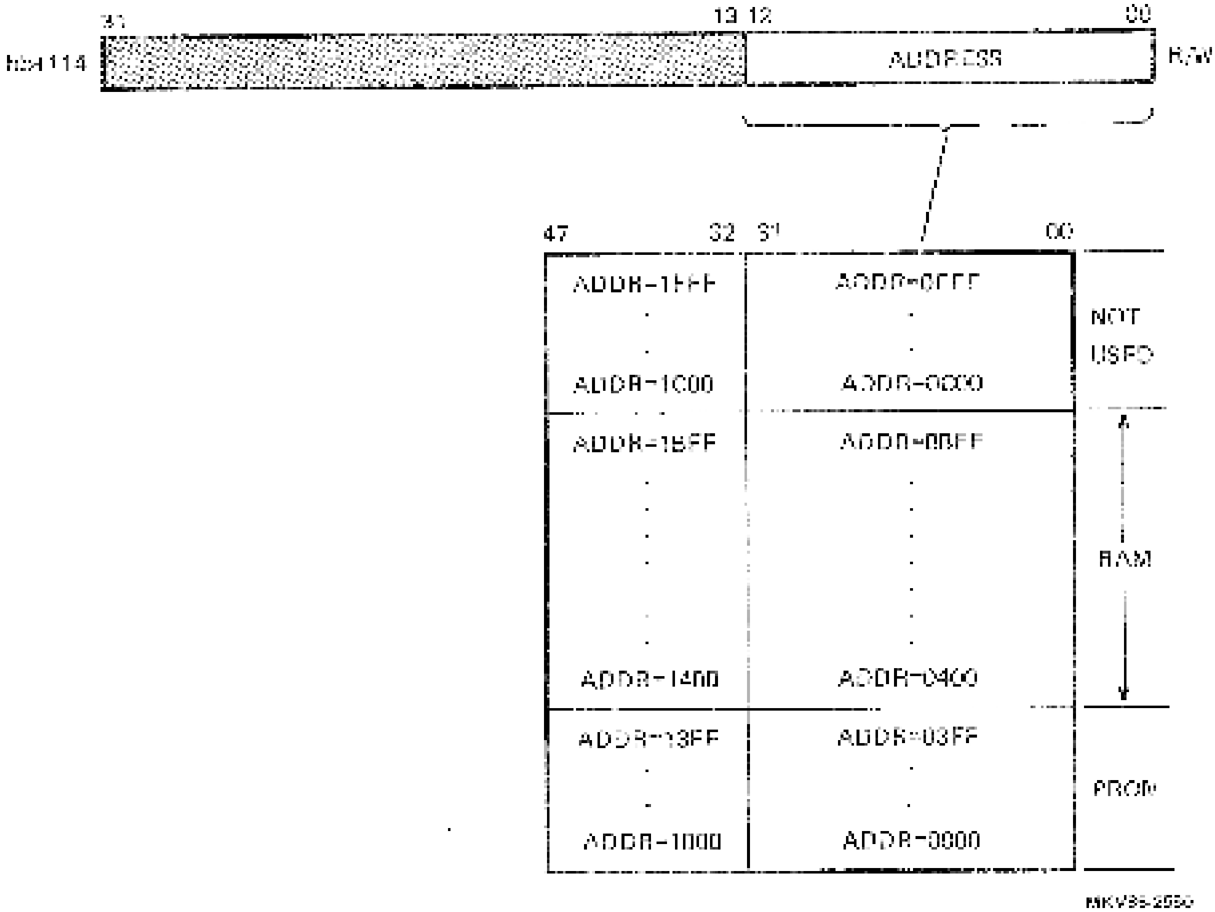


Figure 4-25 Maintenance Address Register Bit Map

Table 4-41 Maintenance Address Register Bits

Bit	Name	Mnemonic	Comment/Function
<12>	A12		Selects a segment of the control store word as follows: 0 = selects microcode bits <31:00> 1 = selects microcode bits <47:32>
<11:10>	A11:A10	—	Selects a 1K bank within the control store as follows: A11 A10 Bank Selected 0 0 0 (000-3FF PROM microcode) 0 1 1 (400-7FF RAM microcode) 1 0 2 (800-BFF RAM microcode) 1 1 3 (C00-FFF reserved)
<09:00>	A09:A00	—	Selects the word within each 1K bank.

4.6.3 Maintenance Data Register (MDATR)

Address Offset — 118 Hexadecimal

VMS uses this register to load CIPA control store microcode. The data in MDATR is loaded into CIPA control store locations specified in the previous description of MADR. Again, only microcode bit 47 in the PROM area can be written. Control store bits 47:32 appear on MDATR bits 15:00 respectively for a read, and likewise contain the data for a write. Figure 4-26 illustrates the register format.

NOTE

Register contents are valid only when the port is in the uninitialized state.

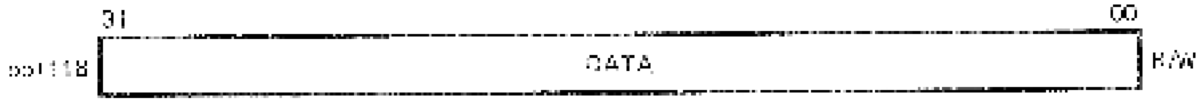


Figure 4-26 Maintenance Data Register Bit Map

4.6.4 Port Status Register (PSR)

Address Offset — 900 Hexadecimal

The PSR contains the status flags that indicate the cause of a port generated interrupt. The PSR contents are fixed once the port issues an interrupt, and are not changed until the VMS port driver software releases PSR by writing the PSRC bit of the PSRCR to a 1. Therefore, it is valid only during that interval. Figure 4-27 illustrates the register format. The bit assignments are described in Table 4-12.

NOTE

This register is used by the VMS port driver software and is valid only when the operational port microcode is loaded and running.

4.6.5 Port Queue Block Base Register (PQBBR)

Address Offset — 904 Hexadecimal

This register contains the physical address of the base of the port queue block. Figure 4-28 illustrates the register format.

NOTE

The PQBBR is a read/write register, however, writing to the PQBBR is permitted only when the port is in the disabled or disabled/maintenance state.

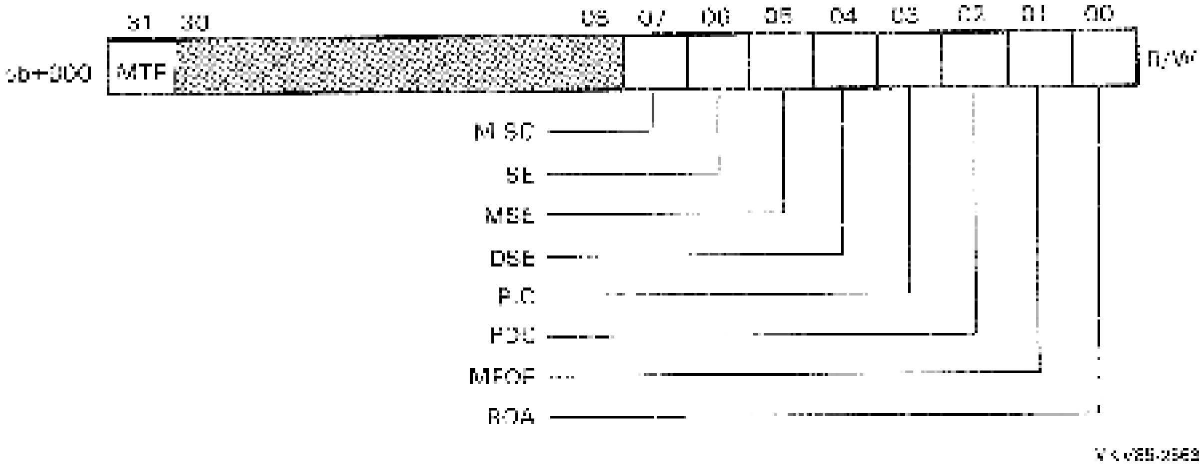


Figure 4-27 Port Status Register Bit Map

Table 4-12 Port Status Register Bits

Bit	Name	Mnemonic	Comment/Function
<31>	Maintenance error	MTE	The CIXCD port has detected an internal hardware failure. The exact error can be determined by reading the XBE, SE, the CONFIG, or the BER registers. When MTE is set: 1. The port enters the uninitialized state. 2. The microcode halts. 3. An interrupt is generated, on the BI (PSR <06:00> is invalid.)
<07>	Miscellaneous error	MISC	The microcode has detected a miscellaneous error condition(s) and is about to enter a hang state.
<06>	Sanity timer expiration	SE	The boot or sanity timer has expired and the port has entered the uninitialized/maintenance state.
<05>	Memory system error	MSE	A VAXBI transaction type error was detected. For example: 1. A read data substitute (RDS) error. Equivalent to an uncorrectable data error. 2. A nonexistent address (NEXM) error. Equivalent to a nonexistent memory error.

Table 4-13 Port Status Register Bits (cont.)

Bit	Name	Mnemonic	Contents/Function
<04>	Data structure error	DSE	The CIBCI port has encountered an error in the host data structure, for example: • Queue entry • PQB • BDT • Page table • Value out of range • Zero bits that are not zero
<03>	Port initialization complete	PIC	The CIBCI port has completed initialization of the local store, virtual circuit descriptor table, and internal data structures. The CIBCI port is in the disabled/maintenance or disabled state.
<02>	Port disable complete	PDC	The CIBCI port has stopped responding to: 1. The command queues. 2. Incoming CI transmissions (except maintenance, if enabled). The port is in the disabled or disabled/maintenance state.
<01>	Message free queue empty	MFQE	The CIBCI port has attempted to remove an entry from the message free queue and found it empty.
<00>	Response queue available	RQA	The CIBCI port has inserted an entry onto an empty response queue.

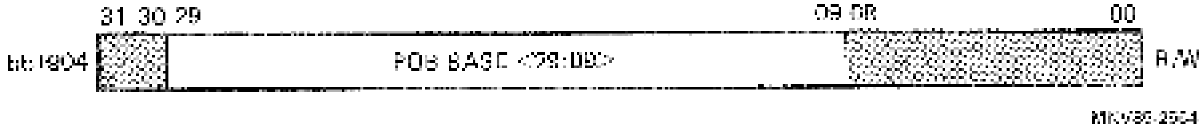


Figure 4-28 Port Queue Block Base Register Bit Map

4.6.6 Port Command Queue 0 Control Register (PCQ0CR)

Address Offset — 908 Hexadecimal

After the VMS port driver software inserts one or more entries into the empty command queue 0, it will write a 1 into bit 00 of the PCQ0CR to initiate port processing of command queue 0. Figure 4-29 illustrates the register format.

NOTE

The port processor ignores PCQ0CR if it is in the uninitialized/maintenance, disabled/maintenance, uninitialized, or disabled state.

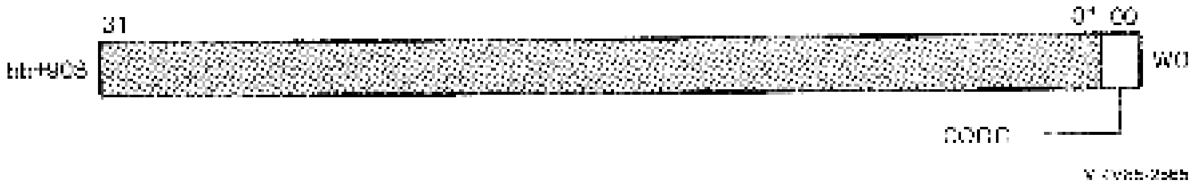


Figure 4-29 Port Command Queue 0 Control Register Bit Map

4.6.7 Port Command Queue 1 Control Register (PCQ1CR)

Address Offset — 90C Hexadecimal

After the VMS port driver software inserts one or more entries into the empty command queue 1, it will write a 1 into bit 00 of PCQ1CR to initiate port processing of command queue 1. Figure 4-30 illustrates the register format.

NOTE

The port processor ignores PCQ1CR if it is in the uninitialized/maintenance, disabled/maintenance, uninitialized, or disabled state.

Figure 4-30 Port Command Queue 1 Control Register Bit Map

4.6.8 Port Command Queue 2 Control Register (PCQ2CR)

Address Offset = 910 Hexadecimal

After the VMS port driver software inserts one or more entries into the empty command queue 0, it will write a 1 into Bit 00 of the PCQ2CR to initiate port processing of command queue 2. Figure 4-31 illustrates the register format.

NOTE

The port processor ignores PCQ2CR if it is in the uninitialized/maintenance, disabled/maintenance, uninitialized, or disabled state.

Figure 4-31 Port Command Queue 2 Control Register Bit Map

4.6.9 Port Command Queue 3 Control Register (PCQ3CR)

Address Offset = 914 Hexadecimal

After the VMS port driver software inserts one or more entries into the empty command queue 0, it will write a 1 into bit 00 of the PCQ3CR to initiate port processing of command queue 3. Figure 4-32 illustrates the register format.

NOTE

The port processor ignores PCQ3CR if it is in the uninitialized/maintenance, disabled/maintenance, uninitialized, or disabled state.

Figure 4-32 Port Command Queue 3 Control Register Bit Map

4.6.10 Port Status Release Control Register (PSRCR)

Address Offset — 918 Hexadecimal

After the VMS port driver software has received the interrupt issued by the port processor, and has read the PSR, CNFGR, and BER registers, it returns control of the PSR (port processor is able to write PSR) back to the port processor by writing a 1 to the PSRC bit. Figure 4-33 illustrates the register format.

NOTE
The port processor ignores PSRCR if it is in the uninitialized/maintenance, disabled/maintenance, uninitialized, or disabled state.

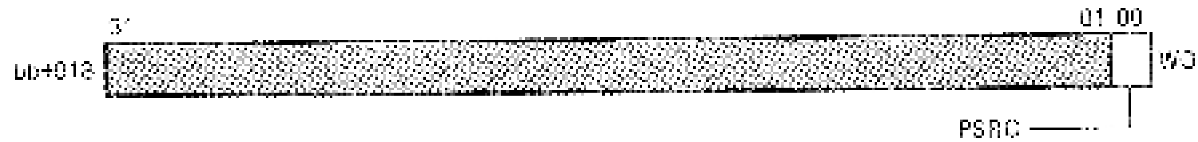


Figure 4-33 Port Status Release Control Register Bit Map

4.6.11 Port Enable Control Register (PECR)

Address Offset — 91C Hexadecimal

The VMS port driver software places the port processor in the enabled or enabled/maintenance state by writing a 1 into bit 00 of the PECR. Figure 4-34 illustrates the register format.

NOTE
The port processor ignores PECR if it is in the enabled, uninitialized, uninitialized/maintenance, or enabled/maintenance state.

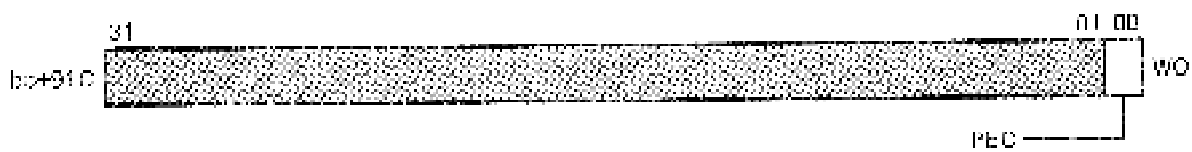


Figure 4-34 Port Enable Control Register Bit Map

4.6.12 Port Disable Control Register (PDCR)

Address Offset — 920 Hexadecimal

The VMS port driver software places the port processor in the disabled or disabled/maintenance state by writing a 1 into bit 00 of the PDCR. When the port processor is disabled (the microcode having completed a disable sequence), it requests an interrupt by setting the PDC bit set in the PSR. Figure 4-35 illustrates the register format.

NOTE

The port processor ignores PDCR if it is in the uninitialized/maintenance, disabled/maintenance, uninitialized, or disabled state.

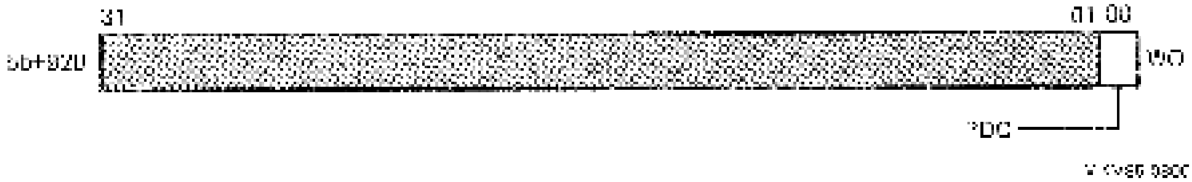


Figure 4-35 Port Disable Control Register Bit Map

4.6.13 Port Initialize Control Register (PICR)

Address Offset — 924 Hexadecimal

The VMS port driver software places the port processor in the initialized state by writing a 1 into bit 00 of the PICR. When the port processor completes initialization, it sets the PIC bit in the PSR, requests an interrupt, and enters the disabled or disabled/maintenance state. Figure 4-36 illustrates the register format.

NOTE

The port processor ignores PICR if it is in the disabled/maintenance or disabled state. The port processor goes to the disabled/maintenance or disabled state if it looses processing status.

Figure 4-36 Port Initialize Control Register Bit Map

4.6.14 Port Datagram Free Queue Control Register (PDFQCR)

Address Offset – 928 Hexadecimal

The VMS port driver software sets the DFQC bit in PDFQCR if the datagram free queue is empty. Figure 4-37 illustrates the register format.

NOTE

The port processor ignores PDFQCR if it is in the uninitialize/maintenance, disabled/maintenance, uninitialized, or disabled state.

Figure 4-37 Port Datagram Free Queue Control Register Bit Map

4.6.15 Port Message Free Queue Control Register (PMFQCR)

Address Offset – 92C Hexadecimal

The VMS port driver software sets the MFQC bit in PMFQCR if the message free queue is empty. Figure 4-38 illustrates the register format.

NOTE

The port processor ignores PMFQCR if it is in the uninitialized, disabled, uninitialized/maintenance, or disabled/maintenance state.

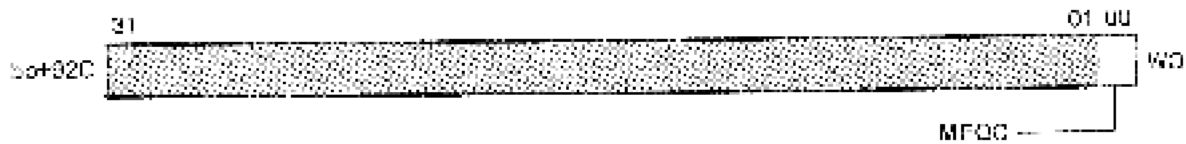


Figure 4-38 Port Message Free Queue Control Register Bit Map

4.6.16 Port Maintenance Timer Control Register (PMTCR)

Address Offset = 930 Hexadecimal

The PMTCR allows the VMS port driver software to control the expiration time of the boot and sanity timers. Both timers are reset to their initial value whenever a 1 is written into bit 00 of the PMTCR. If PMTCR is not written before the expiration time, then the port will request an SE interrupt (SE bit of the PSR is set) and enter the uninitialized/maintenance state. Figure 4-39 illustrates the register format.

NOTE
PMTCR is ignored if the maintenance timers are disabled through either the MTD bit being set in the PMCSR, or the port entering the uninitialized state.

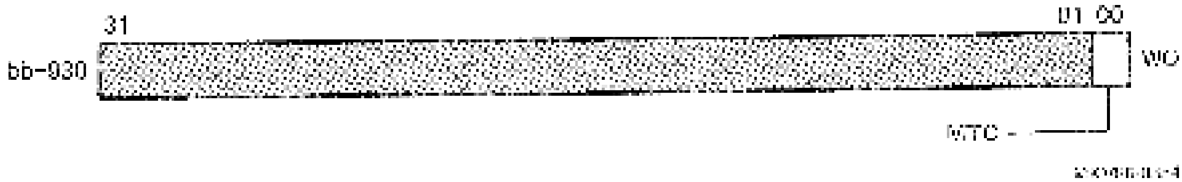


Figure 4-39 Port Maintenance Control Register Bit Map

4.6.17 Port Maintenance Timer Expiration Control Register (PMTECR)

Address Offset = 934 Hexadecimal

The VMS port driver software forces a maintenance timer expiration interrupt by setting bit 00 of the PMTECR. Figure 4-40 illustrates the register format.

NOTE
PMTECR can be written only when the MTD bit in PMCSR is clear, and the port is in the enabled, enabled/maintenance, disabled/maintenance, or disabled state.

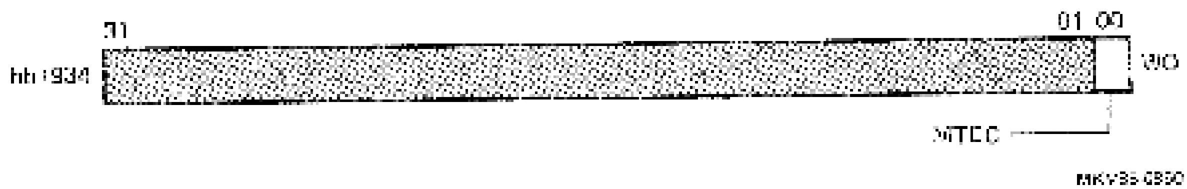


Figure 4-40 Port Maintenance Timer Expiration Control Register Bit Map

4.6.18 Port Failing Address Register (PFAR)

Address Offset = 938 Hexadecimal

The PFAR contains the memory address at which a failure occurred after a memory system error (MSE) or data structure error (DSE) interrupt, or after a response with a buffer memory system error status (type=6 in the response status field of VMS issued port commands). The failing address may be the exact address, an address in the same page as the failing address, or, in the case of DSE interrupts, an address in some part of the data structure.

Figure 4-41 illustrates the register format. The bit assignments are described in Table 4-13.

NOTE

The PFAR is used by the VMS port driver software and is valid only when the operational port microcode is loaded and running.

The contents of the PFAR register are only valid when the MSE or DSE bit of the PSR is set (first error logged), or there is a response with a buffer memory system error (last error logged).

Figure 4-41 Port Failing Address Register Bit Map

Table 4-13 Port Failing Address Register Bits

Bit	Error Type	PFAR Contents
<31:00>	DSE interrupt	Contains either an offset or virtual address
—	MSE interrupt	Contains a physical address
—	Buffer memory system error status	Contains a physical address

4.6.19 Port Error Status Register (PESR)

Address Offset = 93C Hexadecimal

The PESR indicates the type of error which resulted from a DSE interrupt (DSE bit in the PSR being set). Figure 4-42 illustrates the register format. The bit assignments are described in Table 4-14.

NOTE

This register is used by the VMS port driver software and is valid only when the operational port microcode is loaded and running, and only after a DSE interrupt has occurred.

Figure 4-42 Port Error Status Register Bit Map

Table 4-14 Port Error Status Register Bits

Hex Code	VMS Error	PESR Contents	Error Description
1	[illegible]	Virtual address	Illegal system virtual address formed [illegible]
2	NX_SYS_VA	Virtual address	Nonexistent system virtual address [illegible]
3	INV_SYS_PTE	Virtual address	Invalid system PTE (Bits <31, 26, 22>)
4	INV_[illegible]_PTE	PTE virtual address	Invalid system PTE (Bits <31, 26, 22>)
5	NX_[illegible]_SVA	Virtual address	Nonexistent system global virtual address [illegible]

4.6.20 Port Parameter Register (PPR)

Address Offset — 940 Hexadecimal

The PPR contains the port number and other port parameters. It is set up by the CIPA control store microcode during the port initialization process and is valid in any state except the uninitialized state. Figure 4-43 illustrates the register format. The bit assignments are described in Table 4-15.

NOTE

PPR is a read/write register but writing to the PPR will destroy the port state with unpredictable results. This register is used by the VMS port driver software and is valid only when the operational port microcode is loaded and running.

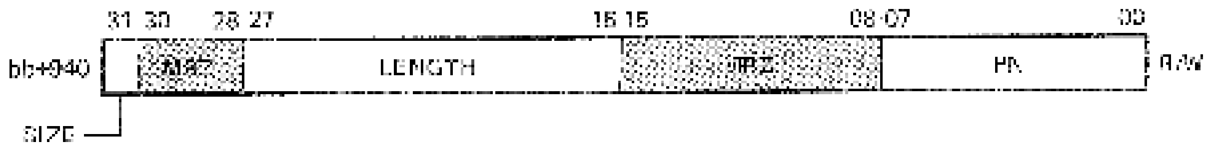


Figure 4-43 Port Parameter Register Bit Map

Table 4-15 Port Parameter Register Bits

Bit	Name	Comment/Function
<31>	System size	When clear indicates that there is maximum of 16 nodes allowed on the CI. When set indicates that there is a maximum of 224 nodes allowed on the system.
<27:16>	Internal buffer length	Indicates the size of the internal buffers available for the message and data transfers. It is preset to 1016 (3F8 hexadecimal).
<07:00>	Port number	Indicates the CI adapter node number set by the switches located on the LINK module.

APPENDIX A
CI TERMINATION

NOTE

Only one unterminated pair of cables are permitted per data path on the CI bus. The above rule allows for the following:

- **Power removal and restoration for an individual node.**
- **Removal of the L0100 module while the node is fully cabled.**
- **Disconnection of any or all cables at any point between a node backplane and the star coupler.**

The procedures listed below will not disrupt the operation of properly configured nodes on the same data path in an on-line situation.

- When testing the CI, the L0100 module should always have proper termination. This implies using either the star coupler or proper attenuator pads for termination.
- If using the star coupler, be aware that diagnostics do not expect collisions on the CI. Diagnostic errors can occur if the node being tested is attached to a "live" star coupler.
- The attenuator pad may be used either at the bulkhead panel assembly of the CI port with modularity cables or at the end of the CI cables at the star coupler end.

APPENDIX B
CI BACKPLANE JUMPER

B.1 BOOT TIMER PARAMETERS

Time Sec	W1	W2	W3	W4
0000	IN	IN	IN	IN
0100		IN	IN	IN
0200	IN		IN	IN
0300			IN	IN
0400	IN	IN		IN
0500		IN		IN
0600	IN			IN
0700				IN
0800	IN	IN	IN	
0900		IN	IN	
1000	IN		IN	
1100			IN	
1200	IN	IN		
1300		IN		
1400	IN			
1500				

B.2 EXTENDED HEADER/TRAILER (W5)

In = Extended header/trailer
Out = Normal header/trailer

B.3 ALTER DELTA TIME (W6)

In = Long delta time
Out = Short delta time

B.4 DISABLE ARBITRATION (W7)

In = Disable normal arbitration
Out = Allow normal arbitration

B.5 EXTENDED ACKNOWLEDGEMENT TIMEOUT (W8)

In — Long timeout
Out — Short timeout

NOTE
Jumper W8 must be in whenever jumper W5 is in.