CONVEX Integrated Disk Channel Service Guide

Document No. 081-001530-001

Second Edition
June 1990

CONVEX Computer Corporation
Richardson, Texas USA

CONVEX Integrated Disk Channel
Service Guide
Order No. DHW-025
Second Edition

Although the material contained herein has been carefully reviewed, CONVEX does not warrant it to be free of errors or omissions. CONVEX reserves the right to make corrections, updates, revisions, or changes to the information contained herein. CONVEX does not warrant the material described herein to be free of patent infringement.

UNLESS PROVIDED OTHERWISE IN WRITING WITH CONVEX COMPUTER CORPORATION (CONVEX), THE EQUIPMENT DESCRIBED HEREIN IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES. THE ABOVE EXCLUSION MAY NOT BE APPLICABLE TO ALL PURCHASERS BECAUSE WARRANTY RIGHTS CAN VARY FROM STATE TO STATE. IN NO EVENT WILL CONVEX BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING ANY LOST PROFITS OR LOST SAVINGS, ARISING OUT OF THE USE OR INABILITY TO USE THIS EQUIPMENT. CONVEX WILL NOT BE LIABLE EVEN IF IT HAS BEEN NOTIFIED OF THE POSSIBILITY OF SUCH DAMAGE BY THE PURCHASER OR ANY THIRD PARTY.

Printed in the United States of America

Revision Sheet
CONVEX Integrated Disk Channel
Service Guide

Edition	Document No.	Date	Description
Second	081-001530-001	June 1990	Included information on the 3-Mbyte-per-second Disk Drive
First	081-001530-000	January 1990	First release, Version 1.0

FCC NOTICE

Warning: This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in strict accordance with the instructions manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

Do not connect external equipment to the utility outlets in CONVEX equipment cabinets. Unauthorized connection voids all agencies' emissions certifications.

Table of Contents

Appendixes

List of Tables

List of Figures

Preface

Purpose and Intended Audience

The *CONVEX Integrated Disk Channel Service Guide* provides a general overview of the CONVEX Integrated Disk Channel (IDC) subsystem and related hardware and how to:

- Install a CONVEX Integrated Disk Channel and related hardware
- Integrate a CONVEX Integrated Disk Channel and related hardware into the CONVEX Operating System (ConvexOS)
- Test a CONVEX Integrated Disk Channel and related hardware
- Remove and replace a CONVEX Integrated Disk Channel and related hardware

Primary Audience

This document is intended for:

- CONVEX Customer Support Engineers and CONVEX manufacturing personnel
- Customers who need to install or maintain a CONVEX Integrated Disk Channel subsystem

Distribution

This document is primarily for CONVEX customer support engineers use. It may be supplied to those customers who wish to install or maintain their own equipment.

Organization

The document consists of the following sections:

- **Chapter 1, Description and Specifications**—Describes the CONVEX Integrated Disk Channel and related hardware. Defines and lists the electromechanical and environmental specifications.
- **Chapter 2, Unpacking and Installation**—Provides guidelines on how to unpack and install a CONVEX Integrated Disk Channel and related hardware.
- **Chapter 3, Integration and Test**—Explains how to integrate a CONVEX Integrated Disk Channel and related hardware into the CONVEX Operating System (ConvexOS). Explains how to test a CONVEX Integrated Disk Channel and related hardware.
- **Chapter 4, Maintenance Procedures and IPB**—Provides removal and replacement instructions for a CONVEX Integrated Disk Channel and related hardware.

- **Appendix A, Problem Reporting**—Provides an example of the CONVEX *contact* utility for reporting minor software and hardware problems.

Terminology

The terminology conventions used in this text are listed below:

BARB—Buffer Arbitration Logic

DIAG—Diagnostic Logic

DICE—Device Interface Command Execution Gate Array

DPED—Data Path and Error Detection Gate Array

ICTL—Interrupt Control Logic

IDC—Integrated Disk Channel

IPI—Intelligent Peripheral Interface

MPU—Main Processing Unit

PBI—PBUS Interface

PIGA—PBUS Interface Gate Array

Notational Conventions

The following are examples of warnings, cautions, and notes and their typical content as used in CONVEX documents:

> **WARNING**
>
> Warnings highlight procedures or information necessary to avoid injury to personnel. A warning immediately precedes the critical information and includes a description of the hazard.

> **CAUTION**
>
> Cautions highlight procedures or information necessary to avoid damage to equipment, loss of data, or invalid test results. A caution immediately precedes the critical information and includes a description of the possible damage.

> **NOTE**
>
> Notes highlight useful information that is supplemental in nature. A note may immediately precede or follow the information that is being highlighted.

Associated Documents

The following is a partial list of other manuals or books that may provide more detailed information on the topics presented in this manual:

- *CONVEX Computers Site Preparation Guide,* Order No. DHW-009
- *CONVEX Diagnostic Utility Manual,* Order No. DHW-082
- *CONVEX PBUS I/O System Diagnostics Manual,* Order No. DHW-008
- *CONVEX Processor Operation Guide,* Order No. DHW-015
- *CONVEX System Manager's Guide,* Product No. DSW-004
- *Electrostatic Discharge Failures of Semiconductor Devices.* Unger, B.A. 1981. Bell Laboratories

Ordering Documentation

To order the most current version of this or any other CONVEX document, use the CONVEX order number. If the order number is not known, order by exact title. In some situations the most current version may not be desired. To receive a specific version of a manual, order the manual by its document, or part, number, which can be obtained by contacting the local CONVEX office or by calling the Technical Assistance Center.

The order number for this document is DHW-025
The part number for this document is 081-001530-001

CONVEX documents can be ordered by mail by sending a request to:

CONVEX Computer Corporation
Customer Service
PO Box 833851
Richardson Texas 75083-3851 USA

Technical Assistance

Hardware and software support can be obtained through the CONVEX Technical Assistance Center (TAC):

- From all locations in the continental United States, call 1(800)952-0379
- From locations in Alaska, Hawaii, and Canada, call 1(214)497-4379
- From all other locations, contact the nearest CONVEX office.

Electronic Mail

The Hardware Documentation Group has an email address for documentation comments. Use this service to give us a quick response mechanism if you have special documentation questions that you would like addressed immediately. If you have a technical question, you should still contact the Technical Assistance Center, as described previously. To use email response service, just send mail addressed to:

```
cnvxhwdoc@convex.COM
```

We will read your comments and give you a personal reply.

What to Include in an Email Message

When you use the electronic mail service, please provide the following information:

- The reader's name and company name
- A return email address in INTERNET notation or UUCP (bang) notation
- The manual that is being critiqued
- The chapter and page number in question
- The comment

Reader's Forum

If you wish to mail your comments to us, please use the form at the end of this manual and list the document page number with your questions and comments. Thank you.

Acknowledgments

I would like to thank the following people for their contributions to this manual:

- Technical contributors: Brad Culter, Joe Machado, Don May, Phil Mueller, Alan Peterson, Larry Price, John Rachels, and Gary Stager
- Document review team: Ray Anderson, Bill Benson, Larry Bonura, Jeff Christenson, Brad Culter, and Gary Stager
- Hardware documentation staff: Bill Benson, Larry Bonura, and Josie Davis

Without the efforts of the aforementioned, this document would not have been possible.

Randall Stiles
CONVEX Hardware Documentation

THIS PAGE INTENTIONALLY LEFT BLANK

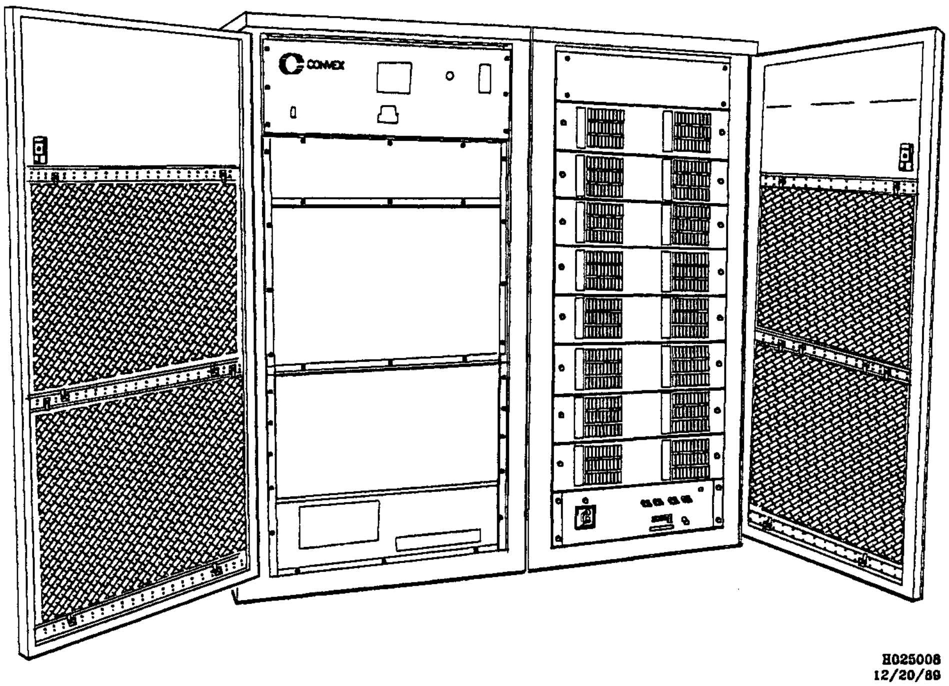
Chapter 1
Description and Specifications

1.1 Overview

The Integrated Disk Channel (IDC) is a high-capacity, high-performance Channel Control Unit (CCU). The IDC includes four peripheral interface ports that conform to the ANSI Intelligent Peripheral Interface (IPI) physical interface specification. Each port is capable of supporting eight 1-Gbyte disk drives. A CONVEX C220 computer and a peripheral cabinet with 16 1-Gbyte disk drives is illustrated in Figure 1–1, "C220/Peripheral Cabinet":

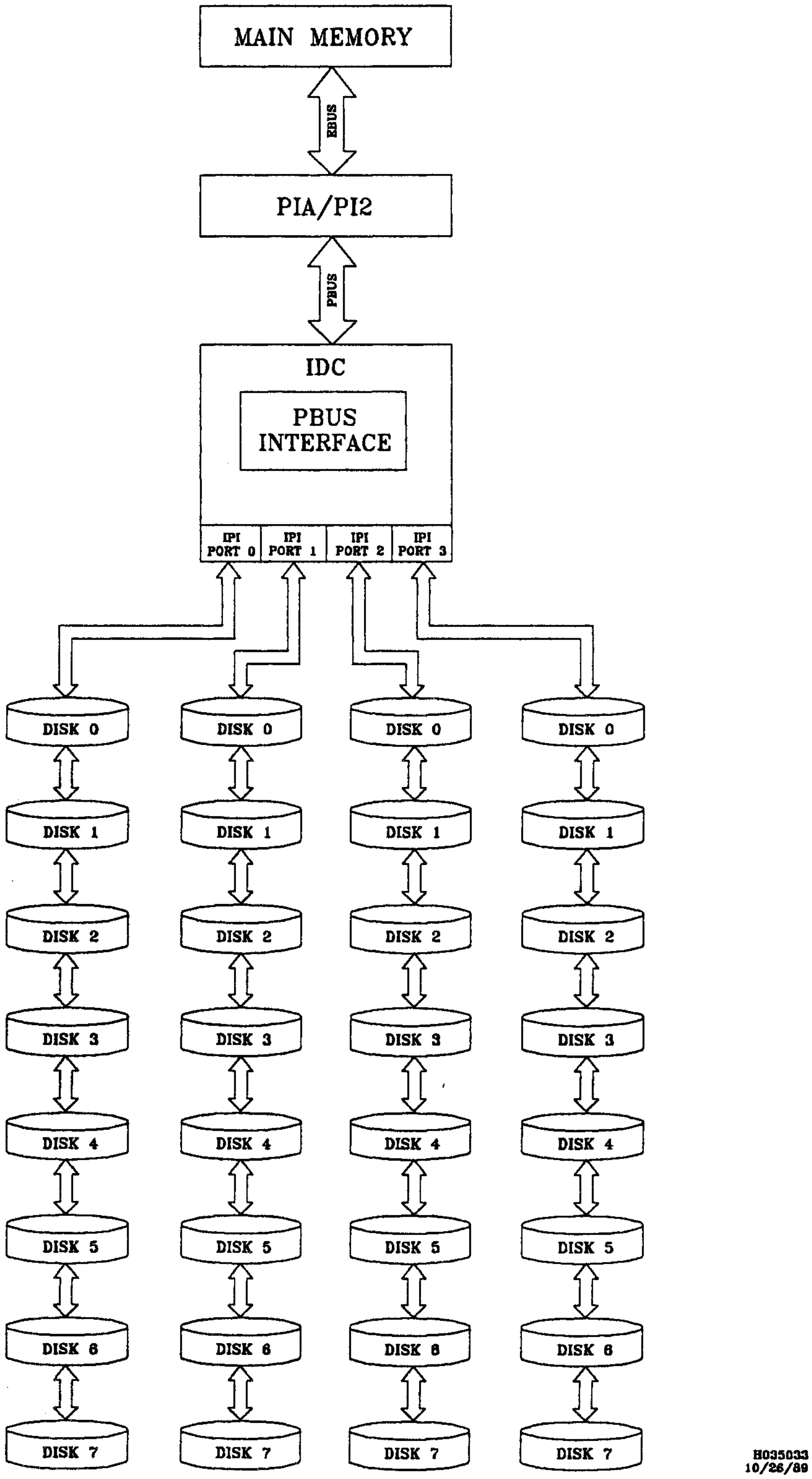
Figure 1–1, C220/Peripheral Cabinet

C220/Peripheral Cabinet
(FRONT VIEW)



The IDC communicates with the operating system kernel via the Message Based System (MBS) and interrupts. The IPI-2 logical command set for magnetic disks is implemented with a device driver in Main Processing Unit (MPU) software. Physically, an IDC consists of a PBUS interface to main memory, an onboard MPU, a high-bandwidth multiplexing data buffer, and four IPI ports. An IDC subsystem block diagram is illustrated in Figure 1–2, "IDC Block Diagram":

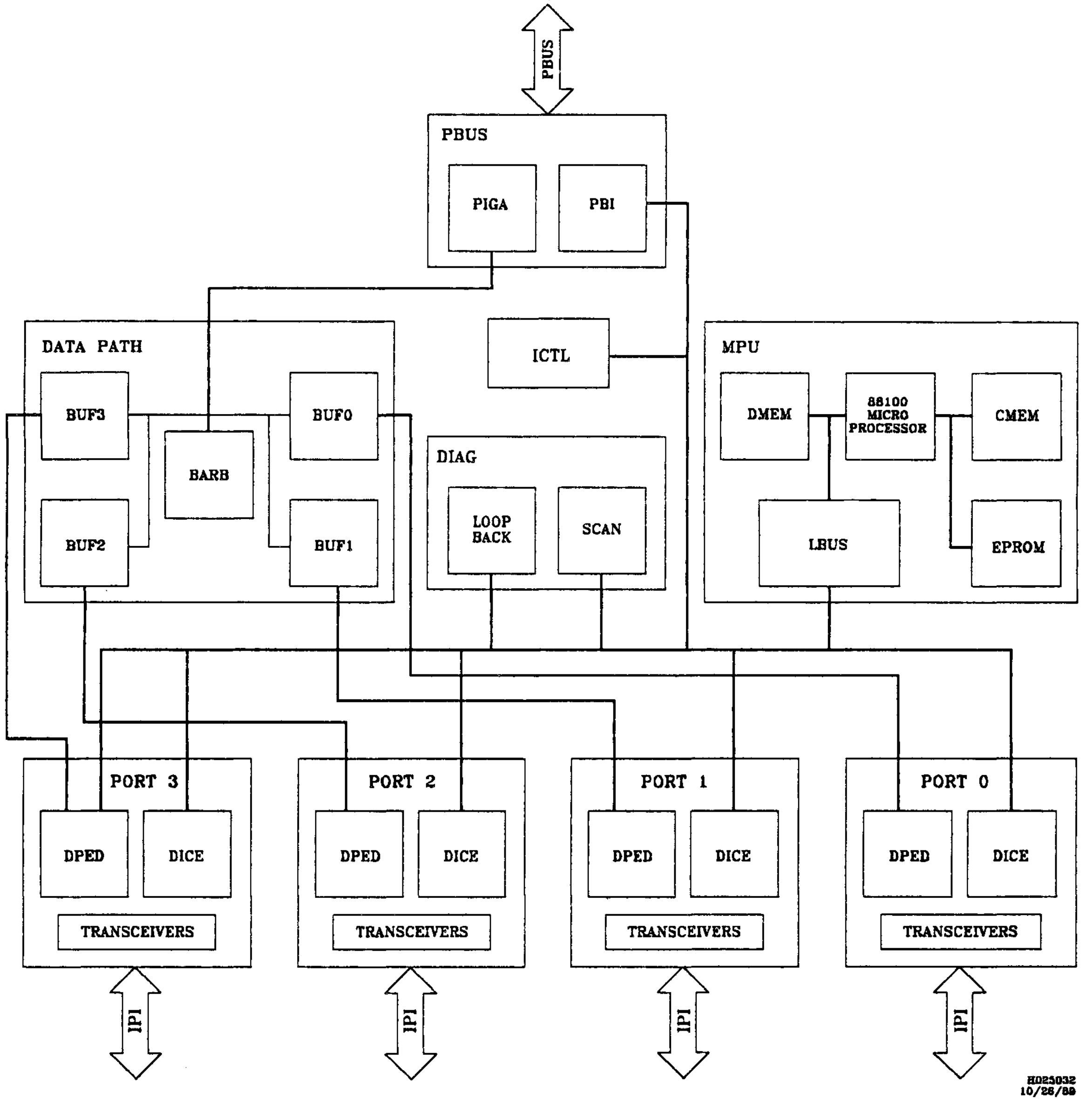
Figure 1–2, IDC Block Diagram



1.2 Integrated Disk Channel Logical Organization

Since many IDC operations are distributed among multiple hardware units, the IDC is organized as several logical units and subsystems according to functionality. An IDC functional block diagram is illustrated in Figure 1–3, "IDC Functional Block Diagram".

Figure 1–3, IDC Functional Block Diagram



The following sections describe the various logical units and subsystems of the IDC.

1.2.1 Data Path

The primary function of the IDC is to transfer data between external storage devices and main memory. The IDC controls nearly all aspects of data I/O including device selection, media positioning, rate and block size matching, buffer management, address translation, and error detection and correction. All elements of the IDC are involved in data transfers to some degree.

During data input operations (read operations), information is received asynchronously, 18 bits (2 bytes plus parity) at a time by the Data Path and Error Detection (DPED) gate array. The data is packed into long words and written to consecutive locations in a dual ported 72-by-2K RAM buffer dedicated to each IPI port. Buffer Arbitration (BARB) logic coordinates buffer access on a *buffer block* granularity. The arbiter detects empty and full conditions and enables buffer access accordingly. The buffer starting address, the transfer count, and the buffer block size are written into registers in the DPED gate array and the PBUS Interface Gate Array (PIGA).

The PIGA is also programmed with the starting logical address in main memory and the parameters (Segment Descriptor Register (SDR) and Page Table Entry (PTE) pointers) required for address translation. Again under control of the arbiter logic, the PIGA removes data from the IDC buffer a block at a time. The data is byte aligned, if necessary, as it passes through the PIGA. The PBUS Interface (PBI) logic functions in concert with the PIGA gate array to transfer blocks of data to main memory in accordance with the PBUS protocol (performing address translations as necessary). The IDC port data buffers are circular, which allows an indefinite number of blocks of data to be transferred between peripherals and main memory. The number of blocks of data that can be transferred continuously is constrained only by IPI and PBUS bandwidth limits.

Data output operations (write operations) are the reverse of read operations. The PIGA fetches blocks of data from main memory and places the data in a port buffer. The DPED gate array transfers two bytes of data at a time across the IPI interface.

1.2.2 PBUS Interface

The PBUS interface on the IDC functions as the physical interface to system level units (Central Processing Unit (CPU), Service Processor Unit (SPU), Peripheral Interface Adapter (PIA)/memory) and is responsible for the movement of data between these units and either the MPU or the onboard data buffer. The PBUS interface can be divided into the following three major functional blocks:

1. A PBUS sequencer
2. A micro machine for data buffer management and address translation
3. Byte alignment logic for the data path

The first function is implemented in the PBUS Interface (PBI) logic. The PBI logic executes the PBUS signal protocol. The latter two functions are implemented in the PIGA. The PIGA generates headers for transfers associated with I/O data transfers (reading SDRs and PTEs, and data block transfers).

The IDC interfaces to the PBUS interrupt structure through the IBUS logic. The IBUS logic is organized into four interrupt receiver groups. Each group has an associated interrupt status bit, an enable bit, and a set of vector map bits that determines which of the 256 possible PBUS interrupts are recognized by that group. Any number and combination of vectors can be mapped to each of the four groups. PBUS interrupts are sent through the Interrupt Control (ICTL) logic to the MPU. The interrupts are cleared by writing to an IBUS register address with the appropriate bit(s) set. The 60-Hz lineclock signal from the PBUS backplane is also sent to the IBUS logic as an interrupt source. This signal is enabled, recorded, and cleared in the same manner as an IBUS vector interrupt.

PBUS interrupts are sent by the IBUS logic at the request of the MPU. The MPU loads and enables a vector number in the IBUS logic. The MPU is notified via an ICTL interrupt upon successful completion of the PBUS interrupt cycle.

1.2.3 Main Processing Unit

The Main Processor Unit (MPU) is the primary source of control for the IDC. MPU software, comprising a real-time kernel and device driver(s), governs all IDC operations. The MPU communicates with other elements of the IDC and external systems (system software and peripheral devices) via memory mapped accesses and interrupts. The MPU consists of a Motorola 88100 RISC processor, separate code and data RAM (both single cycle access), and an EPROM. The processor's data interface is extended by the Local Bus (LBUS) logic to allow asynchronous read/write access to IDC status and control registers, data buffer RAM, and main memory.

The MPU kernel interacts with system software via MBS queue structures in main memory. The PBUS Interface (PBI) logic maps MPU LBUS reads and writes to physical main memory with a 32-by-1,024 RAM lookup table used to generate PBUS headers. The mapping RAM is initialized at boot time. The MPU waits on the completion of a PBUS transfer operation which, depending on ongoing PBUS activity, may take an indeterminate length of time.

MPU software via the LBUS initiates, configures, controls, and monitors IDC operations. Control registers in the IDC gate arrays and board level hardware logic are loaded with parameters that are specific to the application and individual operations. Status pertaining to IDC state and operations in progress is read and interpreted by the MPU.

The MPU device driver can send and receive information directly to and from IPI peripherals through either the Device Interface Command Execution (DICE) transfer register files, or the Data Buffer (DBUF). In both cases the information is transferred by the MPU LBUS without involving main memory.

The MPU uses MPU interrupts to service main memory queues, schedule processes, and detect external events such as operation completion and errors. The Interrupt Control (ICTL) logic enables, records, prioritizes, and asserts interrupts. Since the Motorola 88100 processor has a single interrupt line, the MPU kernel must interrogate ICTL registers, and in some cases other IDC registers, to determine the priority and source of an interrupt.

Interrupt priorities can be programmed with levels zero through seven, with zero defined as no interrupt and seven defined as the highest priority. An ICTL register field, written by the MPU, allows all interrupts below the specified level to be masked. Masked interrupts are still recorded by the ICTL and are asserted if their associated level becomes unmasked. Interrupts are acknowledged by writing to an ICTL register address with the appropriate bit(s) set.

1.2.4 Peripheral Interface

The IDC contains four independent I/O ports that implement the ANSI standard Intelligent Peripheral Interface (IPI). Each port can communicate with up to eight external devices or disk drives via a single 16-bit multiplexed interface. The physical interface configuration and protocol is performed by hardware state machines in the DICE gate array. The logical level functions are contained in MPU code. The initial code release of the IDC supports the IPI level 2 command set for magnetic disk drives.

Operations with peripherals are initiated by the MPU code by programming a set of registers in the DICE and DPED gate arrays. Opcodes, counts, and parameters written to the DICE gate array enable the execution of IPI control sequences and information transfers. Once initialized, the DICE gate array can execute a *command chain* or a complex series of operations without further intervention from the MPU. This capability allows simultaneous and independent operations on all four IDC ports. Command chain completion status is recorded and reported via DICE status registers and interrupts. Exception conditions cause the command chain to terminate while preserving error status.

Logical command and response and control information is also passed between IDC software and the attached external devices through DICE registers. The DPED gate array passes data between the IDC data buffer and the IPI external devices. The IPI port hardware does not interpret the content of IPI information except to perform integrity checks such as Error Correction Code (ECC), Cyclic Redundancy Check (CRC), parity, and transfer length.

1.2.5 Diagnostics, Test, and Clock Control

The IDC hardware incorporates several systems and logic elements to facilitate testing and fault isolation. All storage registers in the IDC gate arrays and selected board level registers are scannable from the Service Processor Unit (SPU). Since the MPU contains its own clock, it cannot be scanned directly. However, a scannable 32-bit register that can be read and written from the LBUS allows scan communication and visibility into the MPU subsystem.

Diagnostic software executing in the MPU can configure any one of the four IPI ports into *loopback* mode. In this mode, an IPI slave device can be emulated in software by reading and writing across the LBUS. This mode allows the IDC hardware to be tested at a level that would be difficult or destructive, or both, with active devices attached.

The IDC clocking requirements are relatively complex. The MPU subsystem generates its own free running 18-MHz clock from an onboard oscillator and control logic. The remainder of the IDC hardware uses a 10-MHz gated clock derived from the 20-MHz PBUS backplane clock and MPHASE and DMODE signals. The IDC gate arrays use the negative edge of the board clock during normal operations. The gate arrays use a 5-MHz clock during scan operations. The PIGA also requires a clock with an asymmetric duty cycle.

1.2.6 IDC Register Map

The register map for the Integrated Disk Channel is listed in the following table:

Table 1–1, IDC Register Map

Reference Area	First Address	Last Address
Data Memory	0x02000000	0x0203FFFF
Code Memory	0x01000000	0x0103FFFF
DIAG Registers	0x00FFB000	0x00FFB03B
FSR	0x00FFB038	0x00FFB03B
RD_HDR_LO	0x00FFB024	0x00FFB027
RD_HDR_HI	0x00FFB020	0x00FFB023
Slot_ID	0x00FFB00C	0x00FFB00F
SCR	0x00FFB008	0x00FFB00B
DCR	0x00FFB004	0x00FFB007
TRR	0x00FFB000	0x00FFB003
PBUS Map	0x00FFA000	0x00FFAFFF
PROT RAM	0x00FF9000	0x00FF93FF
IBUS RAM	0x00FF8000	0x00FF83FF
PIGA Registers	0x00FF7000	0x00FF7FFF
PARERR	0x00FF7400	0x00FF740F
Control Store	0x00FF7200	0x00FF73FF
Register File	0x00FF7000	0x00FF71FF
LOOPBACK Registers	0x00FF6000	0x00FF600F
Loopback Control	0x00FF600C	0x00FF600F
IPI Control	0x00FF6008	0x00FF600B
BUSB	0x00FF6004	0x00FF6007
BUSA	0x00FF6000	0x00FF6003
BARB Registers	0x00FF5000	0x00FF501F
ACREG3	0x00FF501C	0x00FF501F
ACREG2	0x00FF5018	0x00FF501B
ACREG1	0x00FF5014	0x00FF5017
ACREG0	0x00FF5010	0x00FF5013
APCREG3	0x00FF500C	0x00FF500F
APCREG2	0x00FF5008	0x00FF500B
APCREG1	0x00FF5004	0x00FF5007
APCREG0	0x00FF5000	0x00FF5003

Table 1–1, IDC Register Map (continued)

Reference Area	First Address	Last Address
ICTL Registers	0x00FF4000	0x00FF401B
IPREG	0x00FF4018	0x00FF401B
IBREG	0x00FF4014	0x00FF4017
ISREG	0x00FF4010	0x00FF4013
IEREG	0x00FF400C	0x00FF400F
IAREG (Write-only)	0x00FF4008	0x00FF400B
ICREG	0x00FF4004	0x00FF4007
IVREG (Read-only)	0x00FF4000	0x00FF4003
DICE & DPED Registers	0x00FF0000	0x00FF30BF
DPED3	0x00FF3080	0x00FF30BF
DICE3	0x00FF3000	0x00FF307F
DPED2	0x00FF2080	0x00FF20BF
DICE2	0x00FF2000	0x00FF207F
DPED1	0x00FF1080	0x00FF10BF
DICE1	0x00FF1000	0x00FF107F
DPED0	0x00FF0080	0x00FF00BF
DICE0	0x00FF0000	0x00FF007F
BUFFER	0x00C00000	0x00C0FFFF
Main Memory	0x00400000	0x007FFFFF
EPROM	0x00000000	0x0001FFFF

1.3 PBUS Interface Logic (PBI)

The PBUS Interface (PBI) logic is a PAL-based state machine that executes the actual PBUS signal protocol. The PBI logic requests the bus, generates data strobes, and performs error checking.

The PBI logic uses a RAM lookup table to generate PBUS headers that map MPU LBUS reads and writes to physical main memory. The PBI logic also provides partial control of the PIGA.

1.3.1 MPU PBUS Access

For MPU accesses to main memory, a 32-by-1,024 map is provided which maps 1,024 pages of MPU address space to a page in main memory or I/O space. The map contains the page frame number and access control bits for each page. The base address of the map is 0x00FFA000. The base address of the 1,024 pages mapped to main memory is 0x00400000.

All hardware for this mapping is external to the PIGA. No buffering of read or write data is provided for the MPU. Each PBUS read or write results in a new PBUS cycle. The format of the map is shown in the following figure.

Figure 1–4, PBUS Map Register for MPU

Page Frame	Reserved	Function	Valid
31 … 12	11 … 3	2 … 1	0

The four fields are defined as follows:

- Bits <31..12>—Contain the page frame number of the page mapped into memory or I/O space.
- Bits <11..3>—Reserved
- Bits <2..1>—Contain the function code. This code specifies whether the access is for memory or I/O space and the operation to be performed. The function codes are defined in the following table:

Table 1–2, MPU Map Function Codes

Code	Function
0	Memory Read/Write
1	Memory TAS
2	Memory TAC
3	I/O Read/Write

- Bit <0>—Contains the valid bit. When set, this bit enables access to the page associated with the map. When cleared, access to the associated page results in a bus error to the MPU.

1.4 PBUS Interface Gate Array (PIGA)

The PBUS Interface Gate Array (PIGA) is designed to simplify interfacing to the PBUS. The PIGA performs the following two functions:

1. Address translation
2. Data alignment

The PIGA gate array includes a microsequencer that consists of the control store logic, the branch logic, and the microword decode and register logic. The microword control bits are registered after coming out of the control store due to timing considerations.

A micro machine on the PIGA performs address translation. The microprogram resides in a 36-bit by 128-word control store RAM on the PIGA gate array. The microcode for the PIGA gate array is 32 bits wide with the remaining four bits providing parity for each microword. This micro control store RAM consumes one-fourth of the gate array.

Information required for address translation is stored in a 36-bit by 128-word register file. Each word is 32 bits wide with the remaining four bits providing parity for the each word. This 3-port

RAM consumes one-fourth of the gate array. The remaining one-half of the gate array consists of logic gates and registers.

Also involved in the address translation process is a multiplexer that is used to shift and mask data coming out the two read ports on the register file. A 32-bit Arithmetic Logic Unit (ALU) capable of addition, ANDing, and exclusive-NORing is included in the address translation data path. Parity is generated on the output of the ALU so that parity can be written back into the register file. A 32-bit accumulator, an accumulator multiplexer, and complementing logic are also included.

The alignment of data to and from the PBUS is performed in the data blender logic. The data blender logic consists of a staging register and a group of eight-to-one multiplexers. Staging register data can be held by controlling a multiplexer that recirculates the outputs of the staging register to the inputs of the staging register. This method of holding the register data is also used on the input, output, and address registers of the PIGA. All control for data alignment is handled externally to the gate array.

The PIGA also contains input and output registers. Each input or output register is a 72-bit register. These registers are configured like the staging registers in the data blender and are capable of holding their data.

All input signals to the gate array except those signals that hold data in the input and output staging and address registers on PIGA are registered before being used.

Control of the gate array is divided between the external PAL-based PBUS *sequencer (PBI)* and the internal micro machine. The micro control store is internal to the gate array and is loaded by the MPU at boot time. The gate array is not used by the IDC microprocessor during its main memory access.

1.4.1 Address Translation and Buffer Management

The PIGA gate array provides address translation and buffer management functions for the data moving between main memory and the IPI ports. These functions are handled by a microprogrammed controller that consists of the micro machine, the micro control store, and the register file.

1.4.1.1 Micro Machine

The micro machine contains the necessary program flow control and ALU logic to carry out the address translation and buffer management algorithm in the micro control store. It is enabled by a run bit in the gate array control register. Signals from the IPI interface requesting data movement trigger the micro machine into execution.

The initial part of the microprogram deals with buffer management. Each port has pointers that delimit a port's assigned area in the onboard buffer. The buffer area is treated as circular, containing a fixed number of blocks. The block size is determined by the block size register in the register file. Starting at the logical address specified, the buffer manager moves blocks between main memory and buffer memory until the transfer count is exhausted. A handshake between the micro machine and the IPI interface keeps track of the number of blocks in the buffer that are currently in use to prevent underrun or overrun conditions.

Once the size of a particular transfer is determined, the micro machine uses information from the register file to perform the address translation and obtain the physical address for the transfer. Then the micro machine initiates the actual transfer of data between the buffer and main memory.

1.4.1.2 Micro Control Store

The micro control store is loaded by the MPU at boot time with a microprogram provided by the service processor. The control store is 40 bits by 128 microinstructions and occupies one quadrant of the gate array. Because of the RAM organization in the gate array, a 40-bit microinstruction is split into two 20-bit sections. Each 20-bit section is contained in the lower bits of a 32-bit word to the MPU. The upper bits of each microinstruction are contained in an even word and the lower bits are contained in the following odd word to the MPU. The entire control store occupies 1,024 bytes of MPU address space at address 0x00FF7200. The micro control store is word read/write addressable.

1.4.1.3 Register File

The register file is a 32-bit by 128-entry three-port RAM array that occupies one quadrant of the PIGA gate array. The register file is located at address 0x00FF7000. The map of the register file is defined in Table 1–3, "Register File Map":

Table 1–3, Register File Map

Bit	Definition
96–127	Constant Registers (Accessed by S port)
64–95	Registers 8–15 for each port (Accessed by D port)
71	Block size
70	Buffer address
69	Next transaction byte count (in next page)
68	Transaction byte count (in current page)
67	Old logical address
66	Level 2 PTE value
65	Level 1 PTE value
64	SDR value
0–63	Two sets of registers 0–7 for each port (Accessed by W and D ports)
7	Scratch
6	Blender value
5	Header operation
4	SDR 7 physical address
3	SDR 0 physical address
2	Byte count
1	Logical address
0	Control and status

The RAM array contains a total of 512 bytes of registers. Since the array can be read through both of its two read ports, the array appears to occupy 1,024 bytes of MPU address space. One port occupies the lower 512 bytes of the space and the other port occupies the remaining upper 512 bytes. Although only one read port is generally used during normal operation, the MPU can verify the operation of both ports. The register file can be written at either address and is word write addressable only.

The register file contains two sets of registers for each IPI port. The LBUS addresses for these sets of registers are defined in Table 1–4, "Register Set LBUS Addresses":

Table 1–4, Register Set LBUS Addresses

LBUS Address	Description
0x00FF71C0	Port 3 register set 1
0x00FF7180	Port 3 register set 0
0x00FF7140	Port 2 register set 1
0x00FF7100	Port 2 register set 0
0x00FF70C0	Port 1 register set 1
0x00FF7080	Port 1 register set 0
0x00FF7040	Port 0 register set 1
0x00FF7000	Port 0 register set 0

Each register set holds the control bits, logical buffer address, byte count, translation information, and scratch registers for one transfer. Thus, there is a complete setup for an executing and a pending transfer for each port. At the completion of a transfer, the sequencer automatically switches sets. The addresses and sizes of the registers in a transfer register set are shown in the following table:

Table 1–5, Transfer Register Set Registers

LBUS Address	Name	Description	Size
0x00FF703C	BFSIZ	Data buffer block size	32
0x00FF7038	BUFADR	Data buffer starting address	32
0x00FF7034	XBC2	Length of second transaction	32
0x00FF7030	XBC	Transaction length	32
0x00FF702C	OLDLA	Logical address of last transaction	32
0x00FF7028	L2PTE	Level 2 PTE value	32
0x00FF7024	L1PTE	Level 1 PTE value	32
0x00FF7020	SDR	SDR value	32
0x00FF701C	T0	Temporary storage	32
0x00FF7018	BLNDR	PBUS Interface blender logic value	32
0x00FF7014	HDROP	Transaction type	32
0x00FF7010	SDR7	Physical address of SDR 7	32
0x00FF700C	SDR0	Physical address of SDR 0	32
0x00FF7008	COUNT	Bytes remaining to be transferred	32
0x00FF7004	LADDR	Logical address of current transaction	32
0x00FF7000	CSR	Control and status	32

The remaining 32 registers in the register file are used to hold constants and masks needed by the microprogram. The constants may be set up for C1 or C2 page table formats. These registers are loaded by the MPU at boot time. The constant register assignments are shown in the following table:

Table 1–6, Constant Registers

Address	Name	Description	Constant
127	hex7	Hex 7	7
126	pgsiz	Page size	4096
125	csrtpb	Two page buffer	4
124	smsk	SDR mask	0E0000000
123	sl1msk	SDR level 1 mask	0FFC00000
122	sdrvmsk	SDR validity mask	10
121	sdrmsk	SDR mask	0FFFFFE00
120	hdrop	PBUS header operation	0004F9FF
119	pomsk	Page offset mask	00000FFF
118	pfmsk	Page frame mask	0FFFFF000
117	l1l2vmsk	L1L2 validity mask	51
116	l1l2msk	L1L2 mask	0FFFFF000H
115	csrxv	Control and status translation valid	2
114	csrdone	Control and status completion	1
113	bfsiz	Buffer size	512
112	reg0	Register 0	0

1.4.2 Byte Alignment

The main memory location of the buffer for a transfer may have arbitrary byte alignment. While transfers to and from the disk buffer cache are page aligned, raw disk and tape transfers are randomly aligned. Data is stored in the onboard buffer aligned on a 64-bit boundary. The micro machine sets up byte alignment logic in the gate array to perform the necessary rotation of data to match the two different alignments.

1.4.3 Parity Checking

Parity checking is performed at five locations on the PIGA. The gate array contains three groups of parity checkers. One group checks the parity of the data in the input data register. Another group checks the parity of the data in the output data register. The third group checks the parity of the data coming out of the RAMs on the PIGA.

Three parity checkers check parity on three of the four read ports on the RAMs. A parity checker checks the parity on the S port of the control store, but parity is not checked on the D port since the port is not used. A parity checker checks parity on both the S and D ports of the register file. One signal from the PIGA indicates a parity error has been detected. All of the parity checker outputs are ORed together to produce this single signal. The parity error information on the PIGA indicates which byte of the respective bus was in error.

1.4.4 Internal Registers

The formats of the PIGA gate array internal registers are defined in the following sections.

1.4.4.1 Parity Status Register

The format of the Parity Status Register is defined in Table 1–7, "Parity Status Register Format (PARERR)":

Table 1–7, Parity Status Register Format (PARERR)

Bit	Definition
31	0
30	0
29	0
28	0
27	Parity error bit <3> for control store RAM S port
26	Parity error bit <2> for control store RAM S port
25	Parity error bit <1> for control store RAM S port
24	Parity error bit <0> for control store RAM S port
23	Parity error bit <3> for register file RAM S port
22	Parity error bit <2> for register file RAM S port
21	Parity error bit <1> for register file RAM S port
20	Parity error bit <0> for register file RAM S port
19	Parity error bit <3> for register file RAM D port
18	Parity error bit <2> for register file RAM D port
17	Parity error bit <1> for register file RAM D port
16	Parity error bit <0> for register file RAM D port
15	Parity error bit <7> for output data register
14	Parity error bit <6> for output data register
13	Parity error bit <5> for output data register
12	Parity error bit <4> for output data register
11	Parity error bit <3> for output data register
10	Parity error bit <2> for output data register
09	Parity error bit <1> for output data register
08	Parity error bit <0> for output data register
07	Parity error bit <7> for input data register
06	Parity error bit <6> for input data register
05	Parity error bit <5> for input data register
04	Parity error bit <4> for input data register
03	Parity error bit <3> for input data register
02	Parity error bit <2> for input data register
01	Parity error bit <1> for input data register
00	Parity error bit <0> for input data register

[1] The RAM parity enables should not be activated until the RAMs have been fully initialized with good parity, unless one is prepared to ignore the parity errors that will be detected. Likewise, if data with valid parity is not present in the input and output registers when those parity checkers are enabled, then errors will be recorded.

[2] The RMX<2..0> signals need to be taken to 0b100 in order to read the parity error register.

1.4.4.2 Control Store Register

The format of a Control Store Register is defined in Table 1–8, "Control Store Register Format":

Table 1–8, Control Store Register Format

Bit	Definition
39	Bit <19> of even word to MPU
38	Bit <18> of even word to MPU
37	Bit <17> of even word to MPU
36	Bit <16> of even word to MPU
35	Bit <15> of even word to MPU
34	Bit <14> of even word to MPU
33	Bit <13> of even word to MPU
32	Bit <12> of even word to MPU
31	Bit <11> of even word to MPU
30	Bit <10> of even word to MPU
29	Bit <9> of even word to MPU
28	Bit <8> of even word to MPU
27	Bit <7> of even word to MPU
26	Bit <6> of even word to MPU
25	Bit <5> of even word to MPU
24	Bit <4> of even word to MPU
23	Bit <3> of even word to MPU
22	Bit <2> of even word to MPU
21	Bit <1> of even word to MPU
20	Bit <0> of even word to MPU
19	Bit <19> of odd word to MPU
18	Bit <18> of odd word to MPU
17	Bit <17> of odd word to MPU
16	Bit <16> of odd word to MPU
15	Bit <15> of odd word to MPU
14	Bit <14> of odd word to MPU
13	Bit <13> of odd word to MPU
12	Bit <12> of odd word to MPU
11	Bit <11> of odd word to MPU
10	Bit <10> of odd word to MPU
09	Bit <9> of odd word to MPU
08	Bit <8> of odd word to MPU
07	Bit <7> of odd word to MPU
06	Bit <6> of odd word to MPU
05	Bit <5> of odd word to MPU
04	Bit <4> of odd word to MPU
03	Bit <3> of odd word to MPU
02	Bit <2> of odd word to MPU
01	Bit <1> of odd word to MPU
00	Bit <0> of odd word to MPU

1.4.4.3 Register File Registers

All register file registers except the Header Operation Register (HDROP) and the Control and Status Register (CSR) contain a 32-bit hexadecimal value. Therefore, only the formats of the HDROP and CSR registers are described. The format of the Header Operation Register is defined in Table 1–9, "Header Operation Register Format (HDROP)":

Table 1–9, Header Operation Register Format (HDROP)

Bit	Definition
31	0
30	0
29	0
28	0
27	0
26	0
25	0
24	0
23	0
22	0
21	0
20	0
19	0
18	0
17	0
16	0
15	PBUS header type field bit <47>
14	PBUS header type field bit <46>
13	PBUS header type field bit <45>
12	PBUS header type field bit <44>
11	PBUS header type field bit <43>
10	PBUS header type field bit <42>
09	PBUS header type field bit <41>
08	PBUS header type field bit <40>
07	0
06	0
05	0
04	0
03	0
02	0
01	0
00	0

The format of the Control and Status Register is defined in Table 1–10, "Control and Status Register Format (CSR)":

Table 1–10, Control and Status Register Format (CSR)

Bit	Definition
31	Reserved
30	Reserved
29	Reserved
28	Reserved
27	Reserved
26	Reserved
25	Reserved
24	Reserved
23	Reserved
22	Reserved
21	Reserved
20	Reserved
19	Reserved
18	Reserved
17	Reserved
16	Reserved
15	Error code
14	Error code
13	Error code
12	Error code
11	Error code
10	Error code
09	Error code
08	Error code
07	Reserved
06	Reserved
05	Reserved
04	Reserved
03	Reserved
02	Two page block
01	Translation valid
00	Done bit

1.5 Main Processing Unit (MPU)

The Main Processing Unit (MPU) is a microprocessor-based controller for the IDC. The MPU consists of an 88100 RISC microprocessor, two 36-bit-by-64K Static RAM (SRAM) memories, one 32-bit-by-32K EPROM memory, and interface logic that connects the 88100 to the SRAMs, the EPROM, and the IDC's local data bus. The MPU uses an 18-MHz clock that is separate from the 10-MHz clock that supplies the clock signal for the remainder of the IDC board.

1.5.1 Instruction Accesses

Instructions for the 88100 are stored in the EPROM and in one of the SRAM memories. Together, this SRAM and the EPROM are known as the *CSTORE.* The 88100 accesses the CSTORE through the Code Memory (CMEM) logic, which decodes the instruction addresses and determines which memory devices should be enabled. The CMEM also monitors the CSTORE SRAM's parity so that it can notify the 88100 of any code corruption.

CSTORE SRAM instruction accesses require two MPU clock cycles. During the first cycle, the 88100 generates an address that passes through a set of latches in the CMEM logic before it reaches the SRAM. At the end of this cycle, the instruction word from the SRAM is valid and is captured by a set of registers within the CMEM. During the second clock cycle, the instruction word is presented to the 88100's instruction memory port, and the instruction word's parity is checked. If a parity error is detected, the CMEM logic indicates the error to the microprocessor by asserting a *fault* reply to the microprocessor. Otherwise, the CMEM logic asserts a *success* reply.

In contrast to dual-cycle SRAM instruction accesses, CSTORE EPROM accesses require eleven clock cycles. Like the first cycle of an SRAM access, the 88100's instruction address passes through the latches in the CMEM logic before reaching the EPROM. However, the latches close so that the address remains stable at the EPROM during the next nine clock cycles. At the end of the tenth cycle, the instruction word from the EPROM is valid and is captured by a set of registers within the CMEM, like an SRAM access. During the eleventh and final cycle, the instruction is presented to the 88100. Parity is not checked in this case, so the CMEM logic always generates a *success* reply for a CSTORE EPROM instruction access.

Even though two cycles are required to read the CSTORE SRAM and eleven cycles are required to read the CSTORE EPROM, the 88100 receives a new SRAM instruction every cycle and a new EPROM instruction every ten cycles because the 88100's instruction accesses are pipelined. During the cycle in which the CMEM is supplying the 88100 with an instruction word, the 88100 is generating an address for the next instruction access. By overlapping instruction accesses in this manner, the access rate is increased.

Four separate CMEM state machines control the 88100's accesses to the CSTORE. These state machines work together to make sure the EPROM and the SRAM are not active at the same time and that the 88100 does not skip an instruction access while the local data bus is accessing the CSTORE.

One of these state machines, the MP state machine, monitors the instruction addresses and the states of the SRAM and the EPROM memories to determine when the accessed instruction word is valid. If the accessed word is not valid, the MP state machine asserts a *wait* signal to the code reply generator. Then the code reply generator sends a *wait* reply to the 88100. When the MP state machine detects an invalid instruction address, the state machine asserts a *fault* signal to the code reply generator. When the code reply generator detects this signal, it sends a *fault* reply to the 88100.

Another state machine, the RAM state machine, drives the control signals for the CSTORE SRAM. This state machine also keeps track of the status of the SRAMs so the MP state machine can determine when an SRAM instruction word is valid. The RAM state machine has the following three operating modes:

1. Instruction fetch
2. Local data bus read
3. Local data bus write

Each of these modes is mutually exclusive and is controlled by signals from the MP state machine and the LBUS state machine.

The third state machine, the EPROM state machine, controls the CSTORE EPROM. Like the RAM state machine, this state machine monitors the status of the EPROM and reports the status to the MP state machine. The EPROM state machine has the following two mutually exclusive modes of operation:

1. Instruction fetch
2. Local data bus read

The fourth state machine in the CMEM, the LBUS state machine, controls accesses of the CSTORE SRAM and EPROM by the local data bus. When a device on the local data bus wants to read or write the SRAM or EPROM, the LBUS state machine notifies the RAM and the EPROM state machines by asserting an *LBUS access* signal. When this signal is detected, the RAM and EPROM state machines turn off the memory devices they are controlling. Then these state machines notify the LBUS state machine that they are ready by asserting *RAM off* and *EPROM off* signals. The LBUS state machine then communicates with the appropriate state machine until the desired access is complete.

For example, in the case of an SRAM write, the LBUS state machine asserts a *LBUS RAM access* signal. This signal, along with a *LBUS read* signal, informs the RAM state machine that it needs to enter the local data bus write mode. When the RAM machine has completed the write, it asserts a *RAM acknowledge* signal. The LBUS machine detects this signal and acknowledges to the local data bus that the access has completed. When this acknowledge has been received by the local data bus, the LBUS state machine deasserts the *LBUS access* signal which allows the RAM and CMEM EPROM machines to continue accessing instructions for the 88100.

1.5.2 Data Accesses

The 88100 microprocessor can access data from many different areas. One such area is a 36-bit-by-64K SRAM within the MPU which the 88100 uses for general storage. This Static RAM (SRAM), known as the *DSTORE,* is accessible through the Data Memory (DMEM) logic. The 88100 also uses special-purpose registers on the IDC to control the operation of the board. These registers are accessible through the LBUS logic. The CSTORE EPROM and SRAM are also available through the LBUS logic. In addition to these local data sources, the 88100 can also access data stored in the machine's main memory. Accesses to main memory are also made through the LBUS logic.

The DMEM logic allows the 88100 to access the DSTORE SRAM. The DMEM logic also initiates and monitors the progress of any accesses requiring the LBUS logic. The DMEM logic performs these functions by monitoring the address, byte strobe, and read/write signals from the 88100's data memory port. Then the DMEM logic determines which part of the IDC needs to be accessed. If the 88100 needs to read or write the DSTORE SRAM, the DMEM logic generates the necessary control signals for the SRAM devices. If an LBUS access is needed, the DMEM logic asserts an ***external cycle*** signal to the LBUS logic. When the LBUS has finished its access, it asserts a ***reply*** signal to the DMEM logic.

At the completion of every data access, the DMEM logic notifies the 88100 by asserting a completion code on a *reply* signal to the 88100. If the microprocessor is reading data, the DMEM logic also checks the parity of the incoming data to ensure that the 88100 is aware of any corrupted data that it may receive.

Each individual DSTORE SRAM access requires two clock cycles. During the first cycle, the 88100 generates a data address which passes through a set of latches in the DMEM logic before it reaches the SRAM. At the end of this cycle, if the access is a data read, the data word from the SRAM is valid and is captured by a set of registered transceivers within the DMEM logic.

If the access is a data write access, the data from the 88100 is captured at the end of this first cycle by the same registered transceivers. During the second clock cycle, the data word is presented to the 88100's data memory port if the access is a read access, or to the SRAM if the access is a write access. Parity is also checked and generated during the second clock cycle. If a parity error is detected for a read access, the DMEM logic indicates the error by asserting a ***fault*** reply to the microprocessor. Otherwise, a ***success*** reply is generated during the second clock cycle.

Even though two cycles are required to access the DSTORE SRAM, the 88100 can start a new SRAM data access every cycle because the 88100's data accesses are pipelined. During the cycle in which the DMEM logic is supplying the 88100 with a data word or writing a data word into the DSTORE SRAM, the 88100 is generating an address for the next data access. By overlapping data accesses in this manner, the access rate is increased.

Most DSTORE SRAM accesses require two clock cycles as described above. However, if the 88100 performs a DSTORE SRAM read access immediately after a DSTORE SRAM write access, the start of the read access will be delayed by one clock cycle. This delay is necessary because the first cycle of the read in which the data word is read from the SRAM, overlaps the second cycle of the write in which the data word is written to the SRAM. Since both operations cannot be performed simultaneously, the read access is delayed by one clock cycle. When this happens, the latches through which the data address passes, are closed so that the address can remain valid until the SRAM data has been read.

Control of the DMEM logic is achieved through the use of the DMEM state machine. This state machine operates in the following two modes:

1. DSTORE SRAM access
2. LBUS (or external) access

In the DSTORE SRAM mode, the DMEM state machine determines when SRAM accesses begin and end. In the LBUS mode, the state machine initiates LBUS accesses and then waits for a ***reply*** signal from the LBUS logic. In either mode, when the DMEM state machine detects that an access has completed, it signals to the data reply generator that an access has completed. Then the data reply generator asserts a ***reply*** signal to the 88100.

The data reply generator monitors signals from the DMEM state machine, signals from the LBUS logic, the DSTORE SRAM protection bits, and the parity checking logic. The data reply generator uses this information to determine whether an 88100 data access has completed, and if the access has completed, whether the access was successful. Then the data reply generator uses a *reply* signal to report this access status to the 88100.

The LBUS logic provides a path from the 88100's data memory port to the IDC's local data bus. The LBUS logic allows the 88100 to access the MPU's CSTORE SRAM and EPROM registers which control the IDC's operation and the PBUS interface which communicates with the machine's main memory. The LBUS logic consists of the following:

- Two sets of registered transceivers for the transfer of data words to and from the local data bus
- A set of registers to hold the address for a data transfer
- Decoding PALs that use the address to select devices on the local data bus
- PALs that acknowledge the completion of an access

As described above, the LBUS logic begins an access when it receives an *external cycle* signal from the DMEM logic. The LBUS logic signals the end of a transfer by asserting an *external reply* signal to the DMEM logic.

1.6 Buffer Arbitration Logic (BARB)

The Buffer Arbitration (BARB) logic coordinates accesses to the four I/O port data buffers. The PIGA and DPED gate arrays access a buffer through separate dedicated ports which operate independently. The BARB logic is responsible for monitoring buffer accesses on a block basis and detect buffer empty and buffer full conditions. The BARB logic has four identical sections; one section for each port. Each section contains an up/down counter that is incremented or decremented in response to signals from the DPED and PIGA gate arrays. The counter value for each port is available to the MPU in the four Arbiter Counter Registers (ACREG).

Buffer access is granted to the DPED and PIGA gate arrays on a per block basis based on *buffer not full* and/or *buffer not empty* conditions. The buffer size (in blocks) is specified in the Arbiter Port Control Register (APCREG). The information in the APCREG is also used by the BARB logic to configure the PIGA and PBI logic (select port and register set, specify transfer direction and enable operation). Read-only status bits in the APCREG indicate to the MPU whether the associated PIGA port is currently active and/or if an operation is queued awaiting execution.

The BARB logic also monitors PIGA gate array status and forwards completion and error signals to the appropriate DICE gate array which in turn may, if enabled, generate an ICTL interrupt.

1.6.1 Buffer Arbitration Registers

The formats of the registers in the Buffer Arbitration (BARB) logic are listed in this section. The format of the Arbiter Counter Register is defined in Table 1–11, "Arbiter Counter Register Format (ACREG)":

Table 1–11, Arbiter Counter Register Format (ACREG)

Bit	Definition
31	Reserved
30	Reserved
29	Reserved
28	Reserved
27	Reserved
26	Reserved
25	Reserved
24	Reserved
23	Reserved
22	Reserved
21	Reserved
20	Reserved
19	Reserved
18	Reserved
17	Reserved
16	Reserved
15	Reserved
14	Reserved
13	Reserved
12	Reserved
11	Reserved
10	Reserved
09	Reserved
08	Reserved
07	Reserved
06	Reserved
05	Buffer block count bit <5>
04	Buffer block count bit <4>
03	Buffer block count bit <3>
02	Buffer block count bit <2>
01	Buffer block count bit <1>
00	Buffer block count bit <0>

The format of the Arbiter Port Control Register is defined in Table 1–12, "Arbiter Port Control Register Format (APCREG)":

Table 1–12, Arbiter Port Control Register Format (APCREG)

Bit	Definition
31	Reserved
30	Reserved
29	Reserved
28	Reserved
27	Reserved
26	Reserved
25	Reserved
24	PIGA register set 0 valid
23	Reserved
22	Reserved
21	Reserved
20	Reserved
19	Reserved
18	Reserved
17	Reserved
16	PIGA register set 1 valid
15	Reserved
14	Reserved
13	Reserved
12	Reserved
11	Reserved
10	Reserved
09	Reserved
08	PIGA register select
07	Reserved
06	Reserved
05	PIGA port currently active
04	PIGA port request pending
03	Port I/O read operation
02	Port buffer size code bit <2>[1]
01	Port buffer size code bit <1>[1]
00	Port buffer size code bit <0>[1]

[1] This field specifies the number of blocks in the port data buffer as follows:

7 = 0 blocks	6 = 63 blocks
5 = 31 blocks	4 = 15 blocks
3 = 7 blocks	2 = 3 blocks
1 = 1 blocks	0 = unlimited

1.7 Diagnostic Logic (DIAG)

The Diagnostic (DIAG) logic performs a variety of functions associated with IDC system operations. The Diagnostic Control Register (DCR) is used to reset, enable, and disable the IDC's LSI parts and other logic elements. The DCR register is read/write scannable from the SPU and also accessible via the IDC LBUS. At powerup, the DCR register is cleared which serves to place the IDC hardware in a reset state. The IDC may then be made operational by the SPU and/or the MPU.

The DIAG logic contains the IDC system scan interface and control logic. The scan ring is configured and enabled according to scan direction and the assertion of the DMODE signal. The IDC gate arrays can only be scanned in one direction and are scanned at half speed (5 MHz). As a result, the scan ring is different in either direction which complicates the control logic.

The IDC COP chip resides in the DIAG block. This chip is located at the least significant end of the scan ring. The COP chip contains IDC ID and revision information that is read by the SPU during system configuration. A LBUS register that drives LEDs on the foreplane edge of the IDC is written by MPU software to indicate IDC status in real time. The bits of this register are also scannable.

The DIAG block contains a single 32-bit Scan Communication Register (SCR) that can be scanned by the SPU and can be read or written by the MPU. This register does not have a specific format and does not directly control any hardware. This register is used as a general communication path between IDC software and the SPU. The SCR register allows the SPU to indirectly access IDC register and memory locations that are not otherwise scannable.

The slot ID and system type codes are addressed and connected to the LBUS in the DIAG logic. This information is read and used by the IDC kernel software.

1.7.1 Diagnostic Registers

The formats of the registers in the Diagnostic (DIAG) logic are listed in this section. The format of the Fault Source Register is defined in Table 1–13, "Fault Source Register Format (FSR)":

Table 1–13, Fault Source Register Format (FSR)

Bit	Definition
31	Reserved
30	Reserved
29	Reserved
28	Invalid code address
27	Code byte 0 parity error
26	Code byte 1 parity error
25	Code byte 2 parity error
24	Code byte 3 parity error
23	Reserved
22	Reserved
21	Reserved
20	Reserved
19	Reserved
18	PBUS memory error fault
17	PBUS bus error fault
16	Local bus time-out
15	Port 3 DPED fault
14	Port 3 DICE fault
13	Port 2 DPED fault
12	Port 2 DICE fault
11	Port 1 DPED fault
10	Port 1 DICE fault
09	Port 0 DPED fault
08	Port 0 DICE fault
07	Data byte 0 parity error
06	Data byte 1 parity error
05	Data byte 2 parity error
04	Data byte 3 parity error
03	Reserved
02	Local bus data access fault
01	Data RAM read protection violation
00	Data RAM write protection violation

[1] Bits <31..24> are valid only if a CODE fault is detected by the 88100

[2] Bits <23..0> are valid only if a DATA fault is detected by the 88100

[3] Bits <23..8> are valid only if bit <2> is set (local bus data fault)

[4] Any combination of bits <7..4> can be set when a data parity error is detected

[5] Any combination of bits <27..24> can be set when a code parity error is detected

The format of the Read Header Low Register is defined in Table 1–14, "Read Header Low Register Format (RD_HDR_LO)":

Table 1–14, Read Header Low Register Format (RD_HDR_LO)

Bit	Definition
31	PBUS header bit <31>
30	PBUS header bit <30>
29	PBUS header bit <29>
28	PBUS header bit <28>
27	PBUS header bit <27>
26	PBUS header bit <26>
25	PBUS header bit <25>
24	PBUS header bit <24>
23	PBUS header bit <23>
22	PBUS header bit <22>
21	PBUS header bit <21>
20	PBUS header bit <20>
19	PBUS header bit <19>
18	PBUS header bit <18>
17	PBUS header bit <17>
16	PBUS header bit <16>
15	PBUS header bit <15>
14	PBUS header bit <14>
13	PBUS header bit <13>
12	PBUS header bit <12>
11	PBUS header bit <11>
10	PBUS header bit <10>
09	PBUS header bit <9>
08	PBUS header bit <8>
07	PBUS header bit <7>
06	PBUS header bit <6>
05	PBUS header bit <5>
04	PBUS header bit <4>
03	PBUS header bit <3>
02	PBUS header bit <2>
01	PBUS header bit <1>
00	PBUS header bit <0>

The format of the Read Header Hi Register is defined in Table 1-15, "Read Header Hi Register Format (RD_HDR_HI)":

Table 1-15, Read Header Hi Register Format (RD_HDR_HI)

Bit	Definition
31	PBUS header bit <63>
30	PBUS header bit <62>
29	PBUS header bit <61>
28	PBUS header bit <60>
27	PBUS header bit <59>
26	PBUS header bit <58>
25	PBUS header bit <57>
24	PBUS header bit <56>
23	PBUS header bit <55>
22	PBUS header bit <54>
21	PBUS header bit <53>
20	PBUS header bit <52>
19	PBUS header bit <51>
18	PBUS header bit <50>
17	PBUS header bit <49>
16	PBUS header bit <48>
15	PBUS header bit <47>
14	PBUS header bit <46>
13	PBUS header bit <45>
12	PBUS header bit <44>
11	PBUS header bit <43>
10	PBUS header bit <42>
09	PBUS header bit <41>
08	PBUS header bit <40>
07	PBUS header bit <39>
06	PBUS header bit <38>
05	PBUS header bit <37>
04	PBUS header bit <36>
03	PBUS header bit <35>
02	PBUS header bit <34>
01	PBUS header bit <33>
00	PBUS header bit <32>

The format of the Slot ID Register is defined in Table 1–16, "Slot ID Register Format (Slot_ID)":

Table 1–16, Slot ID Register Format (Slot_ID)

Bit	Definition
31	Reserved
30	Reserved
29	Reserved
28	Reserved
27	Reserved
26	Reserved
25	Reserved
24	Reserved
23	Reserved
22	Reserved
21	Reserved
20	Reserved
19	Reserved
18	Reserved
17	Reserved
16	Reserved
15	Reserved
14	Reserved
13	Reserved
12	Reserved
11	System type bit
10	System type bit
09	System type bit
08	System type bit
07	CCU slot ID bit <7>
06	CCU slot ID bit <6>
05	CCU slot ID bit <5>
04	CCU slot ID bit <4>
03	CCU slot ID bit <3>
02	CCU slot ID bit <2>
01	CCU slot ID bit <1>
00	CCU slot ID bit <0>

The format of the Scan Control Register is defined in Table 1–17, "Scan Control Register Format (SCR)":

Table 1–17, Scan Communication Register Format (SCR)

Bit	Definition
31	General purpose register bit <31>
30	General purpose register bit <30>
29	General purpose register bit <29>
28	General purpose register bit <28>
27	General purpose register bit <27>
26	General purpose register bit <26>
25	General purpose register bit <25>
24	General purpose register bit <24>
23	General purpose register bit <23>
22	General purpose register bit <22>
21	General purpose register bit <21>
20	General purpose register bit <20>
19	General purpose register bit <19>
18	General purpose register bit <18>
17	General purpose register bit <17>
16	General purpose register bit <16>
15	General purpose register bit <15>
14	General purpose register bit <14>
13	General purpose register bit <13>
12	General purpose register bit <12>
11	General purpose register bit <11>
10	General purpose register bit <10>
09	General purpose register bit <9>
08	General purpose register bit <8>
07	General purpose register bit <7>
06	General purpose register bit <6>
05	General purpose register bit <5>
04	General purpose register bit <4>
03	General purpose register bit <3>
02	General purpose register bit <2>
01	General purpose register bit <1>
00	General purpose register bit <00>

The format of the Diagnostic Control Register is defined in Table 1–18, "Diagnostic Control Register Format (DCR)":

Table 1–18, Diagnostic Control Register Format (DCR)

Bit	Definition	Access
31	Reserved	-
30	Reserved	-
29	Reserved	-
28	Reserved	-
27	Reserved	-
26	Reserved	-
25	Reserved	-
24	Reserved	-
23	Upper MPU clock enable (active high)	R
22	PBUS test header ready (active high)	R
21	EEPROM write enable (active high)	R
20	Code memory protection enable (active high)	R/W
19	Reserved	R/W
18	LBUS parity checking disable (active high)	R/W
17	PBUS interrupt logic reset (active low)	R/W
16	88100 reset (active low)	R/W
15	Port 3 DPED reset (active low)	R/W
14	Port 3 DICE reset (active low)	R/W
13	Port 2 DPED reset (active low)	R/W
12	Port 2 DICE reset (active low)	R/W
11	Port 1 DPED reset (active low)	R/W
10	Port 1 DICE reset (active low)	R/W
09	Port 0 DPED reset (active low)	R/W
08	Port 0 DICE reset (active low)	R/W
07	ERR-TO-SPU signal (active high)	R/W
06	Dead-man timer enable (active high)	R/W
05	Write bad parity (active high)	R/W
04	Data memory protection reset (active high)	R/W
03	Data memory protection enable (active high)	R/W
02	Buffer arbiter enable (active high)	R/W
01	PIGA run (active high)	R/W
00	PBUS test (active high)	R/W

The format of the Test Result Register is defined in Table 1–19, "Test Result Register Format (TRR)":

Table 1–19, Test Result Register Format (TRR)

Bit	Definition
31	Reserved
30	Reserved
29	Reserved
28	Reserved
27	Reserved
26	Reserved
25	Reserved
24	Reserved
23	Reserved
22	Reserved
21	Reserved
20	Reserved
19	Reserved
18	Reserved
17	Reserved
16	Reserved
15	Reserved
14	Reserved
13	Reserved
12	Reserved
11	Reserved
10	Reserved
09	Reserved
08	Reserved
07	Reserved
06	Reserved
05	Reserved
04	Reserved
03	LED bit
02	LED bit
01	LED bit
00	LED bit

1.8 Interrupt Control Logic (ICTL)

The Interrupt Control (ICTL) logic controls MPU interrupts. Since the 88100 microprocessor has only a single interrupt signal that is enabled or disabled internally by software, external hardware must be used to detect, prioritize, report, and clear IDC interrupts.

The ICTL logic contains seven interrupt ports that are divided into two types. Ports seven through four are associated with operations on I/O channels three through zero respectively. The DICE gate array for each channel inputs a 3-bit encoded signal (ILVL<2..0>) to the ICTL logic. This signal specifies the interrupt level currently being asserted by the channel. ICTL ports three through one each have four independent interrupt input signals or types.

The port three signals are connected to the interrupts signals from the four IBUS interrupt groups. Port two is used for miscellaneous board level interrupts (line clock, PBUS interrupt sent, and two reserved). Port one is used for error interrupts (parity error and three reserved). The interrupt level associated with ports one through three is specified in the Interrupt Control Register (ICREG) which is written by the MPU over the LBUS. This register also defines the interrupt mask level. All interrupts less than or equal to the interrupt mask value are not asserted to the MPU.

The Interrupt Vector Register (IVREG) records status information pertaining to the pending interrupt with the highest priority. This register contains the interrupt level (7–0), the port number (7–0), and the interrupt type (0 for ports 7–4, 3–0 for ports 3–0). Channel interrupts (ports 7–4) are acknowledged and cleared within the DICE gate array from which they were asserted. Board interrupts (ports 3–0) are acknowledged by writing to the Interrupt Acknowledge Register (IAREG) with the appropriate bit(s) set.

1.8.1 Interrupt Control Registers

The formats of the registers in the Interrupt Control (ICTL) logic are listed in this section. The format of the Interrupt Pending Register is defined in Table 1–20, "Interrupt Pending Register Format (IPREG)":

Table 1–20, Interrupt Pending Register Format (IPREG)

Bit	Definition
31	Reserved
30	Reserved
29	Reserved
28	Reserved
27	Reserved
26	Reserved
25	Reserved
24	Reserved
23	Reserved
22	Reserved
21	Reserved
20	Reserved
19	Reserved
18	Reserved
17	Reserved
16	Reserved
15	Reserved
14	Reserved
13	Reserved
12	Reserved
11	Reserved
10	Reserved
09	Reserved
08	PBUS interrupt send enable
07	Reserved
06	Reserved
05	Reserved
04	Reserved
03	Reserved
02	Reserved
01	Reserved
00	Reserved

The format of the Interrupt Send Register is defined in Table 1–21, "Interrupt Send Register Format (IBREG)":

Table 1–21, Interrupt Send Register Format (IBREG)

Bit	Definition
31	Reserved
30	Reserved
29	Reserved
28	Reserved
27	Reserved
26	Reserved
25	Reserved
24	Reserved
23	Reserved
22	Reserved
21	Reserved
20	Reserved
19	Reserved
18	Reserved
17	Reserved
16	Reserved
15	Reserved
14	Reserved
13	Reserved
12	Reserved
11	Reserved
10	Reserved
09	Reserved
08	Reserved
07	IBUS interrupt channel bit <7>
06	IBUS interrupt channel bit <6>
05	IBUS interrupt channel bit <5>
04	IBUS interrupt channel bit <4>
03	IBUS interrupt channel bit <3>
02	IBUS interrupt channel bit <2>
01	IBUS interrupt channel bit <1>
00	IBUS interrupt channel bit <0>

The format of the Interrupt Status Register is defined in Table 1–22, "Interrupt Status Register Format (ISREG)":

Table 1–22, Interrupt Status Register Format (ISREG)

Bit	Definition
31	Reserved
30	Reserved
29	Reserved
28	Reserved
27	Reserved
26	Reserved
25	Reserved
24	Reserved
23	Reserved
22	Reserved
21	Reserved
20	Reserved
19	Reserved
18	Reserved
17	Reserved
16	Reserved
15	IBUS group 3 interrupt
14	IBUS group 2 interrupt
13	IBUS group 1 interrupt
12	IBUS group 0 interrupt
11	Reserved
10	Reserved
09	Line clock interrupt
08	Interrupt sent-interrupt
07	PIGA parity error
06	Reserved
05	Reserved
04	Reserved
03	Reserved
02	Reserved
01	Reserved
00	Reserved

The format of the Interrupt Enable Register is defined in Table 1–23, "Interrupt Enable Register Format (IEREG)":

Table 1–23, Interrupt Enable Register Format (IEREG)

Bit	Definition
31	Reserved
30	Reserved
29	Reserved
28	Reserved
27	Reserved
26	Reserved
25	Reserved
24	Reserved
23	Reserved
22	Reserved
21	Reserved
20	Reserved
19	Reserved
18	Reserved
17	Reserved
16	Reserved
15	IBUS group 3 interrupt enable
14	IBUS group 2 interrupt enable
13	IBUS group 1 interrupt enable
12	IBUS group 0 interrupt enable
11	Reserved
10	Reserved
09	Line clock interrupt enable
08	Interrupt sent-interrupt enable
07	PIGA parity error enable
06	Reserved
05	Reserved
04	Reserved
03	Reserved
02	Reserved
01	Reserved
00	Reserved

The format of the Interrupt Acknowledge Register is defined in Table 1–24, "Interrupt Acknowledge Register Format (IAREG)":

Table 1–24, Interrupt Acknowledge Register Format (IAREG)

Bit	Definition
31	Reserved
30	Reserved
29	Reserved
28	Reserved
27	Reserved
26	Reserved
25	Reserved
24	Reserved
23	Reserved
22	Reserved
21	Reserved
20	Reserved
19	Reserved
18	Reserved
17	Reserved
16	Reserved
15	Port 3/type 3 interrupt acknowledge (PBUS interrupt group 3)
14	Port 3/type 2 interrupt acknowledge (PBUS interrupt group 2)
13	Port 3/type 1 interrupt acknowledge (PBUS interrupt group 1)
12	Port 3/type 0 interrupt acknowledge (PBUS interrupt group 0)
11	Port 2/type 3 interrupt acknowledge (Reserved)
10	Port 2/type 2 interrupt acknowledge (Reserved)
09	Port 2/type 1 interrupt acknowledge (Line clock interrupt)
08	Port 2/type 0 interrupt acknowledge (PBUS interrupt sent)
07	Port 1/type 3 interrupt acknowledge (PIGA parity error)
06	Port 1/type 2 interrupt acknowledge (Reserved)
05	Port 1/type 1 interrupt acknowledge (Reserved)
04	Port 1/type 0 interrupt acknowledge (Reserved)
03	Reserved
02	Reserved
01	Reserved
00	Reserved

The format of the Interrupt Control Register is defined in Table 1–25, "Interrupt Control Register Format (ICREG)":

Table 1–25, Interrupt Control Register Format (ICREG)

Bit	Definition
31	Reserved
30	Reserved
29	Reserved
28	Reserved
27	Reserved
26	Reserved
25	Reserved
24	Reserved
23	Reserved
22	Reserved
21	Reserved
20	Reserved
19	Reserved
18	Reserved
17	Reserved
16	Reserved
15	Reserved
14	Reserved
13	Reserved
11	Interrupt mask level bit <2>
10	Interrupt mask level bit <1>
09	Interrupt mask level bit <0>
08	Port 3 interrupt level bit <2>
07	Port 3 interrupt level bit <1>
06	Port 3 interrupt level bit <0>
05	Port 2 interrupt level bit <2>
04	Port 2 interrupt level bit <1>
03	Port 2 interrupt level bit <0>
02	Port 1 interrupt level bit <2>
01	Port 1 interrupt level bit <1>
00	Port 1 interrupt level bit <0>

The format of the Interrupt Vector Register is defined in Table 1–26, "Interrupt Vector Register Format (IVREG)":

Table 1–26, Interrupt Vector Register Format (IVREG)

Bit	Definition
31	Reserved
30	Reserved
29	Reserved
28	Reserved
27	Reserved
26	Reserved
25	Reserved
24	Reserved
23	Reserved
22	Reserved
21	Reserved
20	Reserved
19	Reserved
18	Reserved
17	Reserved
16	Reserved
15	Reserved
14	Reserved
13	Reserved
12	Reserved
11	Reserved
10	Reserved
09	Reserved
08	Reserved
07	Interrupt level bit <2>
06	Interrupt level bit <1>
05	Interrupt level bit <0>
04	Interrupt port bit <2>[1]
03	Interrupt port bit <1>[1]
02	Interrupt port bit <0>[1]
01	Interrupt type bit <1>[2]
00	Interrupt type bit <0>[2]

[1] Interrupt Port
Port 7–4: IPI/PBUS Channels 3–0
(channel number is subfield)
Port 3: System Interrupt Bus
Port 2: General Interrupt
Port 1: Error Interrupt
Port 0: Unused

[2] Interrupt Type (depends on the port number)
Type 7–4: 00 Ignored
Type 3: XX PBUS Interrupt XX
Type 2: XX General Interrupt XX
Type 1: XX Error Interrupt XX

1.9 Device Interface Command Execution Gate Array (DICE)

The Device Interface Command Execution (DICE) gate array sequences and executes control and transfer operations between the IDC and the peripheral devices. The device interface implemented by the DICE gate array is the Intelligent Peripheral Interface (IPI) physical level protocol as defined by the ANSI Standard X3.129-1986. DICE operations are defined and controlled by the contents of various internal registers that are accessible via a 32-bit memory mapped interface (MPU interface) to the MPU local data bus.

Once initiated, DICE operations proceed without further interaction with the MPU. The DICE gate array functions as the IPI master during protocol sequences with IPI slave devices or controllers. The gate array also configures and controls the electrical interface transceivers and the data path logic. Command completion, exception conditions, and the occurrence of external events are recorded by the DICE gate array and reported to the MPU via status registers and interrupts.

The DICE gate array is designed to be independent of the IPI logical interface (command set). The gate array does not generate or interpret IPI control octets, and operations do not depend on the content of information transfers. As a result, the IDC can support any and all IPI peripherals by adding software enhancements (device drivers).

1.9.1 MPU Interface

The Main Processing Unit (MPU) interface consists of memory mapped registers and interrupt signals. Refer to Table 1–30, "DICE Internal Registers," for a listing of the DICE internal registers.

The DICE internal registers are read and written asynchronously by the MPU. The access protocol begins with the assertion of a 5-bit register address, read/write indicator, and a select (SEL) signal. The DICE gate array responds with either an acknowledge (ACK) or fault (FLT) signal that remains asserted until the SEL signal is negated. Write data is asserted with the SEL signal and must be stable until the ACK signal is received. Read data is stable when the ACK signal is asserted and remains stable until the SEL signal is negated. The DICE gate array will respond to the SEL signal with a FLT signal if a nonpresent register address is received, a write is attempted to a read-only register, or a data parity error is detected.

The DICE interrupt signals continuously present the highest pending interrupt priority and type to the IDC interrupt logic via external pins. The interrupt priorities are programmable via register fields.

1.9.2 Operations

The primary function of the DICE gate array is to transfer logical information between the MPU software (device driver) and the IPI slave within the IPI level-one protocol. The DICE gate array does not generate or interpret logical level information.

The DICE gate array directly executes IPI physical level (level 1) control sequences and information transfer operations. Logical level operations (IPI levels 2 and/or 3) are defined, initiated, and controlled by control octets in the Command Registers (CREGs) and command and response information in the Header Registers (HREGs) and Transfer Registers (XREGs).

1.9.2.1 Commands

Each CREG contains a 3-bit opcode field that identifies one of eight commands or IPI operations to be performed. The CREG also contains a 3-bit opcode modifier field. This modifier field is presented to DICE output signal pins while CREG execution is active. These signals are intended to be used to control external hardware. Internal DICE operations are affected by the modifier bits. The opcodes and the associated functions are described in the ANSI specification X3.129-1986 IPI-1.

Idle (0)—This command initiates an immediate deselect sequence on the IPI interface regardless of the present state. The DICE gate array will interrupt at the command completion level when and if the interface is in the *Idle* state. Failure to reach the *Idle* state (nonresponding slave) will result in a control exception interrupt when the programmable timeout period expires. The opcode modifier field is reserved and must be zero. The CREG Bus A octet field is not used and is ignored.

Master Reset (1)—This command initiates a global reset of the IPI interface. This reset does not require a handshake with the slaves, and a Bus B octet is not returned. The DICE gate array drives the interface to the *Maint* state and then to the *Idle* state. State timing is determined by the DICE programmable timer. The type of reset is determined by the Bus A octet in the CREG. If the *data out* bits are set, the attached slaves reset their logical interfaces and remain released. If the *data out* bits are not asserted, the IPI interface returns to the *Idle* state and normal operations are resumed.

Selective Reset (2)—This command resets the individual slave addressed within the Bus A octet field of the CREG. This sequence may be executed at any time from an *Idle* state to return a released slave to operational status following a *Master Reset.* This sequence does not involve a response from the slave, and a Bus B octet is not returned. State timing is determined by the DICE programmable timer.

Request Interrupts (3)—This command polls all attached slaves regarding their status and conditions as specified in the CREG Bus A octet field. Slaves respond by asserting their bit-significant address on Bus B according to the logical OR of the conditions polled. The state of Bus B is latched by the DICE gate array when the programmable timer value is reached.

Request Status (4)—This command initiates a request transfer settings sequence or a request slave status sequence depending on the format and content of the CREG Bus A octet field. The state sequence is the same, but the Bus B octet returned by the addressed slave differs. The programmable timer is used as a timeout error value.

Selection (5)—This command initiates a selection sequence that establishes a virtual circuit connection between the IDC and the slave addressed by the Bus A octet. The interface does not return to the *Idle* state upon completion of this operation but remains in the *Select* state. The selected slave responds by asserting its bit-significant address on Bus B. The state of Bus B is latched into the Execution Register (EXREG) at the end of the sequence. The successful execution of a selection sequence (the bit-significant address of the selected slave, and no other, set in the Bus B octet) is verified by MPU software prior to the initiation of IPI information transfers.

Output Bus Exchange (6)—This command initiates an IPI bus control sequence, an optional information transfer, and an ending status sequence. The bus exchange is executed between the DICE gate array and a previously selected slave. The operation is specified in the CREG Bus A octet field, and the slave completion status is recorded in the EXREG Bus B octet field. An *Output Bus Exchange* command is used to transfer either IPI commands or data. The command set and data controls are defined at the logical interface.

Input Bus Exchange (7)—This command initiates an IPI bus control sequence, an optional information transfer, and an ending status sequence. The bus exchange is executed between the DICE gate array and a previously selected slave. The operation is specified in the CREG Bus A octet field, and slave completion status is recorded in the EXREG Bus B octet field. An *Input Bus Exchange* command is used to transfer either IPI responses or data. The slave response format and data controls are defined at the logical interface.

The IPI control octets for Bus A and Bus B are listed in Table 1–27, "IPI Control Octets":

Table 1–27, IPI Control Octets

Opcode	Bus A Octet	Bus B Octet
0	None	None
1	Data out	None
2	Selective reset	None
3	Request interrupts	Radial address
4	Request status/Xfr settings	Slave status/Xfr settings
5	Selection	Radial address
6	Bus control (out)	Slave ending status
7	Bus control (in)	Slave ending status

1.9.2.2 Command Execution

Once programmed, the DICE gate array is capable of executing a series of commands without processor intervention. A group of consecutive CREG locations, beginning at CREG[0] and ending with the CREG indicated by the command list end pointer field of the Operational Parameter Register (OPREG), is referred to as a command list. Each CREG command may be executed multiple times, from 0 to 255, as specified in the CREG repetition count field. Each repetition of a command is known as a *command instance.* A *command chain* consists of all of the command instances in a command list.

Command execution begins when the OPREG command list valid bit is set. Execution begins with instances one of CREG[0] and proceeds until the command chain completes or an error is encountered. All instances of a CREG are executed before advancing to the next command.

The *Idle, Master Reset, Selective Reset, Request Interrupts, Request Status,* and *Selection* commands are considered control operations. These commands are used to configure the IPI interface (resets, selection or deselection) or determine its state (request sequences). Control sequence commands may be chained, however, since the DICE gate array can only report the outcome (the Bus B octet) of these operations individually, they typically are not chained.

Transfer commands (opcodes 6 and 7) are used to communicate logical level command and response information and to transfer data. Each transfer command (IPI bus exchange) may involve 0, 1, or 2 transfer fields. The fields are selected and enabled by bits <22..20> of the CREGs. One of the two DICE internal fields (either the register transfer field or the header transfer field, but not both) and/or the external transfer field may be selected. The length of the transfer fields is defined in the Transfer Count Register (CREG). The internal field transfer precedes the external transfer field if both are enabled.

The Field Configuration Register (FCREG) contains pointers that indicate the position in the HREG and Transfer Register (XREG) of the beginning of the data to be transferred. The position is given as an offset, in a 16-bit halfword, from the most significant halfword (bits <31..16>) of HREG/XREG[0]. The FCREG also enables various transfer functions associated with field transfers. The FCREG also enables exceptions from the external transfer data path hardware, CRC for internal fields, and specifies which bytes of the HREGs are to be compared with data input to the XREGs during header input field transfers.

The HREG[3] consists of four loadable, cascadable, one-byte upcounters. The contents of these counters are used as header field output data and as comparison values for header input transfers like the other HREGs. In addition, however, the values of the four bytes of HREG[3] can be individually incremented once per command instance as specified by bits <19..16> of CREG. The HREG[3] counters are cascadable under the control of FCREG <26..24>. Maximum counter values are specified by Header Counter Control Register (HCREG) and incrementing any of the counters that are currently at their maximum (rollover) value resets the value to zero.

1.9.2.3 Status and Interrupts

The EXREG reflects the current status of DICE operations, and the Interrupt Status Register (ISREG) records and accumulates the occurrence of events (command completion, error conditions, and external DICE interrupts). The Error Check Register (ECREG) provides specific status for header transfer miscompares and CRC errors.

The EXREG indicates the DICE gate array's current position in the command chain (command list pointer field and command repetition count field) and if DICE operations are active (command execution active field). The status of the current command instance is defined by the command status field and the transfer status field. Command status values are defined in the following table:

Table 1–28, Command Status Codes

Code	Definition
0	Idle
1–7	Command execution in progress
8	Successful command completion
9	Interface not initially idle
10	Interface not returned to idle
11	Bus B octet parity error
12	Interface not initially selected
13	Slave ending status exception
14	Field transfer exception
15	Operation timeout

As listed in the previous table, a value of zero indicates the command execution unit is idle, at least momentarily. Values of 1 through 7 indicate that command execution is in progress, and a value of 8 indicates successful command instance completion. Values greater than 8 are associated with specific exceptions and error conditions.

The transfer status field indicates the current state of the transfer control logic. The status reflects the currently selected field, whether it is active, or complete, and if any exceptions have been encountered. The transfer status values is defined in the following table:

Table 1–29, Transfer Status Codes

Code	Definition
0	Idle
1	Register field end
2	Header field end
3	External field end
4	Initialize transfer logic
5	Register field select
6	Header field select
7	External field select
8	Successful transfer completion
9	Register field CRC error
10	Header field CRC error
11	External field data error (ECC)
12	Transfer length error
13	Register field parity error
14	Header field parity error
15	External field parity error

The EXREG also latches and stores the Bus B control octet returned by the IPI slave in accordance with IPI protocol requirements.

The occurrence of various conditions and events cause the DICE gate array to queue assert interrupts. There are three general types of interrupts:

1. Command completion
2. Exception conditions
3. External events

All interrupts are asserted at a programmable priority level. Each CREG has a completion interrupt level field that causes an interrupt at the indicated level to be set upon successful completion of all instances of the CREG operation. The OPREG contains a exception interrupt level field that globally defines the interrupt level associated with exception conditions and individual priority level fields for each of the three DICE external event signals.

The assertion of interrupts is recorded by the DICE gate array in the ISREG. The ISREG has a bit associated with each of eight interrupt levels. Interrupts have a fixed priority with seven being the highest and zero the lowest. ISREG interrupt bits are set at the first assertion of the corresponding interrupt level. Subsequent assertions of an active interrupt have no affect.

Interrupts are acknowledged by writing to the ISREG with the bit(s) set that correspond to the interrupt level(s) to be acknowledged. Any or all interrupts can be acknowledged simultaneously. ISREG bits not associated with interrupt levels and non-active interrupt bits are ignored by the

Peripheral Interface during interrupt acknowledge. The external event signals are level sensitive and should be cleared before acknowledging the interrupt to avoid continuously reasserting the resulting interrupt.

The Peripheral Interface asserts, as external signals, the encoded value of the highest active interrupt and a single interrupt line that is the logical OR of interrupt levels one through seven (level zero does not cause an external interrupt to the MPU). Interrupts are prioritized, asserted, and acknowledged by a synchronous state machine that resolves ambiguities and race conditions.

1.9.3 Device Interface Command Execution Internal Registers

The DICE internal registers, associated addresses, and access modes are listed in Table 1–30, "DICE Internal Registers". The format and field definitions of the internal registers are listed in the following sections.

1.9.3.1 Data Registers

Descriptions of the functions of the individual DICE data registers are listed in this section.

Header Registers (HREG[3..0])—The HREGs contain the data transferred to the selected slave during header output transfers and the data used for comparison with header input transfers. HREG[3] consists of four cascadable 8-bit upcounters that can be incremented on a command instance basis under program (CREG) control.

Transfer Registers (XREG[7..0])—The XREGs comprise a 32-byte (16 transfer) data buffer for IPI transfers. This register file is used for register (field 1) input and output transfers and header (field 2) input transfers.

Table 1–30, DICE Internal Registers

Address	Name	Description	Access
31	-	Reserved	-
30	-	Reserved	-
29	-	Reserved	-
28	-	Reserved	-
27	HREG3	Header Register 3	R/W
26	HREG2	Header Register 2	R/W
25	HREG1	Header Register 1	R/W
24	HREG0	Header Register 0	R/W
23	XREG7	Transfer Register 7	R/W
22	XREG6	Transfer Register 6	R/W
21	XREG5	Transfer Register 5	R/W
20	XREG4	Transfer Register 4	R/W
19	XREG3	Transfer Register 3	R/W
18	XREG2	Transfer Register 2	R/W
17	XREG1	Transfer Register 1	R/W
16	XREG0	Transfer Register 0	R/W
15	FCREG	Field Configuration Register	R/W
14	HCREG	Header Counter Control Register	R/W
13	ECREG	Error Check Register	R
12	TCREG	Transfer Count Register	R/W
11	EXREG	Execution Register	R
10	ISREG	Interrupt Status Register	R/W
09	OPREG	Operational Parameter Register	R/W
08	-	Reserved	-
07	CREG7	Command Register 7	R/W
06	CREG6	Command Register 6	R/W
05	CREG5	Command Register 5	R/W
04	CREG4	Command Register 4	R/W
03	CREG3	Command Register 3	R/W
02	CREG2	Command Register 2	R/W
01	CREG1	Command Register 1	R/W
00	CREG0	Command Register 0	R/W

1.9.3.2 Status Registers

The DICE status registers are used to record and report operational conditions and completion status to the MPU. Descriptions of the function and the format of the individual DICE status registers are listed in this section.

Error Check Register (ECREG)—The ECREG records the results of byte comparisons between input header data, the contents of HREG[3], and the Cyclic Redundancy Check (CRC) bits. The format of the Error Check Register is defined in Table 1–31, "Error Check Register Format (ECREG)":

Table 1–31, Error Check Register Format (ECREG)

Bit	Definition
31	HREG miscompare byte 00
30	HREG miscompare byte 01
29	HREG miscompare byte 02
28	HREG miscompare byte 03
27	HREG miscompare byte 04
26	HREG miscompare byte 05
25	HREG miscompare byte 06
24	HREG miscompare byte 07
23	HREG miscompare byte 08
22	HREG miscompare byte 09
21	HREG miscompare byte 10
20	HREG miscompare byte 11
19	HREG miscompare byte 12
18	HREG miscompare byte 13
17	HREG miscompare byte 14
16	HREG miscompare byte 15
15	Cyclic redundancy check bit <15>
14	Cyclic redundancy check bit <14>
13	Cyclic redundancy check bit <13>
12	Cyclic redundancy check bit <12>
11	Cyclic redundancy check bit <11>
10	Cyclic redundancy check bit <10>
09	Cyclic redundancy check bit <09>
08	Cyclic redundancy check bit <08>
07	Cyclic redundancy check bit <07>
06	Cyclic redundancy check bit <06>
05	Cyclic redundancy check bit <05>
04	Cyclic redundancy check bit <04>
03	Cyclic redundancy check bit <03>
02	Cyclic redundancy check bit <02>
01	Cyclic redundancy check bit <01>
00	Cyclic redundancy check bit <00>

Execution Status Register (EXREG)—The EXREG contains the status of the currently executing (or last completed) command instance, including condition codes, transfer status, and the IPI level one Bus B octet (if any) returned from the slave. The format of the Execution Register is defined in Table 1–32, "Execution Register Format (EXREG)":

Table 1–32, Execution Register Format (EXREG)

Bit	Definition
31	Command execution active
30	Command list pointer bit <2>
29	Command list pointer bit <1>
28	Command list pointer bit <0>
27	Command status bit <3>
26	Command status bit <2>
25	Command status bit <1>
24	Command status bit <0>
23	Transfer status bit <3>
22	Transfer status bit <2>
21	Transfer status bit <1>
20	Transfer status bit <0>
19	External interrupt 0
18	External interrupt 1
17	External interrupt 2
16	Reserved
15	Command repetition count bit <7>
14	Command repetition count bit <6>
13	Command repetition count bit <5>
12	Command repetition count bit <4>
11	Command repetition count bit <3>
10	Command repetition count bit <2>
09	Command repetition count bit <1>
08	Command repetition count bit <0>
07	IPI Bus B octet bit <7>
06	IPI Bus B octet bit <6>
05	IPI Bus B octet bit <5>
04	IPI Bus B octet bit <4>
03	IPI Bus B octet bit <3>
02	IPI Bus B octet bit <2>
01	IPI Bus B octet bit <1>
00	IPI Bus B octet bit <0>

Interrupt Status Register (ISREG)—The ISREG records the occurrence of DICE interrupts (levels 0 through 7). When written, the assertion of bits <31..24> acknowledge and clear the corresponding interrupt level. Bits <23..0> are read as zero and are ignored on writes. The format of the Interrupt Status Register is defined in Table 1–33, "Interrupt Status Register Format (ISREG)":

Table 1–33, Interrupt Status Register Format (ISREG)

Bit	Definition
31	N/C
30	N/C
29	N/C
28	N/C
27	N/C
26	N/C
25	N/C
24	N/C
23	N/C
22	N/C
21	N/C
20	N/C
19	N/C
18	N/C
17	N/C
16	N/C
15	N/C
14	N/C
13	N/C
12	N/C
11	Command exception interrupt
10	External interrupt 0
09	External interrupt 1
08	External interrupt 2
07	Command completion interrupt level
06	Command completion interrupt level
05	Command completion interrupt level
04	Command completion interrupt level
03	Command completion interrupt level
02	Command completion interrupt level
01	Command completion interrupt level
00	Command completion interrupt level

1.9.3.3 Control Registers

The DICE control registers are used by the MPU to define, initiate, and control DICE operations. Descriptions of the function and the format of the individual DICE control registers are listed in this section.

Field Control Register (FCREG)—The FCREG enables and configures functions associated with IPI information transfers. The FCREG enables exception conditions, error checking (CRC and HREG/XREG comparisons), and contains start pointers for XREG and HREG transfers. The format of the Field Configuration Register is defined in Table 1–34, "Field Configuration Register Format (FCREG)":

Table 1–34, Field Configuration Register Format (FCREG)

Bit	Definition
31	Efr length error enable
30	External exception enable 0 (XERR)
29	External exception enable 1 (BERR)
28	Register field CRC enable
27	Header field CRC enable
26	Counter cascade enable 0-1
25	Counter cascade enable 1-2
24	Counter cascade enable 2-3
23	Register transfer start pointer bit <3>
22	Register transfer start pointer bit <2>
21	Register transfer start pointer bit <1>
20	Register transfer start pointer bit <0>
19	Header transfer start pointer bit <3>
18	Header transfer start pointer bit <2>
17	Header transfer start pointer bit <1>
16	Header transfer start pointer bit <0>
15	HREG compare enable byte 00
14	HREG compare enable byte 01
13	HREG compare enable byte 02
12	HREG compare enable byte 03
11	HREG compare enable byte 04
10	HREG compare enable byte 05
09	HREG compare enable byte 06
08	HREG compare enable byte 07
07	HREG compare enable byte 08
06	HREG compare enable byte 09
05	HREG compare enable byte 10
04	HREG compare enable byte 11
03	HREG compare enable byte 12
02	HREG compare enable byte 13
01	HREG compare enable byte 14
00	HREG compare enable byte 15

Header Counter Control Register (HCREG)—The HCREG specifies the rollover values for the counters associated with HREG[3]. The HREG[3] counters reset to zero when incremented past these values. The format of the Header Counter Control Register is defined in Table 1–35, "Header Counter Control Register Format (HCREG)":

Table 1–35, Header Counter Control Register Format (HCREG)

Bit	Definition
31	Counter 0 rollover value bit <7>
30	Counter 0 rollover value bit <6>
29	Counter 0 rollover value bit <5>
28	Counter 0 rollover value bit <4>
27	Counter 0 rollover value bit <3>
26	Counter 0 rollover value bit <2>
25	Counter 0 rollover value bit <1>
24	Counter 0 rollover value bit <0>
23	Counter 1 rollover value bit <7>
22	Counter 1 rollover value bit <6>
21	Counter 1 rollover value bit <5>
20	Counter 1 rollover value bit <4>
19	Counter 1 rollover value bit <3>
18	Counter 1 rollover value bit <2>
17	Counter 1 rollover value bit <1>
16	Counter 1 rollover value bit <0>
15	Counter 2 rollover value bit <7>
14	Counter 2 rollover value bit <6>
13	Counter 2 rollover value bit <5>
12	Counter 2 rollover value bit <4>
11	Counter 2 rollover value bit <3>
10	Counter 2 rollover value bit <2>
09	Counter 2 rollover value bit <1>
08	Counter 2 rollover value bit <0>
07	Counter 3 rollover value bit <7>
06	Counter 3 rollover value bit <6>
05	Counter 3 rollover value bit <5>
04	Counter 3 rollover value bit <4>
03	Counter 3 rollover value bit <3>
02	Counter 3 rollover value bit <2>
01	Counter 3 rollover value bit <1>
00	Counter 3 rollover value bit <0>

Transfer Count Register (TCREG)—The TCREG contains the transfer length count for HREG, XREG, and external field IPI transfers. All count values are in halfwords (16 bits). The format of the Transfer Count Register is defined in Table 1–36, "Transfer Count Register Format (TCREG)":

Table 1–36, Transfer Count Register Format (TCREG)

Bit	Definition
31	Register field count bit <3>
30	Register field count bit <2>
29	Register field count bit <1>
28	Register field count bit <0>
26	Header field count bit <3>
26	Header field count bit <2>
25	Header field count bit <1>
24	Header field count bit <0>
23	External field count bit <23>
22	External field count bit <22>
21	External field count bit <21>
20	External field count bit <20>
19	External field count bit <19>
18	External field count bit <18>
17	External field count bit <17>
16	External field count bit <16>
15	External field count bit <15>
14	External field count bit <14>
13	External field count bit <13>
12	External field count bit <12>
11	External field count bit <11>
10	External field count bit <10>
09	External field count bit <09>
08	External field count bit <08>
07	External field count bit <07>
06	External field count bit <06>
05	External field count bit <05>
04	External field count bit <04>
03	External field count bit <03>
02	External field count bit <02>
01	External field count bit <01>
00	External field count bit <00>

Operational Parameter Register (OPREG)—The OPREG defines global DICE parameters. This register enables command execution and specifies command list length. This register also associates interrupt levels with exception conditions and external events, and contains the time out value used in IPI control sequences. The format of the Operational Parameter Register is defined in Table 1–37, "Operational Parameter Register Format (OPREG)":

Table 1–37, Operational Parameter Register Format (OPREG)

Bit	Definition
31	Command list valid bit
30	Command list end pointer bit <3>
29	Command list end pointer bit <2>
28	Command list end pointer bit <1>
27	Bad ending status enable bit <5>
26	Bad ending status enable bit <4>
25	Bad ending status enable bits <3..0>
24	Exception interrupt level bit <2>
23	Exception interrupt level bit <1>
22	Exception interrupt level bit <0>
21	External interrupt 0 level bit <2> (IPI attention)
20	External interrupt 0 level bit <1> (IPI attention)
19	External interrupt 0 level bit <0> (IPI attention)
18	External interrupt 1 level bit <2> (PIGA complete)
17	External interrupt 1 level bit <1> (PIGA complete)
16	External interrupt 1 level bit <0> (PIGA complete)
15	External interrupt 1 level bit <2> (PIGA error)
14	External interrupt 2 level bit <1> (PIGA error)
13	External interrupt 2 level bit <0> (PIGA error)
12	Extended timer enable
11	Programmable timer value bit <11>
10	Programmable timer value bit <10>
09	Programmable timer value bit <09>
08	Programmable timer value bit <08>
07	Programmable timer value bit <07>
06	Programmable timer value bit <06>
05	Programmable timer value bit <05>
04	Programmable timer value bit <04>
03	Programmable timer value bit <03>
02	Programmable timer value bit <02>
01	Programmable timer value bit <01>
00	Programmable timer value bit <00>

Command Register (CREG[0..7])—The DICE gate array contains eight CREGs. The CREGs contain parameters associated with individual IPI functions. Each register specifies the opcode, modifiers, repetition count, completion interrupt level, and supplies the IPI level one Bus A octet. The CREGs can be randomly accessed by the MPU and are executed by DICE logic in order beginning at CREG[0]. The format of a Command Register is defined in Table 1–38, "Command Register Format (CREG)":

Table 1–38, Command Register Format (CREG)

Bit	Definition
31	Opcode bit <2>
30	Opcode bit <1>
29	Opcode bit <0>
28	Opcode modifier bit <2> (Inhibit buffer block count)
27	Opcode modifier bit <1> (Reserved)
26	Opcode modifier bit <0> (Reserved)
25	Completion interrupt level bit <2>
24	Completion interrupt level bit <1>
23	Completion interrupt level bit <0>
22	Register transfer enable
21	Header transfer enable
20	External transfer enable
19	Counter increment enable byte 0
18	Counter increment enable byte 1
17	Counter increment enable byte 2
16	Counter increment enable byte 3
15	Repetition count bit <7>
14	Repetition count bit <6>
13	Repetition count bit <5>
12	Repetition count bit <4>
11	Repetition count bit <3>
10	Repetition count bit <2>
09	Repetition count bit <1>
08	Repetition count bit <0>
07	IPI Bus A octet bit <7>
06	IPI Bus A octet bit <6>
05	IPI Bus A octet bit <5>
04	IPI Bus A octet bit <4>
03	IPI Bus A octet bit <3>
02	IPI Bus A octet bit <2>
01	IPI Bus A octet bit <1>
00	IPI Bus A octet bit <0>

1.10 Data Path and Error Detection Gate Array (DPED)

The Data Path and Error Detection (DPED) gate array was designed to perform the following two functions:

1. Input/Output Transfers—Transfers of data between the data buffer and the IPI port (between the Buffer Bus and Bus A & Bus B)
2. Local Bus Accesses—Transfers between internal DPED registers and the Local Bus

1.10.1 Operations

Double octets received from an Intelligent Peripheral Interface (IPI) port are packed into longwords and output to a data buffer that resides on the IDC. Data transferred in the opposite direction is unpacked and output to the IPI port as double octets.

On outputs to an IPI port, Error Correction Codes (ECC) can be generated and output along with the data. During inputs from an IPI port, data can be checked for correct ECC.

1.10.1.1 Input/Output Transfers

An input transfer consists of the transfer of data from Bus A and Bus B to the Buffer Bus (BB). The DPED gate array clocks data from Bus A and Bus B into serial register SERD, and the data is moved along the pipe of serial registers. When the serial registers SERD, SERC, and SERB contain the most significant six bytes of a longword, the six bytes along with the least significant two bytes coming off of Bus A and Bus B are loaded into the parallel registers PARA, PARB, PARC, PARD. The parallel registers directly feed the Buffer Bus. The data from the Buffer Bus is written into the data buffer, and at the same time, the most significant two bytes of a new word are loaded into the serial register SERD and the process is begun again.

After all non-ECC data has been read into the serial register SERD, the DPED gate array is ready to accept ECC data. The ECC data is clocked into the ECC logic, and this data is checked against the ECC generated by the DPED gate array during the input of non-ECC data. After the last piece of data is registered in the DPED gate array, the operation is complete if there was ECC data. If there was no ECC data, then the data remaining in the serial or parallel registers is transferred to the data buffer.

An output operation consist of the transfer of data from the Buffer Bus to Bus A and Bus B. The DPED gate array prefetches a longword of data from the data buffer and reads the data into serial registers SERA, SERB, SERC, and SERD with the most significant two bytes being read in SERA. After the DPED gate array prefetches the first longword of data, the data in the serial registers is shifted out onto Bus A and Bus B. After each halfword of a longword had been registered in SERA, a new longword is loaded into the serial registers. These operations continues until all data has been transferred. After all data has been transferred, ECC data is output if ECC generation was specified for this operation.

1.10.1.2 Input/Output Parameters

Four parameters need to be set for an input or output operation. The transfer size, the block size, and the ECC interleave are set in the Basic Control Register (BCREG). The BCREG can be written during local bus writes. The last parameter is specified with the Block Reset Inhibit (IBRT) signal. This parameter prevents the DPED gate array from incrementing to the next block boundary at the start of the next operation and from sending out a Block Transferred (OBKX) signal at the end of the current operation unless the operation has ended on a block boundary.

1.10.1.3 Error Conditions

The following four types of errors can occur for IPI transfers:

1. ECC errors
2. Length errors
3. Overrun/underrun errors
4. Parity errors

Error Correction Code (ECC) errors occur only during reads from the IPI port when the ECC generated by the DPED gate array for the incoming data is different from the ECC coming in from the IPI port. An ECC error is valid only between operations and then only for the last operation.

Length errors are the result of receiving more or less synchronized input then is expected during the time that the DPED gate array is ready to receive synchronized input. The number that is expected is equivalent to the transfer size. This error is valid only between operations and then only for the operation that was just completed.

Overrun/underrun errors are the result of not receiving a buffer grant at the necessary times during an operation. For reads from an IPI port, if a buffer grant is not available when the last longword of a block goes out on the Buffer Bus, an overrun or underrun error is generated. A signal indicating that the DPED gate array is not accessing data on the Buffer Bus is also generated. On writes to an IPI port, if a buffer grant is not available when the second-to-the-last longword in a block is read from the Buffer Bus then an overrun or underrun error is generated.

Parity is checked on data going to the Buffer Bus and coming from the Buffer Bus. If parity is bad, a data buffer bus parity error is generated and not cleared until the start of the next operation.

1.10.1.4 Local Bus Accesses

A local bus access is a local read of data from the Local Bus (BL) or a write of data to the BL. A local access is started provided an IPI read or write operation is not in progress. A register address specifies the address of the internal register that is to be accessed. The register address is used directly to gate the contents of the indicated register directly to the BL on local reads.

When data has been output onto the BL or written into an internal register, a register access acknowledgment is generated. If an invalid access is attempted, then a local access fault is generated. An invalid access involves attempts to write read-only registers or to access addresses above 1000. A local access fault is also generated if there is a parity error on the incoming data from BL.

1.10.2 Data Path and Error Detection Internal Registers

The DPED registers, associated addresses, access modes, and sizes are listed in Table 1–39, "DPED Internal Registers".

Table 1–39, DPED Internal Registers

Micro Address	Name	Description	IAD	Access	Size
00FFx0A0	BADDR	Buffer Address Register	1000	R/W	16
00FFx09C	SYNDROME	ECC Syndrome Register	0111	R	32
00FFx098	XCNT	Transfer Count Register	0110	R	24
00FFx094	BKCNT	Block Count Register	0101	R	13
00FFx090	BCREG	Basic Control Register	0100	R/W	32
00FFx08C	PARC & PARD	Parallel C & D Data Registers	0011	R	36
00FFx088	PARA & PARB	Parallel A & B Data Registers	0010	R	36
00FFx084	SERC & SERD	Serial C & D Data Registers	0001	R	36
00FFx080	SERA & SERB	Serial A & B Data Registers	0000	R	36

Descriptions of the DPED internal registers are listed below:

- **Buffer Address (BADDR)**—Contains the longword address for the data buffer (RAMs) from which data will be read or to which data will be written. Parity is generated for this register on local reads.
- **ECC Syndrome (SYNDROME)**—When first read, this register contains bytes 0, 1, 4, and 5 of an interleave. After all interleaves have been read once, this register contains bytes 2, 3, 6, and 7 of an interleave.
- **Transfer Count (XCNT)**—Contains the remaining longwords that need to be transferred to complete an operation.
- **Block Count (BKCNT)**—Contains the remaining number of longwords that need to be transferred to complete the transfer of one block. Parity is generated for this register on local reads.
- **Basic Control Register (BCREG)**—Bit <31> contains a copy of OVUN from the previous operation. Bit <30> contains a copy of OLGR from the previous operation. Neither of these bits is writable. Bits <29..26> contain the block size represented as a power of two with a minimum block size of 32 bytes (block size = 2^n bytes where n >= 5). Bits <25..24> contain the interleave factor for ECC: 00 being no interleave, 01 being an interleave of one, 10 being an interleave of two, and 11 being an interleave of four. The interleave must be less than or equal to the transfer size in halfwords (interleave <= transfer size in halfwords). Bits <23..0> contain the transfer size in halfwords. Parity is generated for this register on local reads.

- **Parallel Data A (PARA)**—Contains bytes 0 and 1 of the data being transferred to the data buffer. Parity is taken from this register and is not generated. The only time this register will contain non-zero data when read is after the termination of a read operation from an IPI port.
- **Parallel Data B (PARB)**—Contains bytes 2 and 3 of the data being transferred to the data buffer. Parity is taken from this register and is not generated. The only time this register will contain non-zero data when read is after the termination of a read operation from an IPI port.
- **Parallel Data C (PARC)**—Contains bytes 4 and 5 of the data being transferred to the data buffer. Parity is taken from this register and is not generated. The only time this register will contain non-zero data when read is after the termination of a read operation from an IPI port.
- **Parallel Data D (PARD)**—Contains bytes 6 and 7 of the data being transferred to the data buffer. Parity is taken from this register and is not generated. The only time this register will contain non-zero data when read is after the termination of a read operation from an IPI port.
- **Serial Data A (SERA)**—Contains data that is output onto the IPI port during writes to a disk (output operations). For reads from a disk, this register is not used. Parity is taken from this register and is not generated. The only time this register will possibly contain non-zero data when read is after the premature termination of a write operation to an IPI port.
- **Serial Data B (SERB)**—For writes to a disk, this register contains data that goes to SERA. For reads from a disk, this register contains data that is clocked into PARA every time a longword is read from the IPI port. Parity is taken from this register and is not generated. The only time this register will possibly contain non-zero data when read is after the premature termination of a read or write operation to an IPI port.
- **Serial Data C (SERC)**—For writes to a disk, this register contains data that goes to SERB. For reads from a disk, this register contains data that is clocked into PARB every time a longword is read from the IPI port. Parity is taken from this register and is not generated. The only time this register will possibly contain non-zero data when read is after the premature termination of a read or write operation to an IPI port.
- **Serial Data D (SERD)**—For writes to a disk, this register contains data that goes to SERC. For reads from a disk, this register contains data that is clocked into PARC every time a longword is read from the IPI port. Parity is taken from this register and is not generated. The only time this register will possibly contain non-zero data when read is after the premature termination of a read or write operation to an IPI port.

1.11 IDC LEDs

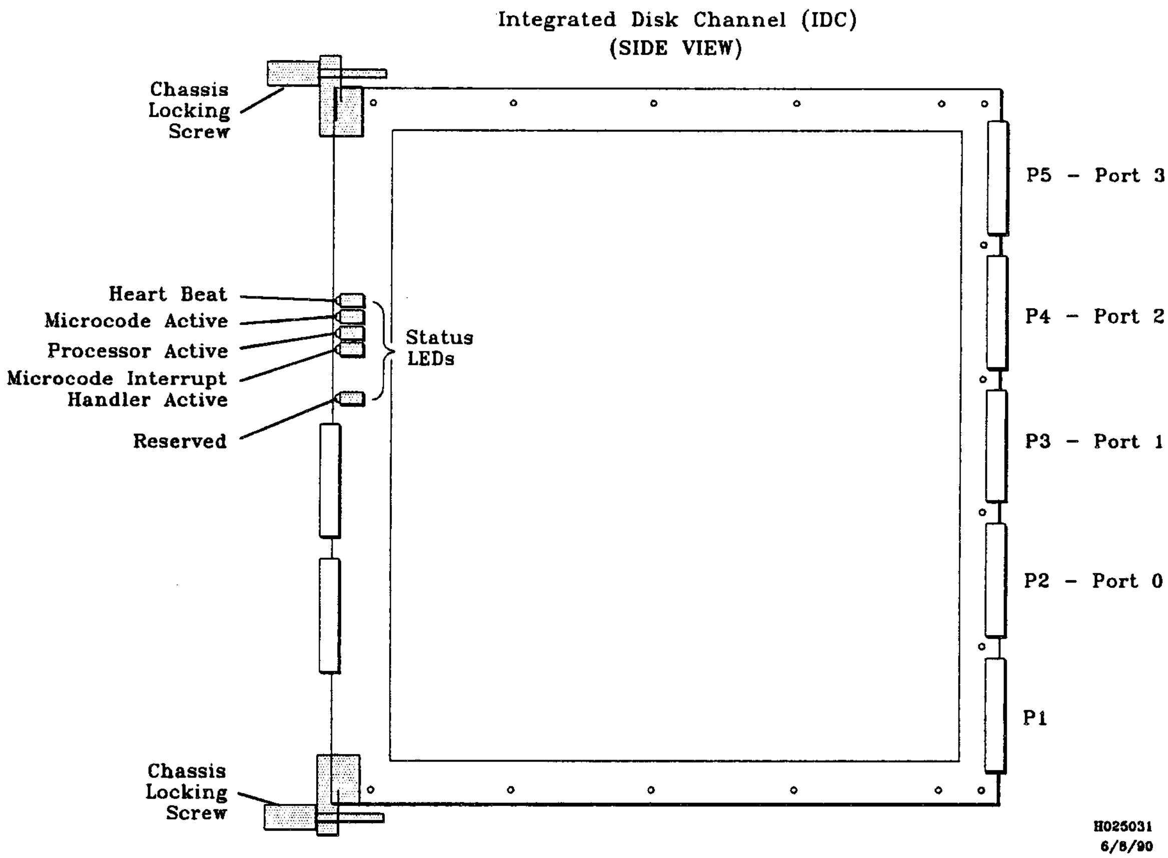
The IDC LEDs are defined as follows:

- **Heartbeat**—The Heartbeat LED blinks to indicate a 1-Hz clock is active.
- **Microcode Active**—The Microcode Active LED is on while the microcode is executing. The Microcode Active LED is turned off when the microcode crashes. However, the Microcode Active LED may remain on if a crash occurs and the crash is not detected.
- **Processor Active**—The Processor Active LED is on while the processor is executing noninterrupt code. The Processor Active LED is off when the processor is idle.

- **Microcode Interrupt Handler Active**—The Microcode Interrupt Handler Active LED is on while the microcode is in an interrupt handler. The Microcode Interrupt Handler Active LED is off when an interrupt is not present.
- **Reserved**—The Reserved LED is reserved for future use.

The IDC LEDs are shown in the following figure:

Figure 1–5, IDC LEDs



1.12 Drive Front Panel Indicator and Switch Descriptions

The front panel indicators and switches are described in the following table:

Table 1–40, Front Panel Indicators and Switches

Item	Type	Description
Address	Indicators	Indicates the binary logical address (0–7) of drive
	Switch	Used to set the logical address of a drive[1] Press for 2 to 3 seconds to advance the logical address
Selected	Indicator	**ON** = Drive selected by controller
		OFF = Drive not selected by controller
Start	Indicator	**ON** = Drive is ready
		OFF = Drive is not ready
		FLASHING = Drive is in START or STOP cycle
	Switch	Used to start or stop a drive Press and release to begin START cycle Press and release again to begin STOP cycle
Fault	Indicator	**ON** = Drive fault condition exists
		OFF[2] = Drive fault condition does not exist
	Switch	Used to turn off FAULT indicator Press and release to turn off FAULT indicator when a fault condition no longer exists
Write Protect	Indicator	**ON** = Drive write operations disabled
		OFF = Drive write operations enabled
	Switch[3]	Used to enable or disable drive write operations Press and release to disable drive write operations Press and release again to enable drive write operations

[1] The logical address is stored in memory when the power supply On/Standby switch on the front of the power supply is switched to the Standby (0) position, the AC power switch on the rear of the disk drive is switched to the OFF position, or when there is a loss of site AC power.

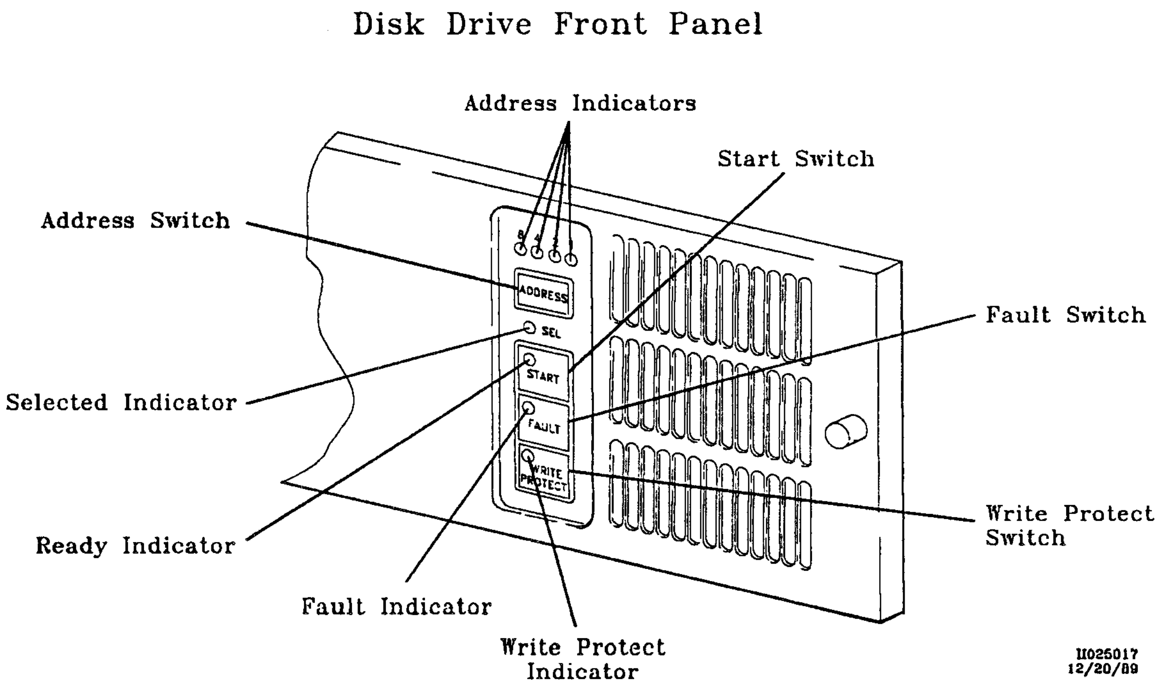
[2] The indicator is turned OFF by any of the following, if the fault condition no longer exists:

1. Pressing the FAULT switch
2. A drive start cycle
3. A Fault Clear command from the controller

[3] The WRITE PROTECT switch cannot enable drive write operations if the WP/N switch on the drive control board is set to WP (Write Protect).

The drive front panel indicators and switches are shown in the following figure:

Figure 1–6, Drive Front Panel



1.13 Drive Control Board Jumper and Switch Descriptions

The drive control board jumpers and switches are described in the following table:

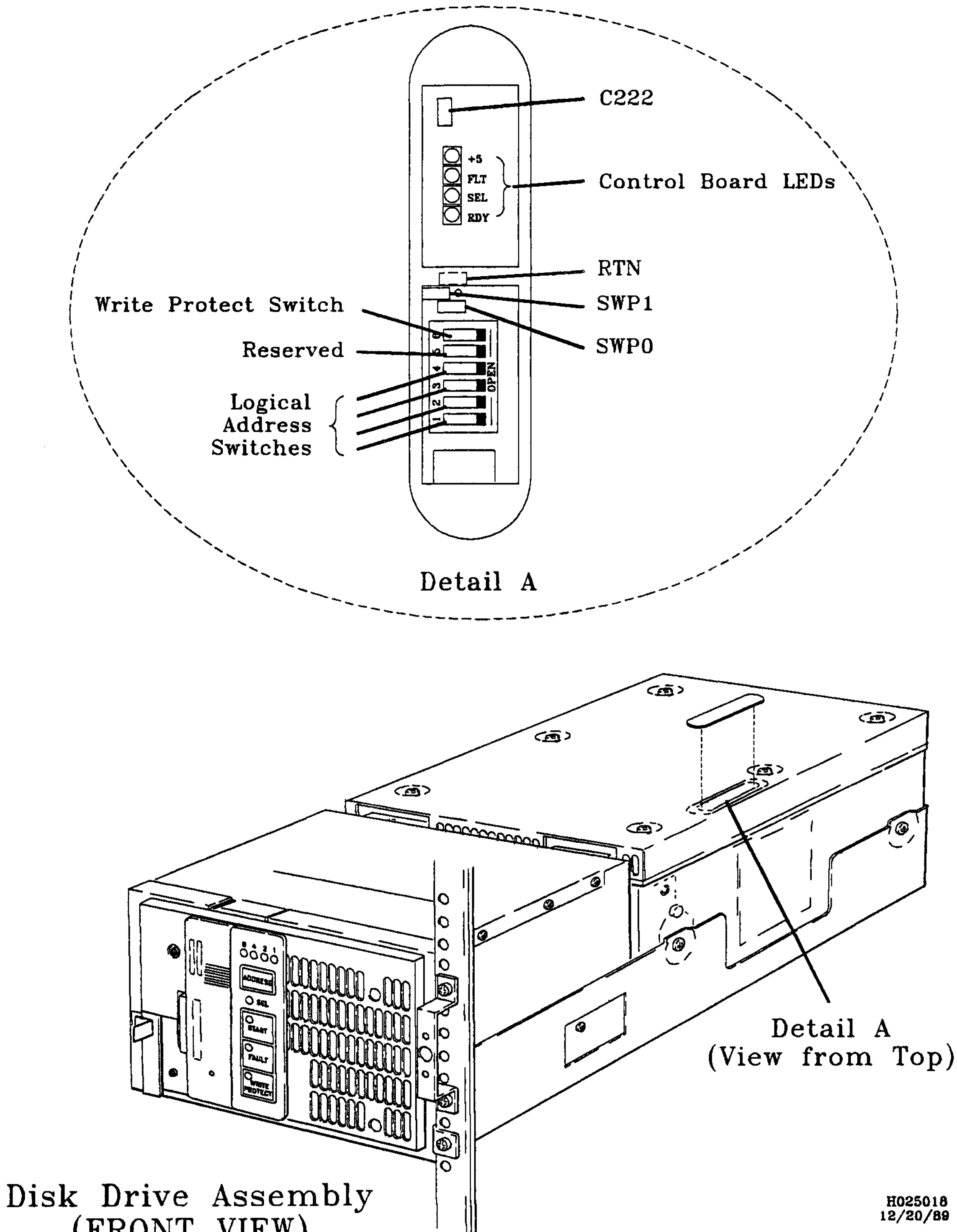
Table 1–41, Drive Control Board Jumpers and Switches

Item	Type	Setting	Description
RTN	Jumper	ON	Enables option to return heads to original position following sweep segment if drive was selected during last 12 minutes, otherwise heads stay on last track of sweep segment
		OFF[1]	Disables option to return heads to original position following sweep segment
SWP1	Jumper	ON	Disables option for sweep cycle on seeks
		OFF[1]	Enables option for sweep cycle on seeks
SWPD	Jumper	ON[1]	Disables sweep cycle operation
		OFF	Enables sweep cycle operation
Write Protect	Switch	ON-CLOSED[1]	Write operations enabled
		OFF-OPEN	Write operations disabled
Reserved	Switch	OFF-OPEN[1]	Reserved for future use
Logical Address	Switches	ON-CLOSED[1]	Ignored if a front panel is used

[1] CONVEX setting

The drive control board jumpers and switches are shown in the following figure:

Figure 1–7, Drive Control Board



1.14 Drive Input/Output Board Switch Descriptions

The drive input/output board switches are described in the following table:

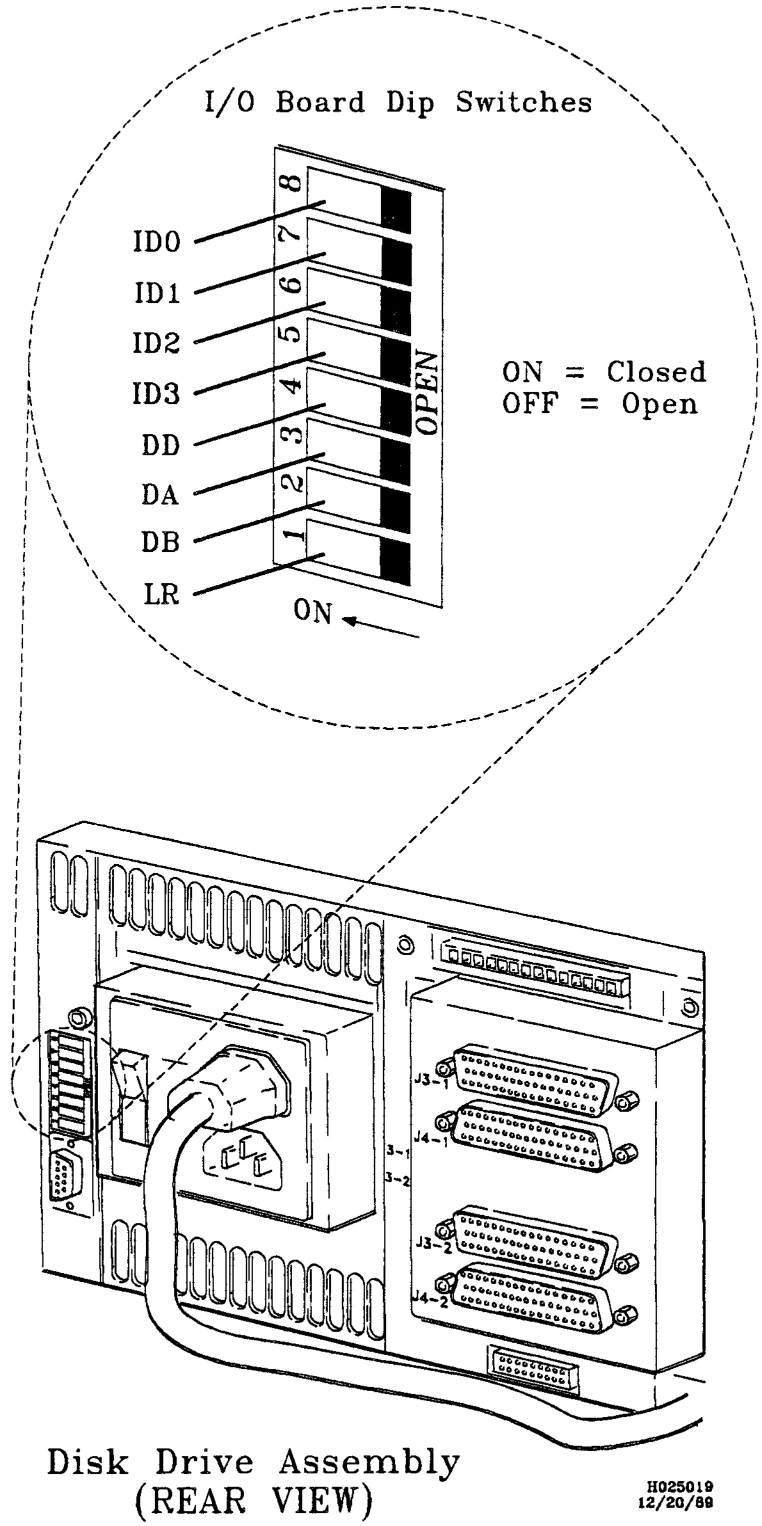
Table 1–42, I/O Board Dip Switches

Switch	Setting	Description
ID0	ON[1]	**Microcode ID**—Switches ID0–ID3 are used to assign a unique device configuration code
ID1	OFF[1]	
ID2	ON[1]	
ID3	OFF[1]	
DD	ON	**Disable Diagnostics**—Disables internal diagnostics from performing read and write operations
	OFF[1]	**Enable Diagnostics**—Enables internal diagnostics to perform read and write operations
DA	ON	**Disable Port A**—Disables port A for normal operation
	OFF[1]	**Enable Port A**—Enables port A for normal operation
DB	ON[1]	**Disable Port B**—Disables port B for normal operation
	OFF	**Enable Port B**—Enables port B for normal operation
LR	ON	**Remote Operation**—Spin-up command is required to start drive
	OFF[1]	**Local Operation**—Disk spin-up starts when DC power is applied to the drive or when the drive front panel START switch is pressed and released

[1] CONVEX switch setting

The drive input/output board switches are shown in the following figure:

Figure 1–8, Drive Input/Output Board



1.15 Power Controller Description

The 48-amp power controller supplies 200 VAC–240 VAC 50 Hz–60 Hz single-phase power. The power controller delivers filtered, fused power to single phase peripherals. The power controller also delivers power to cabinet blowers and cabinet sensors and provides local or remote control when powering up the peripheral equipment in the cabinet.

The 48-amp power controller (CONVEX part number 500-001025-200) is used in domestic and international peripheral cabinets. The domestic power cord (CONVEX part number 605-030005-200) and the international power cord (CONVEX part number 605-030006-200) for the power controller are different. The domestic power cord includes a HUBBELL connector, but the international power cord does not include a connector.

A 24-amp power strip (CONVEX part number 500-000296-200) can be installed on each side of a peripheral cabinet to distribute power to peripheral devices and cabinet blowers. Each power strip contains ten, 6-amp IEC appliance outlets. A power strip connects to a 24-amp twist-lock connector on the rear panel of the power controller.

The power controller front panel switches and indicators are defined as follows:

- **Main Circuit Breaker**—A 2-pole circuit breaker that disconnects power to the unit in the event of an internal short to the power controller. The main circuit breaker also functions as the main ON/OFF switch. The main circuit breaker is listed in the following table:

Table 1–43, Main Circuit Breaker

Contactor	Breaker	Rating	Trip (10 sec)	Connectors
N/A	CB1	60A	N/A	J15 Main Input

- **Output Circuit Breakers**—These 2-pole circuit breakers are designed to disconnect power to the power outlets. The output circuit breakers are listed in the following table:

Table 1–44, Output Circuit Breakers

Contactor	Breaker	Rating	Trip (10 sec)	Connectors
K2	CB2	30A	75	J14 Twist-lock
K1	CB3	30A	75	J9 Twist-lock
K2	CB4	20A	50	J12 & J13 Duplex
K1	CB5	20A	50	J10 & J11 Duplex

- **Transformer Circuit Breaker**—A 2-pole push-to-reset breaker that disconnects power to the transformer in the event the transformer's secondary outputs are shorted. The transformer circuit breaker is listed in the following table:

Table 1–45, Transformer Circuit Breaker

Contactor	Breaker	Rating	Trip (10 sec)	Connectors
N/A	CB6	2A	5	Transformer

- **Local/Remote Switch**—A 3-position switch that provides control of two contactors. The three positions are defined as follows:
  - **OFF**—24 VDC is disconnected from the contactors
  - **REMOTE**—A remote switch closure is required to short J7-1 to J7-3 REMOTE IN. Once the connection is made, the contactors sequence as described in Table 1-46, "Contactor Closing Sequence."
  - **LOCAL**—Contactor K1 is energized with 24 VDC instantly and contactor K2 is energized after delay T1 as described in Table 1-46, "Contactor Closing Sequence." If the switch is moved to OFF before T1 times out, T1 is reset in preparation for the next turn-on cycle.

Table 1–46, Contactor Closing Sequence

Time	Event
T0	24 VDC supplied, 2-pole contactor K1[2] closes
T0 + T1[1] sec	2-pole contactor K2[2] closes
T1 + T2[1] sec	J8-1 is shorted to J8-3 REMOTE OUT to next cabinet

[1] T1 and T2 are set to an 8-second maximum interval. Each time interval can be adjusted from 1 to 8 seconds by accessing internal switches.
[2] K1 is used to switch power to twist-lock connector J9 and duplexes J10 and J11. K2 is used to switch power to twist-lock connector J14 and duplexes J12 and J13.

- **Main Power Indicator**—This indicator is connected to the load side of the main circuit breaker. The indicator is ON while input power is being delivered to the load side of the main circuit breaker.
- **Contactor Indicators**—An indicator is connected to the DC supply of each contactor to indicate the contactor is closed. An indicator is ON while the output circuit breakers are in the ON position and input power is being delivered to the associated output connectors. The contactor indicators are listed in the following table:

Table 1–47, Contactor Indicators

Indicator	Description
Contactor 2	Right power strip
Contactor 1	Left power strip

- **Shutdown Input Indicators**—An indicator is provided for each of the six sensor inputs. If a sensor shuts a contactor OFF, power remains to the indicator to provide a visible display of the problem source. The main circuit breaker must be reset to clear an indicator and to allow the power controller to power back up. The shutdown input indicators are listed in the following table:

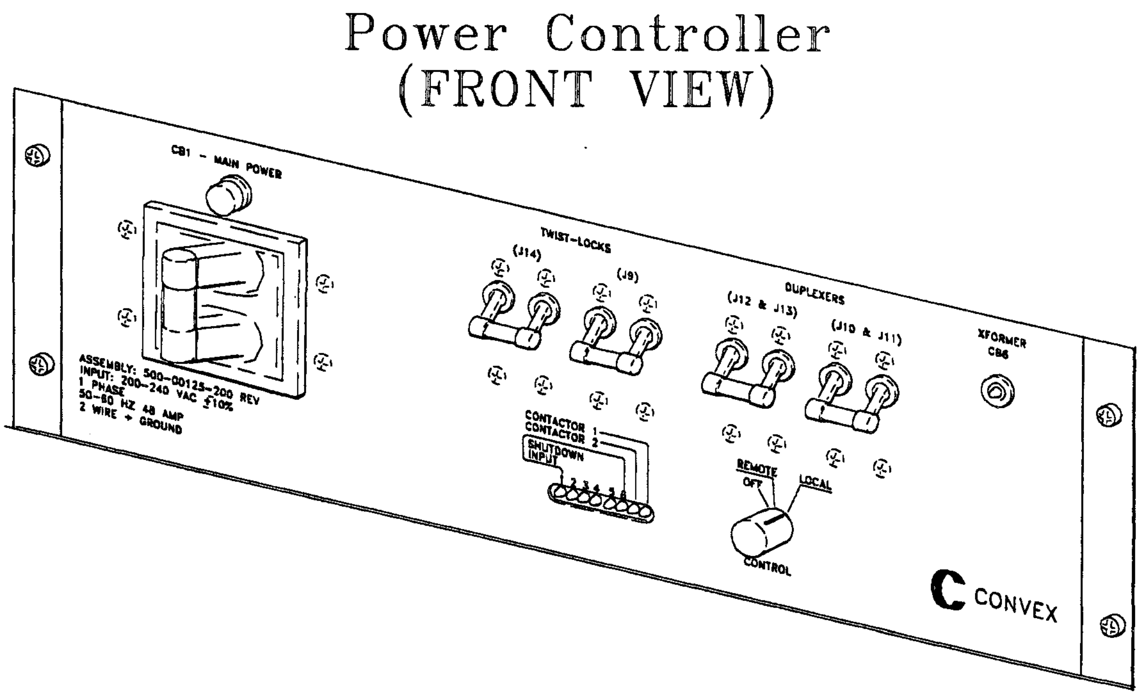
Table 1–48, Shutdown Input Indicators

Indicator	Description
1	Peripheral cabinet right fan failed
2	Peripheral cabinet left fan failed
3	Peripheral cabinet ambient air temperature exceeded 120 ° F
4–6	Unused sensor inputs[2]

[1] All unused input connectors must have pin 2 and pin 4 jumpered to override the open cable detection.

The power controller front panel switches and indicators are shown in the following figure:

Figure 1–9, Power Controller Front Panel Switches and Indicators



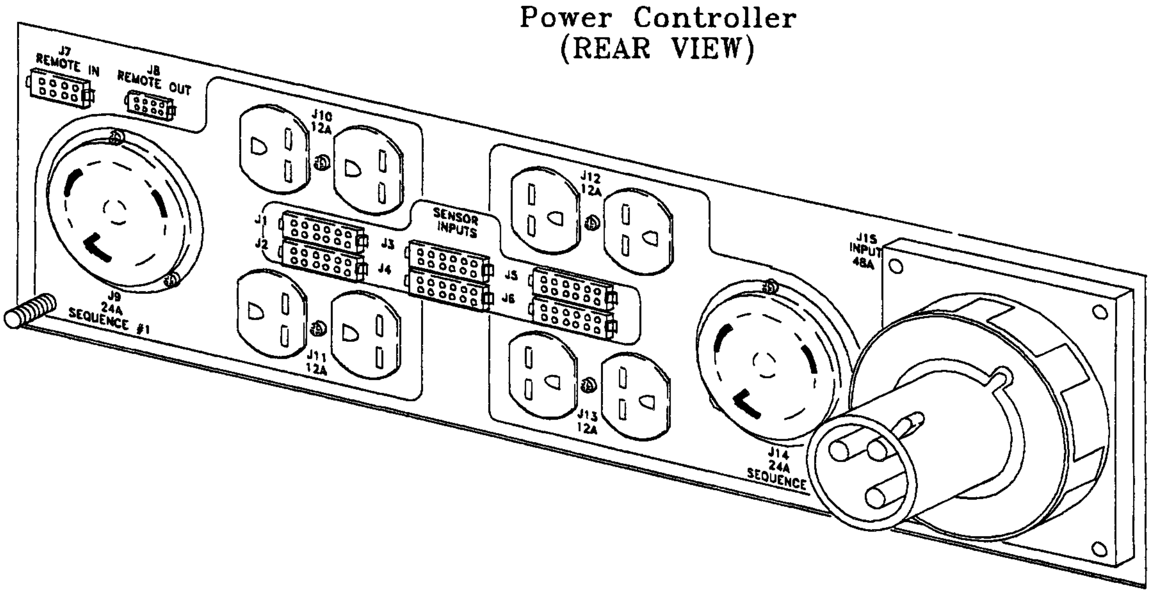
H025034
11/22/89

The power controller rear panel connections are defined as follows:

- J1—Fan operation sensor input (4 wire)
- J2—Fan operation sensor input (1 wire)
- J3—Thermostat sensor input
- J4—Unused sensor input
- J5—Unused sensor input
- J6—Unused sensor input
- J7—Power sequencing control cable connection; connection for cable from CPU cabinet or previous peripheral cabinet
- J8—Power sequencing control cable connection; connection for cable to next peripheral cabinet
- J9— Power strip connection
- J10—Unused duplex outlet
- J11—Unused duplex outlet
- J12—Unused duplex outlet
- J13—Unused duplex outlet
- J14—Power strip connection
- J15—Main AC power cord connection

The power controller rear panel connections are shown in the following figure:

Figure 1–10, Power Controller Rear Panel Connections



H025035
1/25/90

1.16 IDC Specifications

The following table lists the specifications for the IDC:

Table 1–49, IDC Specifications

Parameter	Value/Comment
Width	19.0 in (48.26 cm)
Height	20.3 in (51.56 cm)
Thickness	0.56 in *(approx)* (1.65 cm)
Weight	8 lb *(approx)* (3.62 kg)
Power Dissipation, Maximum	180 W
Temperature Range,[1] **Maximum**	60 °F to 90 °F (15 °C to 32 °C)
Temperature Range,[1] **Recommended**	70 °F to 80 °F (21 °C to 26.6 °C)
Temperature Change, Maximum Rate	14.4 °F/hr (8 °C/hr)
Humidity Range, Maximum	10% to 90% with no condensation
Humidity Range, Recommended	40% to 60% with no condensation

[1] At altitudes above 3,280 ft (1,000 m), lower air densities affect air conditioning. If the unit is located above this altitude, lower the recommended temperature value by 1 °F/1,000 ft (2 °C/1,000 m).

1.17 Cabinet/Drive Specifications

Specifications for the peripheral cabinet and disk drive are listed in the following sections.

1.17.1 Dimensions and Weights

The physical dimensions of the peripheral equipment are listed in the following table (metric equivalents are shown in parentheses).

Table 1–50, Peripheral Equipment Dimensions and Weights

Equipment	Weight pounds (kg)	Width inches (cm)	Depth/Length inches (cm)	Height inches (cm)
EXP-105[1] **Peripheral Cabinet**	382.0 (173.3)	25.1 (63.7)	39.3 (99.8)	62.3 (158.1)
DKD-501 Disk Drive	60.0[2]/46.0[3] (27.2/20.9)	19.0[4] (48.1)	28.0 (71.1)	5.25[5] (13.3)
DKD-502 Disk Drive	60.0[2]/46.0[3] (27.2/20.9)	19.0[4] (48.1)	28.0 (71.1)	5.25[5] (13.3)

[1] Includes power controller and one side panel
[2] Includes drive tray
[3] Drive only
[4] Includes one or two drives in tray
[5] Includes one or two drives in tray

1.17.2 Electrical Specifications

The following sections provides the North American and International electrical specifications.

1.17.2.1 Equipment North American Specifications

The following table lists electrical specifications for the peripheral cabinet and peripheral equipment:

Table 1–51, Peripheral Equipment North American Specifications

Equipment	Voltage ±10%	Frequency ±1 Hz	Phase No.	Current (amps)
EXP-105 Peripheral Cabinet[1]	200 – 240	60	1	48.0[2]
DKD-501 Disk Drive	85 – 132	48 – 62	1	1.51 – 1.22[3]
DKD-502 Disk Drive	85 – 132	48 – 62	1	1.51 – 1.22[3]

[1] This device contains a standalone power distribution system.
[2] This is the maximum current rating for a peripheral cabinet's power control circuits.
[3] This is the nominal line current with disk rotating and carriage moving

1.17.2.2 Equipment International Specifications

The peripheral cabinet and peripheral equipment electrical specification are listed in the following table for international installations:

Table 1–52, Peripheral Equipment International Specifications

Equipment	Voltage ± 10%	Frequency ± 1 Hz	Phase No.	Current (amps)
EXP-105 Peripheral Cabinet[1]	208 – 240	50 – 60	1	48.0[2]
DKD-501 Disk Drive	177 – 264	48 – 62	1	1.51 – 1.22[3]
DKD-502 Disk Drive	177 – 264	48 – 62	1	1.51 – 1.22[3]

[1] This device contains a standalone power distribution system.
[2] This is the maximum current rating for a peripheral cabinet's power control circuits.
[3] This is the nominal line current with disk rotating and carriage moving

1.17.3 Power Cord and Cable Specifications

The following tables lists the specifications for power cords and power cables:

Table 1–53, Input Cord Specifications

Type	Jack	Connector	Voltage	Agency	Min Wire AWG	Length
Domestic[1]	J15	360C6W	250 VAC	UL/CSA	12 AWG	14 ft
International[2]	J15	360C6W	250 VAC	VDE	$10mm^2$	4.3 m

[1] The customer needs a HUBBELL Part Number 360C6W (CONVEX Part Number 304-000036-001) cable connector to cable the power controller to the site AC power source.
[2] The international cord must be hard wired into the site AC power source.

Table 1–54, Output Cable Specifications

Type	Jack	Connector	Voltage	Agency	Min Wire AWG	Length
Twist Lock	J14 & J9	NEMA L11-30P	250 VAC	UL/CSA	12 AWG	7.6 ft
Duplex	J10–J13	NEMA 6-15P	250 VAC	UL/CSA	None	7.6 m

Table 1–55, Control Cable Specifications

Type	Jack	Voltage	Min Wire AWG	Length
Remote In	J7	24 VDC	18 AWG	50 ft
Remote Out	J8	24 VAC or 24 VDC	18 AWG	50 ft
Sensor Inputs	J1–J6	24 VDC	18 AWG	10 ft

1.17.4 Operating Ranges and Cooling Requirements

The following sections provide the environmental specifications and cooling requirements for CONVEX equipment.

1.17.4.1 Temperature and Humidity

CAUTION

Maintain the ***recommended*** intake air temperature for CONVEX equipment. Operating the systems in temperatures towards the maximum ranges for even brief periods may result in damage to components and electronic assemblies.

NOTE

Measure all operating temperatures directly in front of the cabinet rather than checking room conditions. Ambient intake air temperature may be up to 5% greater than the ambient room temperature. Operating ranges pertain to the ambient intake air temperature at the front of the cabinet.

The following table lists the temperature and humidity ranges for CONVEX equipment:

Table 1–56, Temperature and Humidity Specifications

Parameter	Maximum Operating Range	Recommended Operating Range	Maximum Rate of Change	Recommended Maximum Rate of Change
Temperature	60°F to 90°F (15°C to 32°C)	70°F to 80°F (21°C to 26.6°C)	9.0°F/hour (5°C/hour)	3.6°F/hour (2°C/hour)
Humidity	10% to 90% with no condensation	40% to 60% with no condensation	2%/hour	2%/hour

1.17.4.2 Dissipation and Air Conditioning Requirements

The following table lists power dissipation and air conditioning requirements for various CONVEX IDC equipment:

Table 1–57, Dissipation and Air Conditioning Requirements

Equipment	Watts	Btu/hr	Kcals/hr	Tons of Refrigeration
EXP-105 Peripheral Cabinet	2,500	8,532	2,150	0.71
IDC Channel Control Unit	200	683	172	0.06
DKD-501 Disk Drive	147	502	126	0.04
DKD-502 Disk Drive	147	502	126	0.04

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2
Unpacking and Installation

2.1 Overview

Inspection and unpacking are discussed, major components are identified, and installation procedures are provided.

2.2 Electrostatic Discharge (ESD) Damage

The IDC and disk drives can be damaged by Electrostatic Discharges (ESD) during maintenance procedures such as installation. A grounded wrist strap (or other grounding method) must be used when handling all Printed Circuit Boards (PCBs) to prevent ESD damage.

Static charges take place when various objects are separated or rubbed together, often creating a high voltage level. The main factors that determine a voltage level charge are:

- Types of materials
- Relative humidity
- Rate of change or separation

Table 2–1, "Static Charge Levels and Relative Humidity," contains approximate electrostatic charge levels based on various personnel activities and humidity levels:

Table 2–1, Static Charge Levels and Relative Humidity

Personnel Activity[1]	Humidity[2] & Charge Levels (Volts[3])			
	26%	32%	40%	50%
Person walking across linoleum floor	6,150V	5,750V	4,625V	3,700V
Person walking across carpet	18,450V	17,250V	13,875V	11,100V
Person getting up from a plastic chair	24,600V	23,000V	18,500V	14,800V

[1] Source: B. A. Unger, *Electrostatic Discharge Failures of Semiconductor Devices* (Bell Laboratories, 1981).
[2] A high rate of air flow produces higher static charges than a low air flow rate, for the same relative humidity level.
[3] Some data in this table has been extrapolated.

Table 2–2, "Components Susceptibility to ESD Damage," contains a list of various components and their susceptibility to static damage:

Table 2–2, Components Susceptibility to ESD Damage

Susceptibility Ranges of Various Devices Exposed to Electrostatic Discharge (Human Body Model[1])	
Device Type	**Level of ESD Susceptibility (Volts)**
MOSFET	>10
JFET	> 140
CMOS	> 250
Schottky Diodes, TTL	> 300
Bipolar Transistors	> 380
ECL (For Hybrid use, PCB level)	> 500
SCR	> 680

[1] Source: B. A. Unger, *Electrostatic Discharge Failures of Semiconductor Devices* (Bell Laboratories, 1981).

2.3 Inspection

General guidelines for inspecting equipment are provided.

2.3.1 Sales Order Packing Slip

The sales order packing slip lists all equipment shipped from CONVEX. It should be used as a checklist to ensure that all equipment has arrived. Figure 2–1, "Sales Order Packing Slip" shows a typical sales order packing slip:

Figure 2–1, Sales Order Packing Slip

CONVEX

CONVEX COMPUTER CORPORATION
701 PLANO RD. RICHARDSON,TX 75081

SALES ORDER
PACKING SLIP

PACKING SLIP NO.	
SALES ORDER	CUSTOMER PURCHASE ORDER
CHANGE NO.	PAGE NO.

BILL TO: SHIP TO:

	ORDER DATE	BILL TO CUSTOMER	REP. CODE	TAX	TAX%	SHIP TO CUSTOMER CODE	TERMS	SHIP VIA	SALESMAN

LN.	PART NUMBER	DESCRIPTION	ORDER QUANITY	BALANCE DUE	UM	DELIVERY DATE	UNIT PRICE	QUANTITY THIS SHIPMENT	SHIP NO.

COMMENTS

CARRIER	WAYBILL NO.	
CARTONS	TOTAL WEIGHT	

H096009
5/30/89

1. Inspect each item of equipment for any sign of shipping damage as it is unpacked.
2. If equipment damage is found, document and proceed to the next section.

NOTE

Save all packing material until after operational checkout of the equipment. This enables equipment to be returned safely to CONVEX, if required.

2.3.2 Inspection for Damage

All shipping containers have been specially designed to protect their contents under normal shipping conditions. Carefully inspect each carton for signs of shipping damage as it is unpacked. If damage is found after visual inspection, document the damage with photographs and contact the transport carrier immediately. Unpack the equipment as described in the next section.

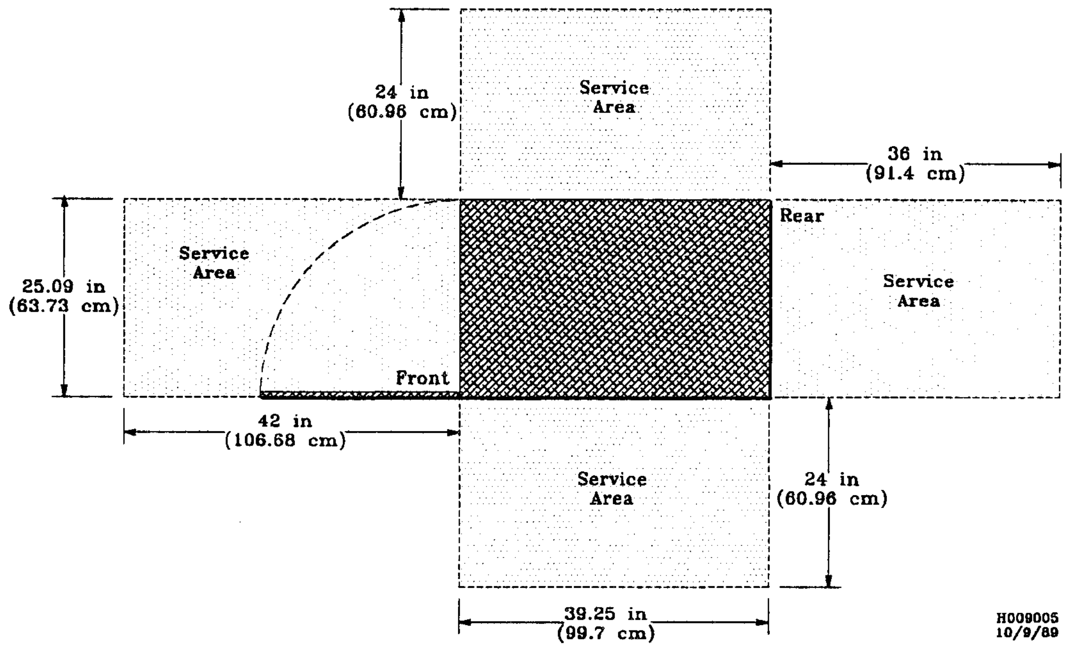
2.3.3 Damage Claims

If the IDC equipment is damaged, a damage claim must be completed. Damage claims should be completed by the customer and given to the shipping representative. Claim forms are normally obtained from the shipping representative.

2.4 Peripheral Cabinet Template

A peripheral cabinet template is provided to show the basic equipment dimensions and space requirements for servicing. The servicing areas are shaded in light gray and are shown as part of the template. The peripheral cabinet service areas are illustrated on four sides. One side of the peripheral cabinet service area is not required, because one end of the peripheral cabinet is joined to the computer. The template for a peripheral cabinet is illustrated in Figure 2-2, "Peripheral Cabinet Template":

Figure 2–2, Peripheral Cabinet Template



Refer to the *CONVEX Computer Site Preparation Guide* for additional site preparation information.

2.5 Unpacking

General guidelines for unpacking the equipment are provided.

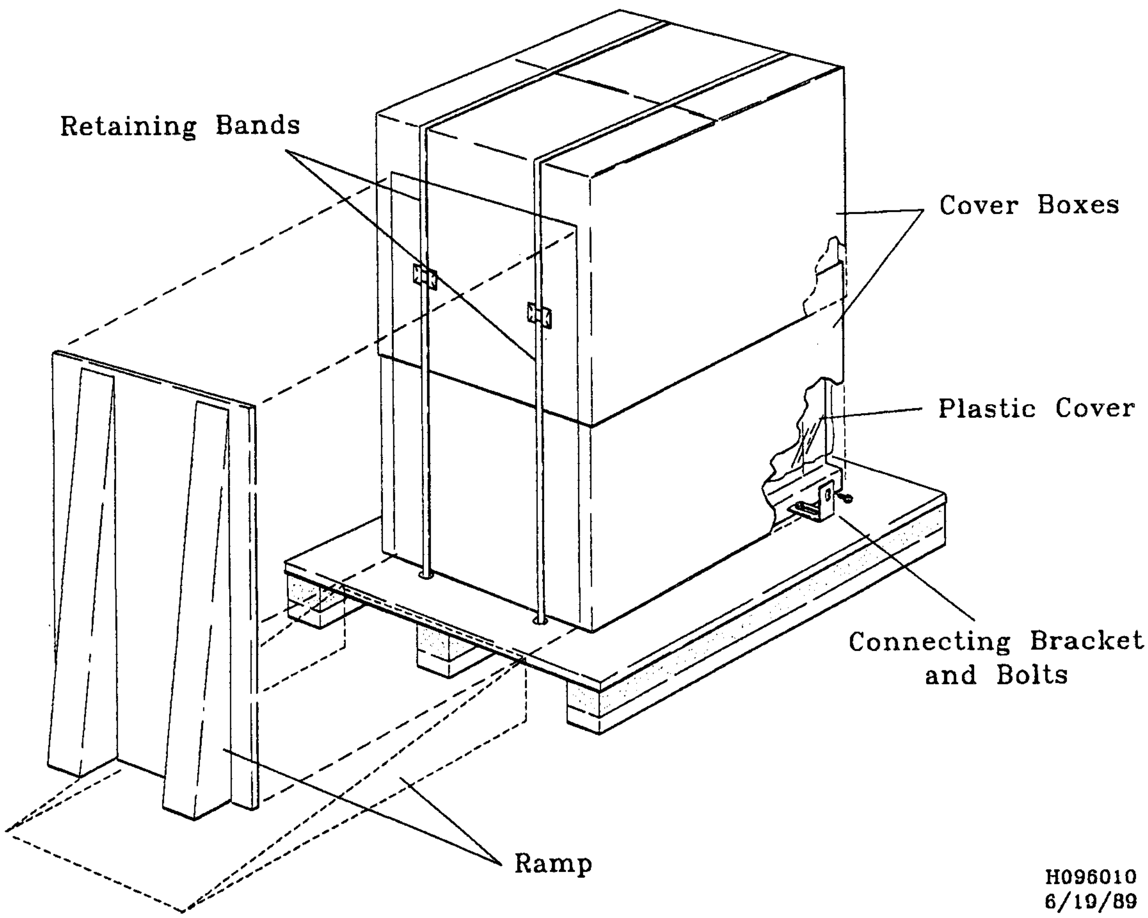
2.5.1 Cabinet

Figure 2–3, "Cabinet Packaging" shows a cabinet with its packaging:

NOTE

For a peripheral cabinet, one ramp is shipped with the peripheral cabinet.

Figure 2–3, Cabinet Packaging



2.5.1.1 Removing Packaging Around the Cabinet

While removing the packaging, visually inspect each cabinet for any sign of shipping damage.

1. Use wire cutters to cut the 2 bands that cross the top of the cabinet box.
2. Remove the ramp from the shipping pallet.
3. Set aside the box containing the trim items.
4. Remove the top cover box, then the lower cover box.

5. Remove the plastic film cover from the cabinet.

2.5.1.2 Removing the Cabinet from the Pallets

Place the pallet in an open area with enough room to connect the ramp to the pallet and to maneuver the cabinet at the foot of the ramp. Figure 2–4, "Cabinet Pallet, Ramp, and Auxiliary Ramp" shows a cabinet mounted on a pallet with a ramp and auxiliary ramp:

Figure 2–4, Cabinet Pallet, Ramp, and Auxiliary Ramp

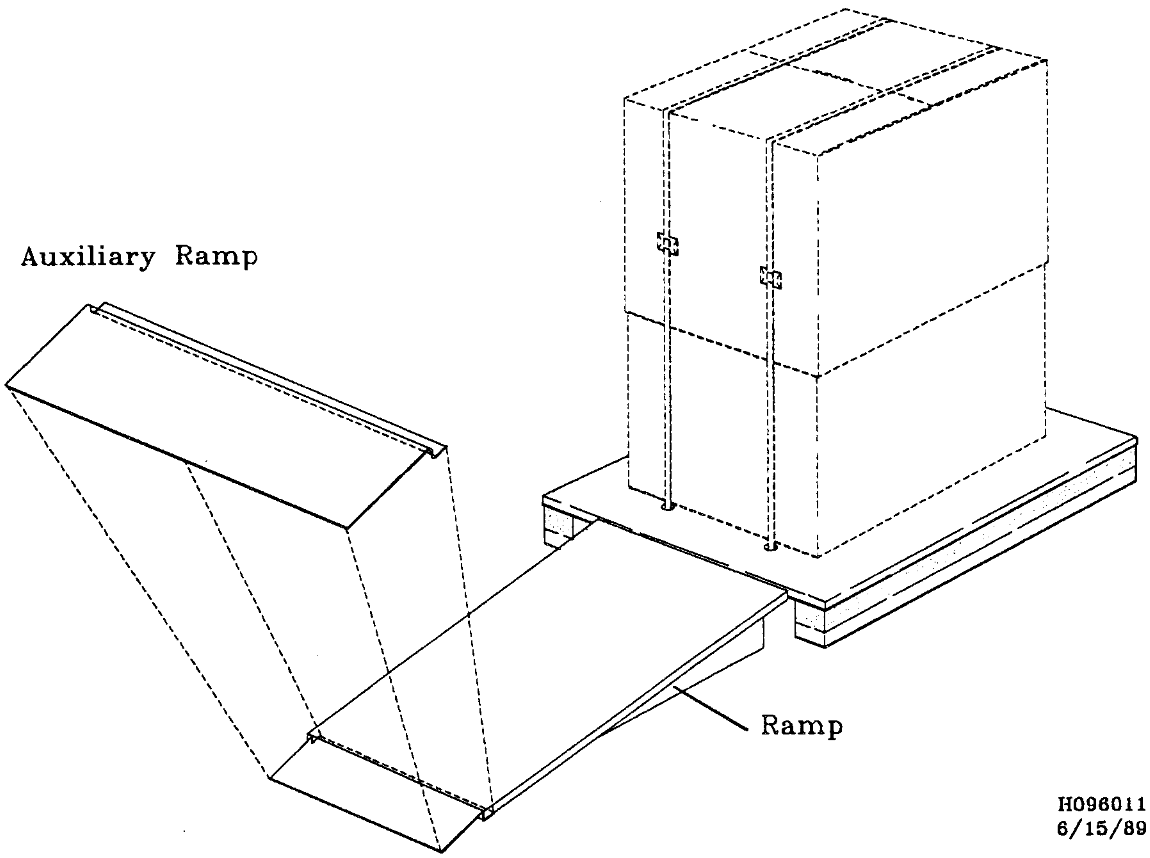
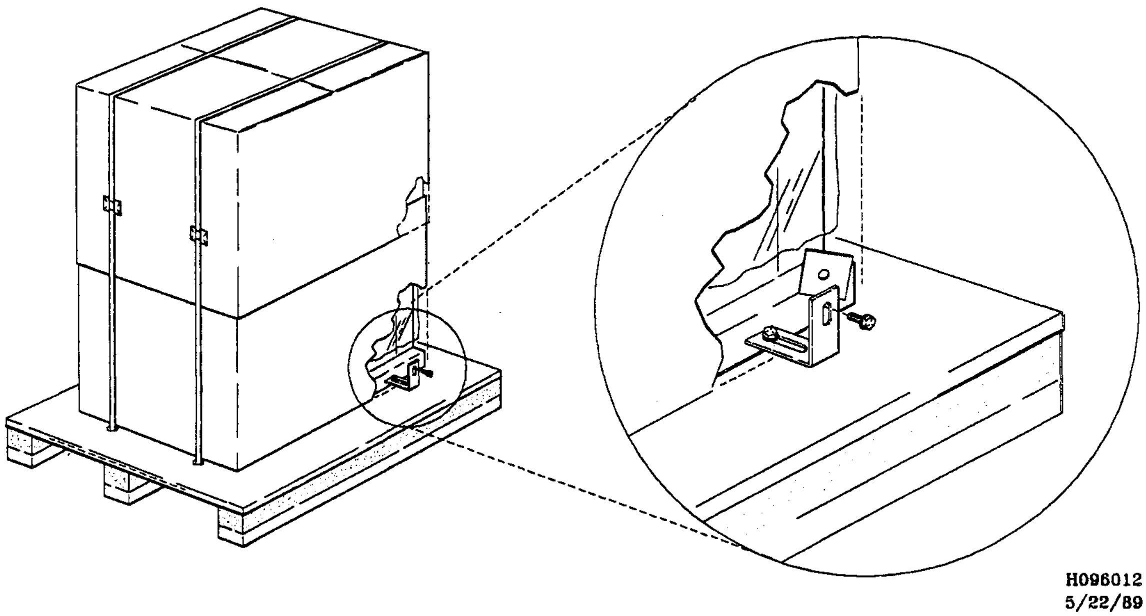


Figure 2–5, "Peripheral Cabinet and Pallet" shows a peripheral cabinet mounted on a pallet with a pallet bracket and spacers:

Figure 2–5, Peripheral Cabinet and Pallet



> **WARNING**
>
> Use care when moving a CONVEX cabinet. Failure to do so may cause injury to personnel and damage to equipment.

CONVEX cabinets are not top-heavy but may tip over when being moved. Because of the weight of the peripheral cabinet, injury to personnel or damage to equipment may result unless two people are available to install the cabinet.

Use the following procedures to remove the peripheral cabinet from the pallet:

1. Slide the legs of the ramp under the front (open) edge of the pallet. The high edge of the ramp should be against the front edge of the pallet.

> **NOTE**
>
> The ramp and pallet height may not be the same. The pallet height can be adjusted by tightening or loosening the carriage bolts that compress the foam cushions built into the pallet.

2. Place the auxiliary ramp at the base of the pallet ramp.
3. Remove the 4 bolts that connect the pallet brackets to the anchor holes of the cabinet.
4. Loosen the bracket bolts that connect to the pallet.
5. Pull the brackets away from the cabinet.
6. Remove the cardboard spacers from between the brackets and the cabinet.

> **CAUTION**
>
> Verify that the 4 feet are raised so the cabinet rests on its wheels. The feet must be completely raised when moving the cabinet. Failure to do so may damage the feet as the cabinet is moved.

7. Roll the cabinet slowly down the ramp, keeping the cabinet centered on the ramp. A person standing on the pallet should guide the cabinet.
8. Verify that cutouts for a raised floor are positioned according to cabinet placement and raised floor design. The cutouts should have channels for routing the external peripheral cables and the AC power cord of the cabinet.
9. Roll the cabinet into position.
10. Verify that the AC power connection point is within 5 feet of the cabinet.
11. Remove the cabinet trim items from the box that was packaged with the cabinet.

2.5.2 Accessories

The accessories include all items that were not shipped on the peripheral cabinet pallet. These items arrive at the site on a separate pallet. Inventory all accessories and inspect for damage while unpacking them.

Use the following procedures to remove the accessories from the pallet:

1. Cut the bands around the accessories on the pallet.
2. Remove the plastic film covering the accessories.
3. Remove each box from the pallet.
4. Assemble the table, and position it near the cabinets.
5. Remove the accessories from their boxes.
6. Place the accessories on the table and ensure they are within cabling distance (less than 6 feet) of the processor cabinet bulkhead next to the peripheral cabinet.
7. Unpack and inventory the cables and the manuals.

2.6 Installation Procedures

> **NOTE**
>
> Before installing an individual IPI disk drive check the voltage selector for the correct setting. The location of the switch is shown in "Figure 4-23, Power Supply Voltage Setting" of this document.

The following sections contain procedures for the initial installation of a peripheral cabinet and IDC.

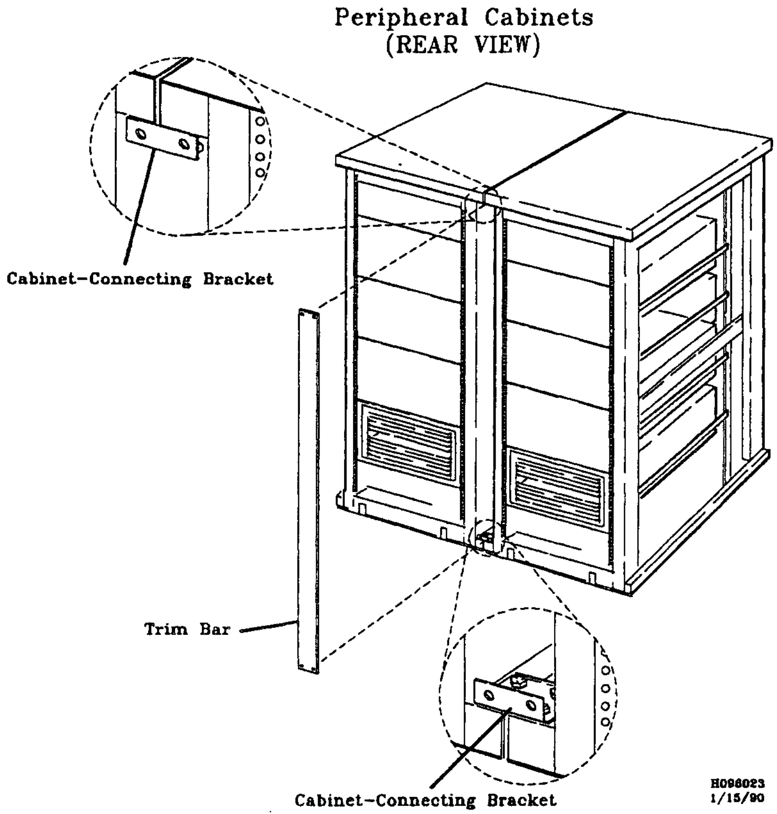
2.6.1 Peripheral Cabinet Installation Procedures

Procedures to install a peripheral cabinet include mating, securing, and cabling the cabinets. Mounting hardware (screws, retaining plates) is used to mount the peripheral cabinet to the processor cabinet.

2.6.1.1 Mating and Securing the Peripheral Cabinets

Before mating the cabinets, first position and level the processor cabinet(s). When the processor cabinet(s) is(are) positioned, secure additional cabinets (to the processor I/O cabinet of a C230 or C240 system) with cabinet connecting-brackets. Figure 2-6, "Cabinet-Connecting Bracket" shows how to position the cabinet-connecting brackets to the peripheral cabinet, with the holes for the trim bar facing the front of the peripheral cabinet:

Figure 2–6, Cabinet-Connecting Bracket



Use the following procedures to secure and level the peripheral cabinet(s) next to the processor cabinet (the processor I/O cabinet of a C230 or C240 system):

1. Verify that the processor cabinet is level. If the cabinet(s) is(are) not level adjust the feet of the cabinet(s) until level.
2. Install a cabinet-connecting bracket, using 2 of the supplied bolts, to the top front and top back of the cabinet bulkhead, just under the cabinet top. Do not tighten the bolts at this time.
3. Install a cabinet-connecting bracket using 2 of the supplied bolts to the bottom front and bottom rear of the next cabinet. Do not tighten the bolts at this time.

> **CAUTION**
>
> Ensure that no cables are in the gutter between the new peripheral cabinet and the existing cabinet when pushing the cabinets together. Failure to do so may result in damage to the cables.

4. Position the peripheral cabinet next to the processor cabinet.
5. Lower the feet of the peripheral cabinet so the wheels are able to spin freely.
6. Adjust the level of the peripheral cabinet at the connecting edge to match the processor cabinet.
7. Level the remaining sides of the peripheral cabinet.
8. Install the remaining 2 bolts, and tighten all bolts on the 4 cabinet connecting brackets between the new peripheral cabinet and the existing cabinet.
9. Recheck the leveling of the cabinets.
10. Position a trim bar between the front of the new peripheral cabinet and the existing cabinet. Align the paint stripe of the bar with the stripe on each of the cabinets.
11. Install the 4 screws that attach the trim bar to the cabinet-connecting bracket. The trim bar has 2 screws at the top and 2 at the bottom.
12. Position a trim bar between the rear of the new peripheral cabinet and the existing cabinet. Align the paint stripe of the bar with the stripe on each of the cabinets.
13. Install the 4 screws that attach the trim bar to the cabinet-connecting brackets. The trim bar has 2 screws at the top and 2 at the bottom.

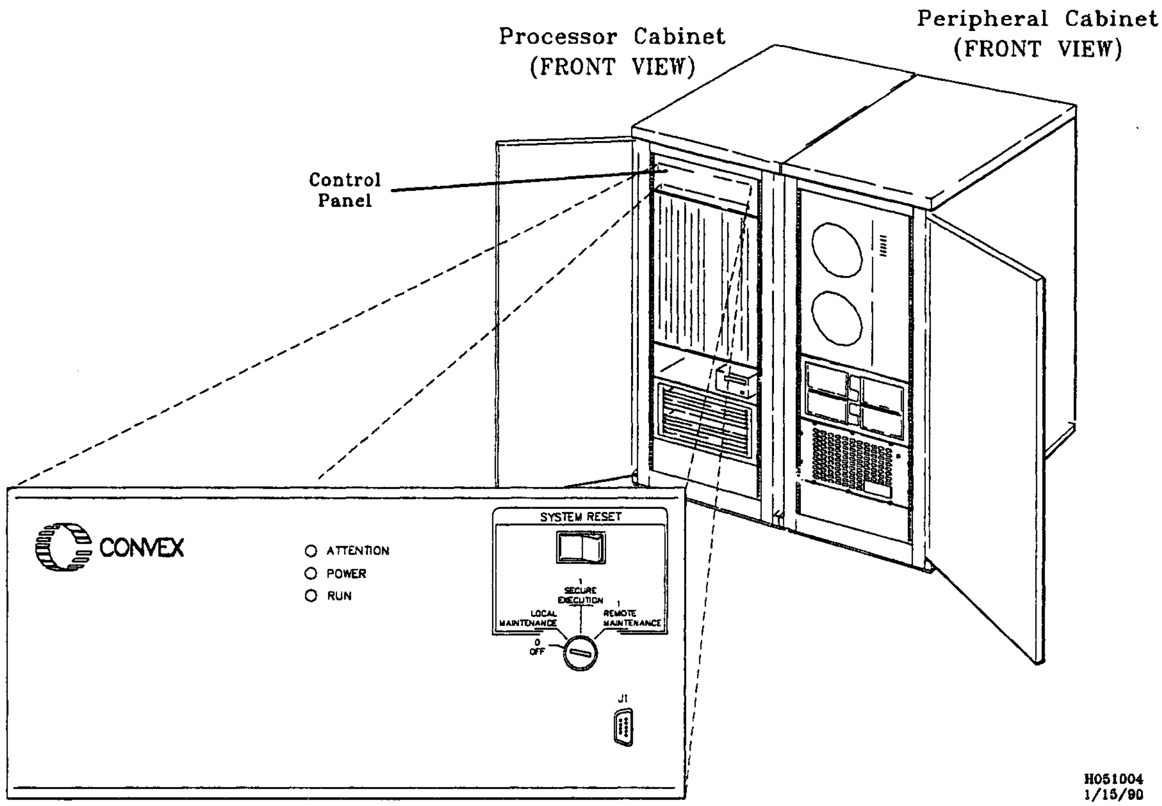
2.6.2 IDC Installation Procedures

The IDC is installed in an available Channel Control Unit (CCU) slot in the computer's card cage. The following section describes the installation of the IDC.

> **CAUTION**
>
> Failure to remove power before installing the IDC board assembly will damage electronic components on the IDC board assembly. Refer to the *CONVEX Processor Operation Guide (C100 Series, C200 Series)* for the shutdown procedures for a CONVEX computer.

1. Turn the processor's front control panel key switch to the **OFF** position as shown in Figure 2–7, "Typical Front Panel Power Control Switch":

Figure 2–7, Typical Front Panel Power Control Switch

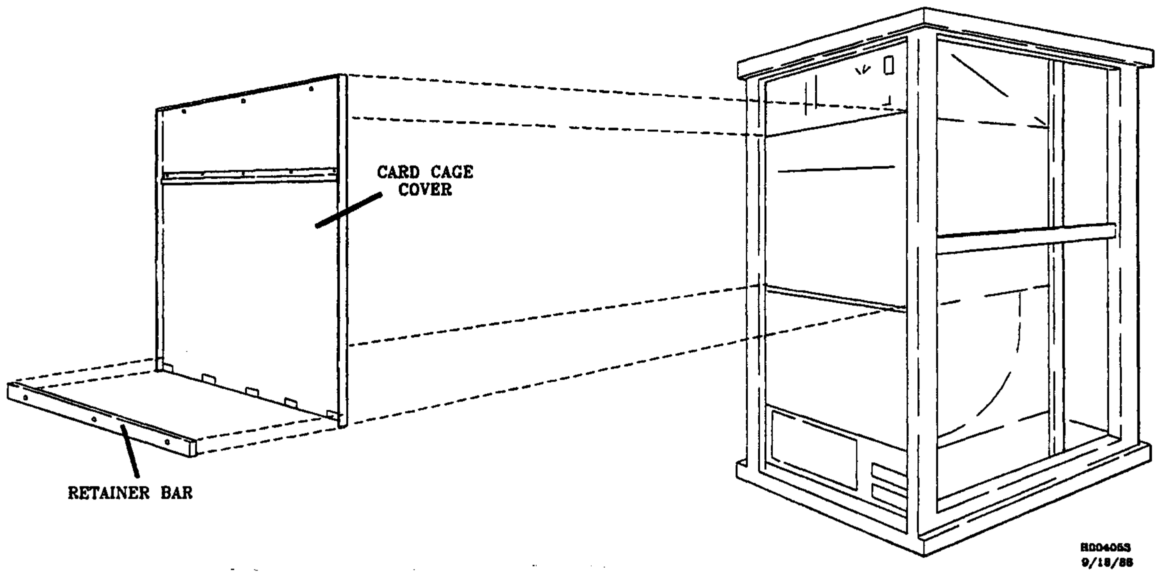


2. Set the power controller's main circuit breaker to the **OFF** position and set the LOCAL/REMOTE switch to the **OFF** position.

> **CAUTION**
>
> The IDC board assembly can be damaged by Electrostatic Discharge (ESD). A grounded wrist strap (or other grounding method) must be used when handling the board assembly to prevent ESD damage.

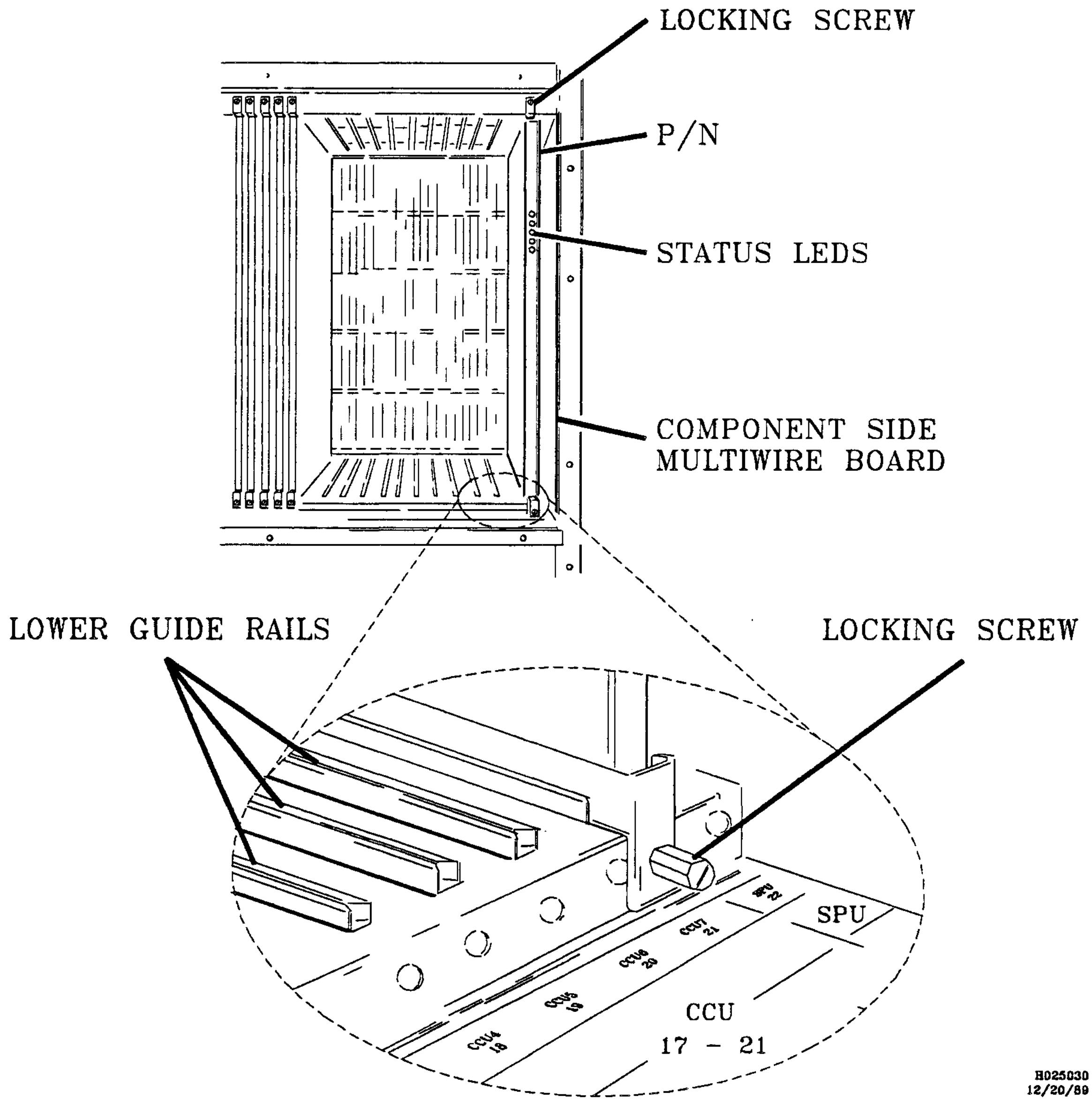
3. Remove the cover plate on the front of the processor card cage as shown in Figure 2–8, "Card Cage Cover Plate":

Figure 2–8, Card Cage Cover Plate



4. Install the IDC board assembly in one of the available **CCU** slots in the processor cabinet's logic rack as shown in Figure 2–9, "CCU Slots and Mounting Hardware":

Figure 2–9, CCU Slots and Mounting Hardware



CAUTION

Failure to tighten the two chassis locking screws simultaneously may damage the connector or result in a faulty connection.

5. Using 2 nut drivers, simultaneously tighten the 2 chassis locking screws on the ends of the IDC board assembly as shown in the previous figure.

6. Replace the cover plate on the front of the processor card cage.

2.6.3 Peripheral Cabinet Cabling Procedures

The following sections contain procedures for the installation of peripheral cabinet cabling. These procedures includes backplane to bulkhead cabling, bulkhead to drive cabling, power sequencing control cabling, and AC power cabling.

Interconnect cables connect the disk drives to the Channel Control Unit (CCU) mounted in the processor cabinet card cage. Four ribbon cables and four IPI cables are used to connect the initial four disk drives to the IDC. Four ribbon cables connect to four backplane port connectors (sets of pins) on the backplane and four ribbon cable connectors on two IPI interface boards at the bulkhead. Four IPI cables connect to four IPI connectors on the IPI interface boards at the bulkhead and four initial disk drives in the peripheral cabinet.

2.6.3.1 Backplane to Bulkhead Cabling

Four ribbon cables connect directly to the backplane in the processor cabinet. The cables connect to the backplane port connectors P2, P3, P4, and P5 on the processor cabinet backplane and also to four ribbon cable connectors on two IPI interface boards at the bulkhead.

The ribbon cables include the following:

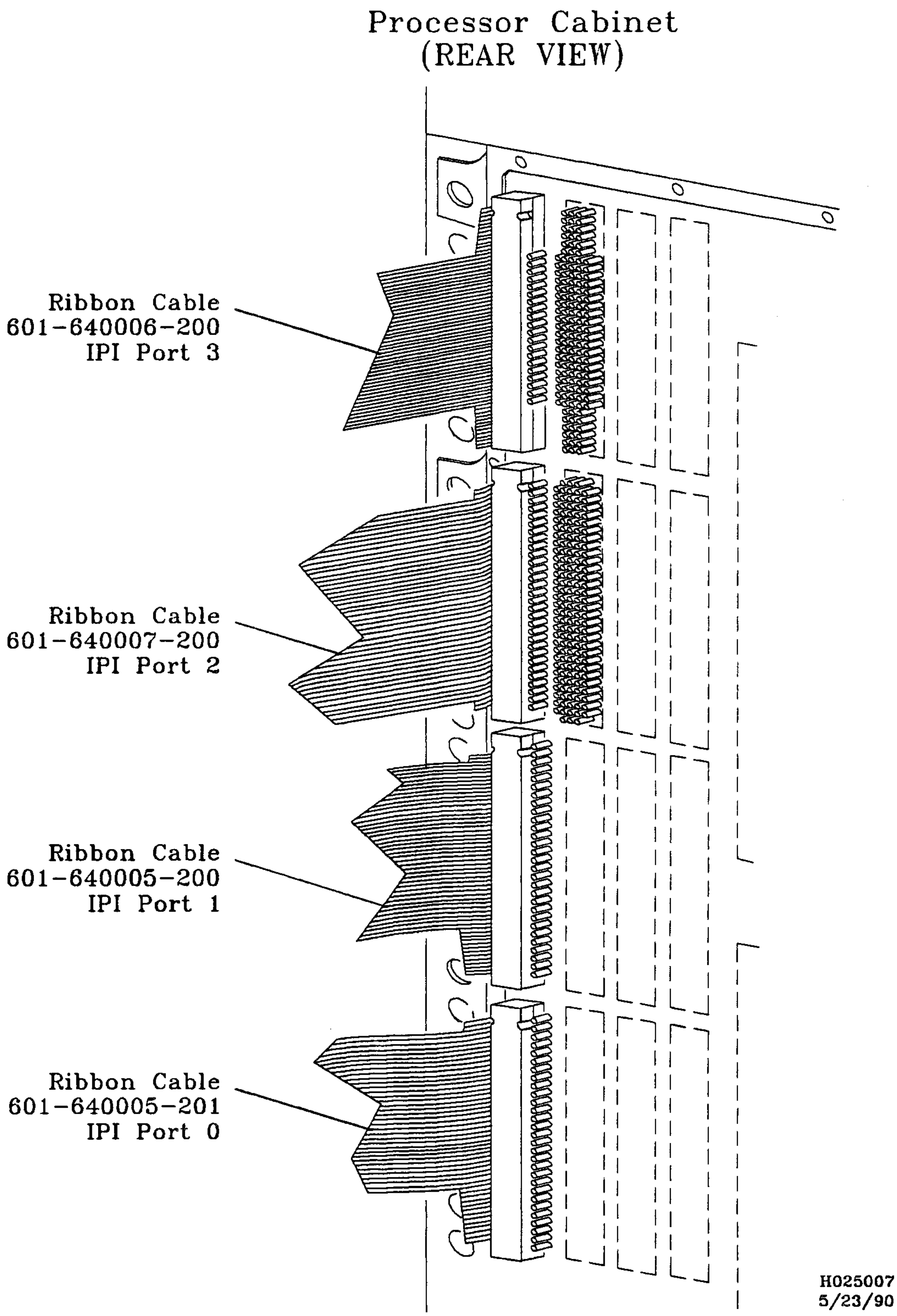
- **601-640006-200**—40 strand ribbon cable
- **601-640007-200**—60 strand ribbon cable
- **601-640005-200**—50 strand ribbon cable
- **601-640005-201**—50 strand ribbon cable (notched bulkhead connector)

Use the following procedures to install the ribbon cables:

1. Remove the rear panel of the peripheral cabinet.
2. Remove the rear panel of the processor cabinet (processor I/O cabinet on C230 or C240 systems).
3. Engage personal grounding system.

> **NOTE**
>
> A flashlight may be required to locate the proper connection point on the processor cabinet backplane.
>
> Depending on the processor backplane, the top and bottom row of 3 pins in a set may or may not be present.

Figure 2–10, Backplane Cabling Connections



4. Connect each ribbon cable to the corresponding set (middle three columns) of backplane port connector pins on the processor cabinet backplane as shown in Figure 2–10, "Backplane Cabling Connections".

> **NOTE**
>
> To install an IPI interface board, two vertically adjacent bulkhead slots must be open to accept the two IPI cable connectors on an IPI interface board.

5. Remove two pair of vertically adjacent bulkhead cover panels from the bulkhead located on the side of the processor cabinet.
6. Install a connection plate (CONVEX P/N 320-000299-500) on each open bulkhead slot.
7. From the processor side of the bulkhead, pass the IPI cable connectors on the IPI interface board (CONVEX P/N 411-000202-200) through the openings in the upper two connection plates. Align the nutplates on the IPI cable connectors with the slots in the connection plates and install the locking screws.
8. From the processor side of the bulkhead, pass the IPI cable connectors on the IPI interface board (CONVEX P/N 411-000201-200) through the openings in the lower two connection plates. Align the nutplates on the IPI cable connectors with the slots in the connection plates and install the locking screws.
9. Connect the ribbon cables to the ribbon cable connectors on the two IPI interface boards as shown in Figure 2–11, "Bulkhead Cabling Connections":

2.6.3.2 Bulkhead to Drive Cabling

Use the following procedure to install the IPI cables:

> **NOTE**
>
> An IPI cable connects the initial disk drive in a series of daisy-chained disk drives to a bulkhead IPI connector.

1. Connect the four IPI cables from the four initial disk drives to the four IPI connectors on the bulkhead as shown in Figure 2–12, "Bulkhead IPI Cable Connections":

Figure 2–11, Bulkhead Cabling Connections

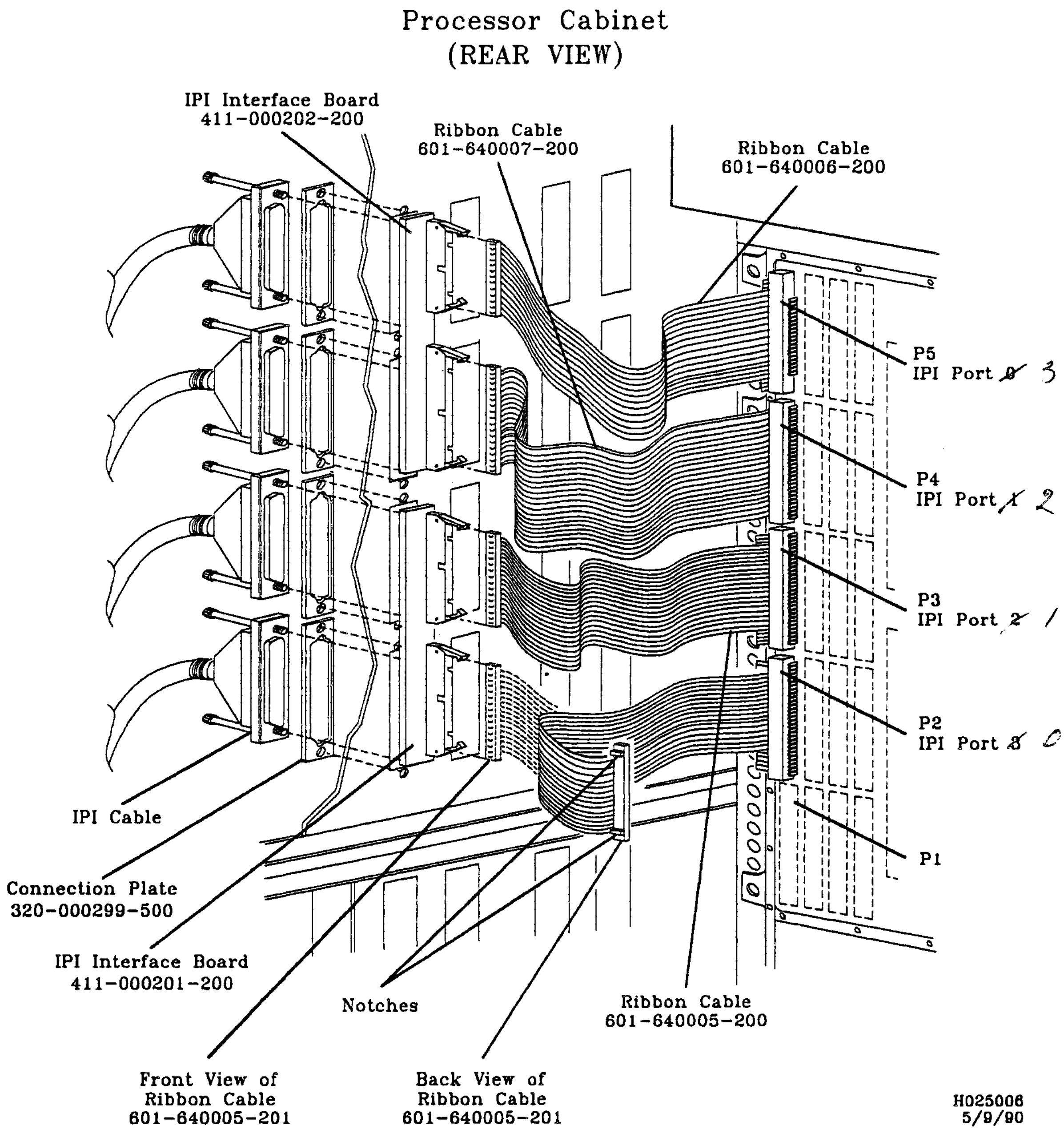
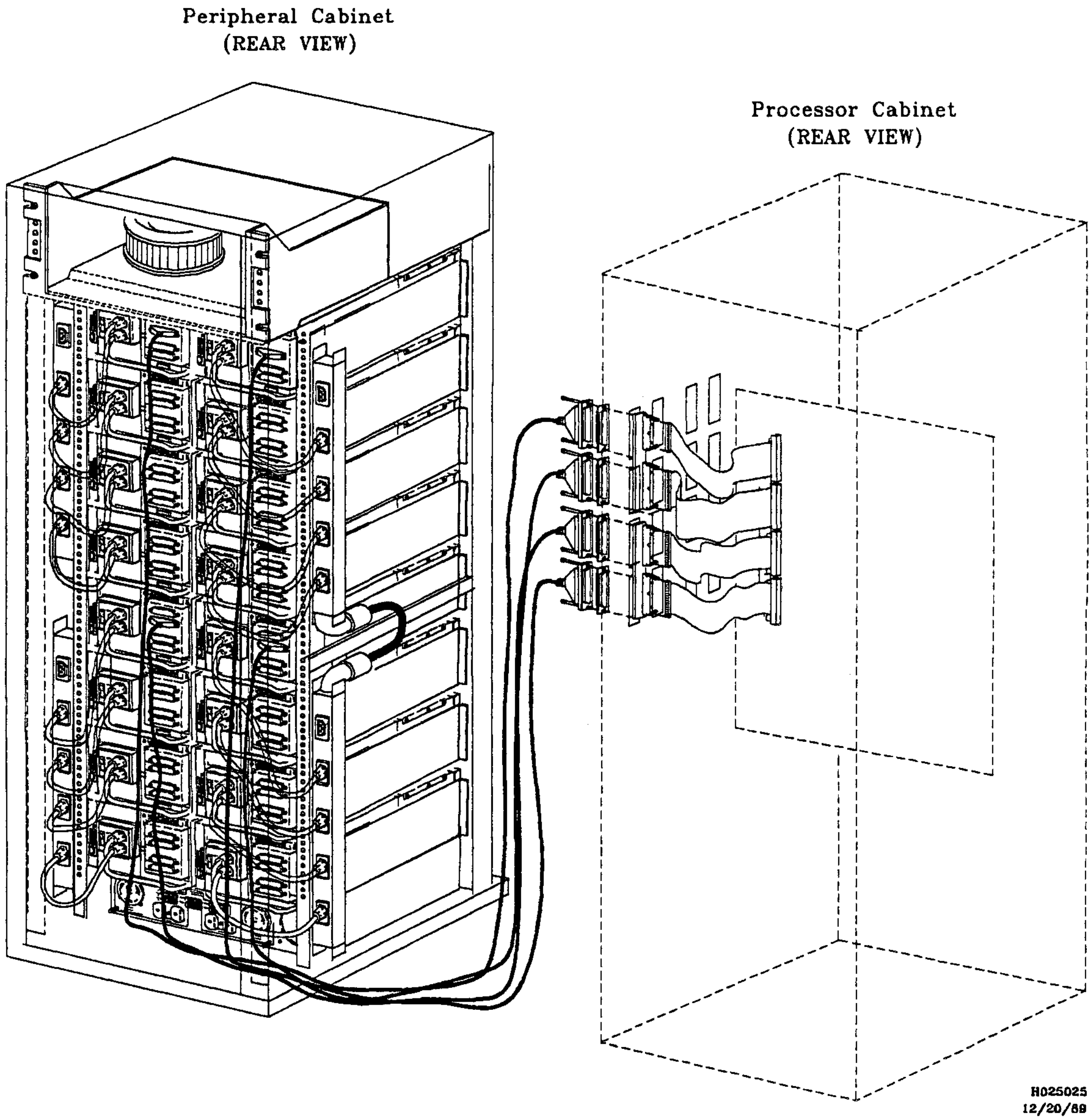


Figure 2–12, Bulkhead IPI Cable Connections



2.6.3.3 Daisy-Chain and Spindle Synchronization Cabling

Use the following procedure to install the daisy-chain cables and the spindle synchronization cable on each disk drive.

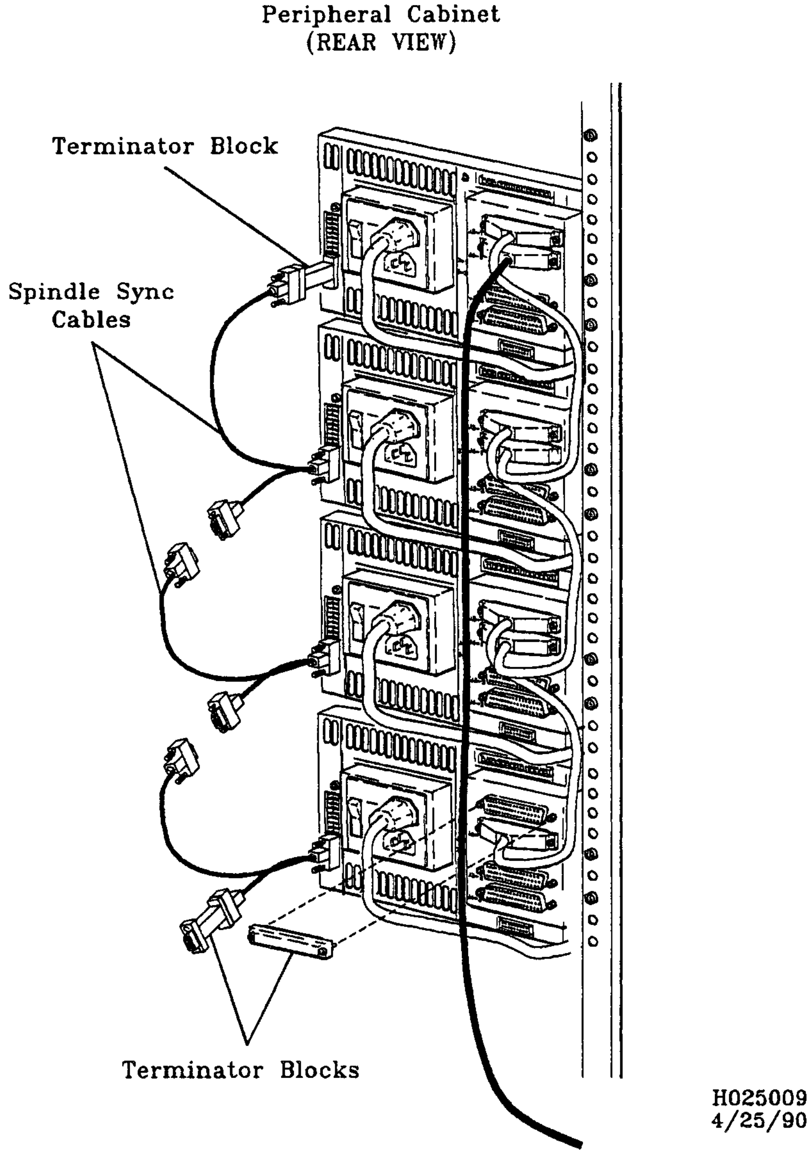
1. Connect the data input cable to the **J4-1** connector and the output data cable to the **J3-1** connector on the rear of the disk drive. The data input cable comes from either the **J3-1** connector of the previous disk drive or from the IPI bulkhead connector. The output data cable will go to the **J4-1** of the next disk drive in the daisy-chain. If there are no other disk drives in the daisy-chain, a terminator connector is placed on the **J3-1** connector of the last disk drive.

> **NOTE**
>
> The spindle synchronization cable is used *only* on the 6-Mbyte/second IDC disk drives. The 3-Mbyte/second IDC disk drives do not use this feature.

2. Connect a terminator to the spindle synchronization connector on the lower left rear of the first (master) disk drive. Connect the spindle synchronization cable to the terminator connector and to the next disk drive. Connect the remaining disk drives in the cabinet in this manner. When there are no other disk drives in the daisy-chain, a terminator is placed on the next unused connector on the spindle synchronization cable.

Figure 2–13, "Daisy-Chain and Spindle Synchronization Cabling," shows the daisy-chain cabling between two 6-Mbyte disk drives, the terminator connector, and the spindle synchronization cabling:

Figure 2–13, Daisy-Chain and Spindle Synchronization Cabling



2.6.3.4 Power Sequencing Control Cabling

The power sequencing control cable allows control of peripheral cabinet AC power from the power controller. The power controller provides 220 VAC to the peripheral devices. Use the following procedures to install the power sequencing control cable:

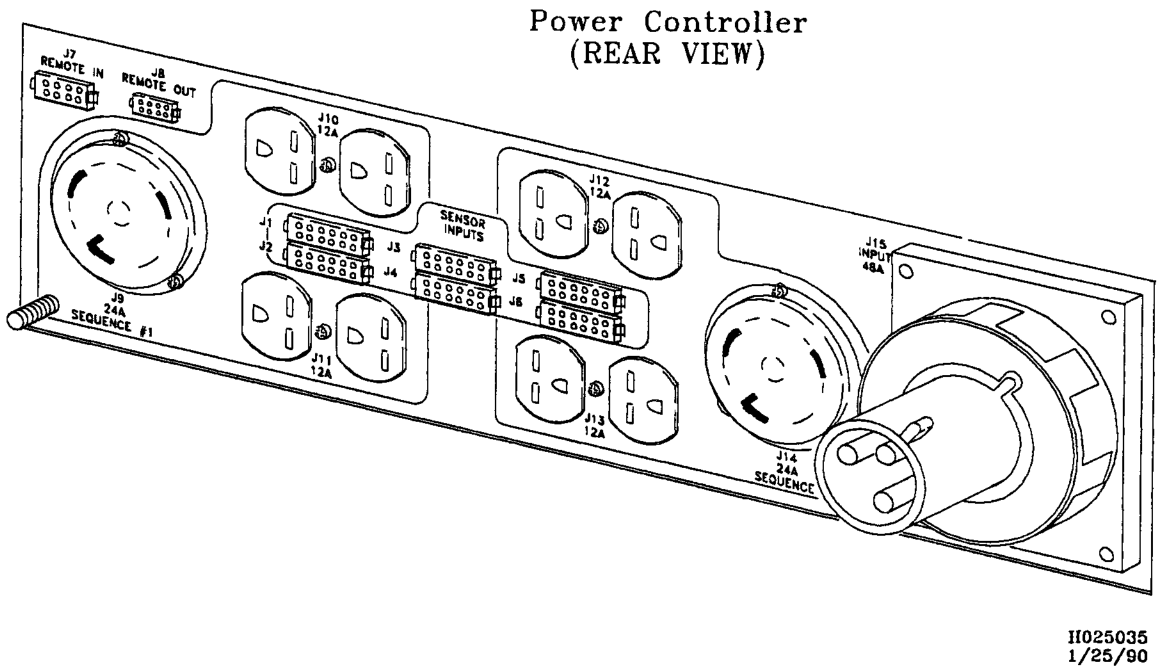
1. Verify that the peripheral devices are plugged into the power strips and the power strips are plugged into the power controller.

NOTE

Installing a remote power sequencing control cable is optional.

2. Plug the remote power sequencing control cable from the bulkhead of the processor cabinet into the J7 REMOTE IN connector on the rear of the power controller as shown in Figure 2–14, "Power Controller Remote In Connection":

Figure 2–14, Power Controller Remote In Connection



3. Install the rear panel of the peripheral cabinet.
4. Install the rear panel of the processor cabinet (processor I/O cabinet on C230 or C240 systems).

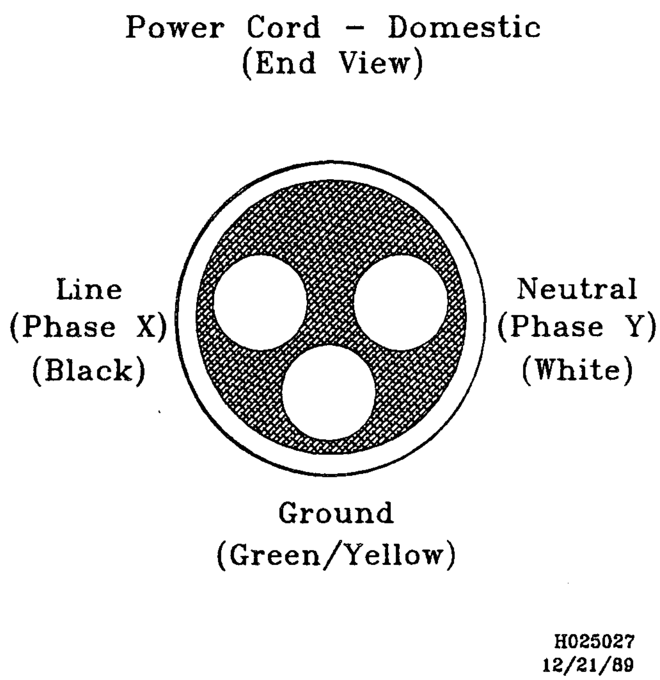
2.6.3.5 AC Power Cabling, Domestic

The following information is provided for reference when installing a site AC power cord and connecting the power controller AC power cord to the site AC power cord.

> **NOTE**
>
> The domestic AC power cord has a HUBBELL Part Number 360C6W receptacle on the power controller end and a 360P6W plug on the opposite end. A receptacle, HUBBELL Part Number 360C6W (CONVEX Part Number 304-000036-001), must be installed on the site AC power cord.

The conductor configuration for a domestic AC power cord is shown in Figure 2–15, "Power Cord Conductor Configuration – Domestic":

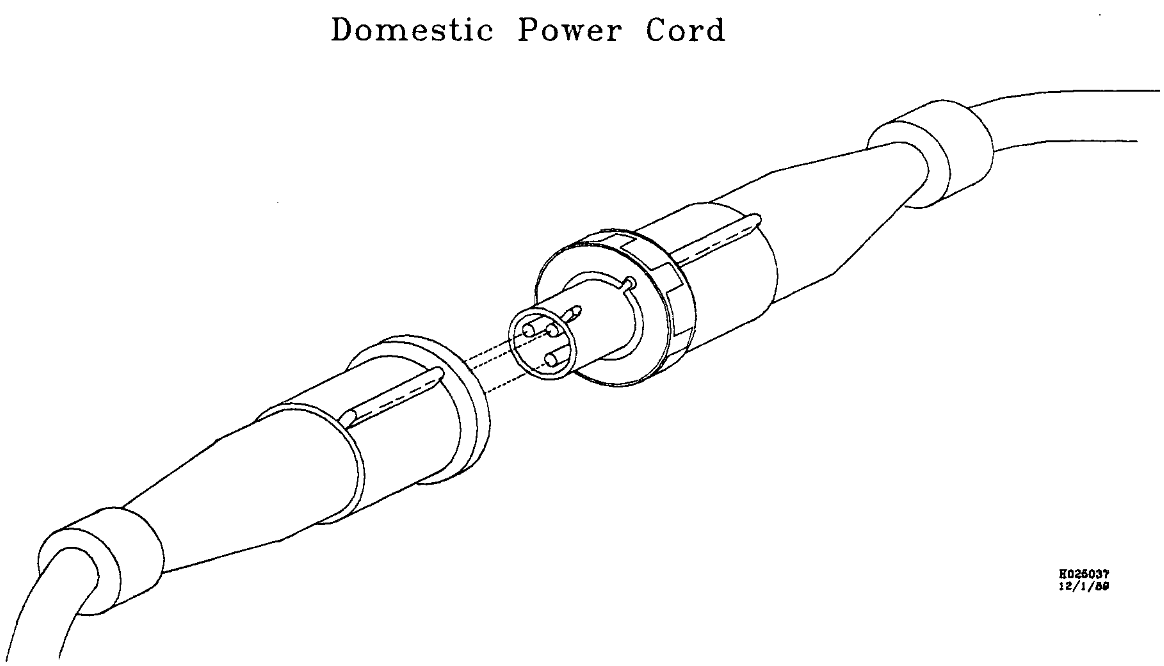
Figure 2–15, Power Cord Conductor Configuration – Domestic



The power controller AC power cord should be connected to the site AC power cord according to the following procedures:

1. Ensure the power controller's main circuit breaker is set to the **OFF** position.
2. Install the receptacle on the site AC power cord.
3. Connect the site AC power cord to a site AC power source at a site power distribution panel.
4. Connect the power controller AC power cord to the site AC power cord as shown in Figure 2–16, "Power Cord Connection - Domestic":

Figure 2–16, Power Cord Connection - Domestic



5. Set the power controller's main circuit breaker to the **ON** position and select local or remote operation with the LOCAL/REMOTE switch.
6. Turn the processor's front control panel key switch to the **ON** position.

2.6.3.6 AC Power Cabling, International

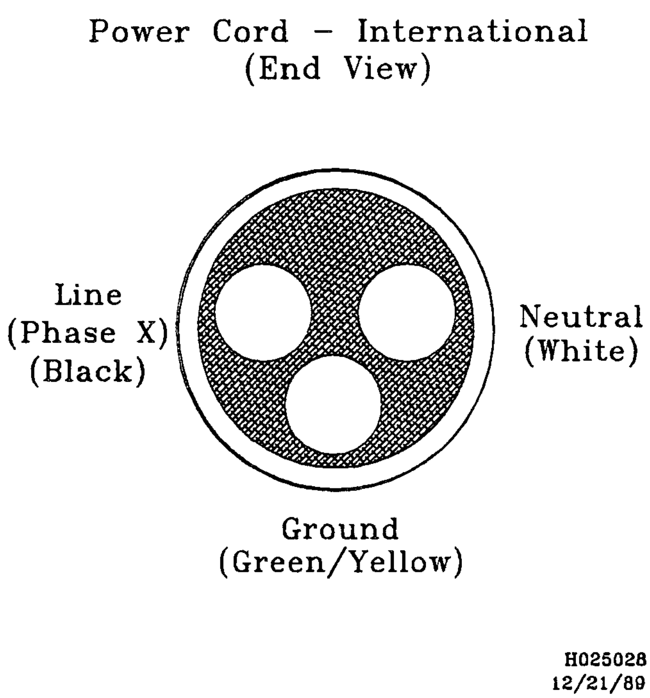
The following information is provided for reference when connecting the power controller AC power cord to a site power distribution panel.

NOTE

The international AC power cord has a HUBBELL P/N 360C6W receptacle on the power controller end but does not have a plug on the opposite end.

The conductor configuration for an international AC power cord is shown in Figure 2–17, "Power Cord Conductor Configuration - International":

Figure 2–17, Power Cord Conductor Configuration - International



The power controller AC power cord should be connected to the site power distribution panel according to the following procedures:

WARNING

Failure to set the power controller's main circuit breaker to **OFF** before connecting the power cord may cause injury to personnel.

1. Ensure the power controller's main circuit breaker is set to the **OFF** position.
2. Connect the power controller AC power cord to a site AC power source at a site power distribution panel.
3. Set the power controller's main circuit breaker to the **ON** position and select local or remote operation with the LOCAL/REMOTE switch.
4. Turn the processor's front control panel key switch to the **ON** position.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3
Integration and Test

3.1 Overview

This chapter contains guidelines for integrating an Integrated Disk Channel (IDC) into the CONVEX Operating System (ConvexOS), a brief description of disk striping and spindle synchronization, and information about the IDC diagnostic program.

IDC modules, Intelegent Peripheral Interfaces (IPIs), device types, and unit types must be integrated into ConvexOS before they can be used. How they are integrated depends on the type of performance or features required.

3.2 Disk Striping and Spindle Synchronization

Striped disk partitions are logical disk partitions that are interleaved over several physical disk partitions. Striped disk partitions are used to take advantage of performance improvements made possible by the parallel operation of the several disk arms that make up the striped disk partition. UNIX file systems can be mounted on striped disk partitions just as with "normal" disk partitions. Striping should only be done with disk drives of the same type. Striping across a mixture of 3-Mbyte/second disk drive and 6-Mbyte/second disk drives will operate at only 3 Mbyte/second.

NOTE

The spindle synchronization cable is used *only* on the 6-Mbyte/second IDC disk drives. The 3-Mbyte/second IDC disk drives are not equipped to use this feature.

Spindle synchronization improves the performance of striped file systems on multiple disk drives. The 6-Mbyte/second IDC drive is the only IDC device that supports spindle synchronization.

When spindle synchronization is used, the slave drives in a spindle synchronized chain are synchronized with the master drive. For optimum performance, daisy chain drives from each IPI port and cable all the drives for spindle synchronization with one master drive. For example, a spindle synchronized chain could consist of a master unit (drive, or unit, 0 on IPI port 0) and three slave units (drive, or unit, 0 from each set of drives daisy chained to ports 1–3).

3.3 Software Integration

The operating system for CONVEX C200 Series computers contain all software drivers for the hardware supported by CONVEX. This means that a system generation is not required when an IDC is installed in a CONVEX computer.

> **NOTE**
>
> The operating system release for CONVEX C100 Series computers do not contain an IDC device driver.

System-level hardware is identified to ConvexOS via a configuration file (*/ioconfig*) located on the Service Processor Unit (SPU) disk. The */ioconfig* file describes, in hierarchical fashion, the connections between IDC modules, IPI interfaces, device types, unit types. The operating system uses this information to assign logical and physical device numbers to a device of a given type.

If the *master* keyword is present on the end of the unit line, then the unit will be configured as a master unit; otherwise, the unit is a slave unit. A slave unit's clock input may or may not be connected to a master unit's clock output. An unclocked slave is *free running.* Each type of device is identified to the operating system by a mnemonic device code. The device codes for IDC system are listed below:

- DKC-IP2 — IDC IPI controller
- DKD-501 — 3-Mbyte/second IDC disk drive
- DKD-502 — 6-Mbyte/second IDC disk drive

These codes, and other information, are entered into the */ioconfig* file on the SPU disk. This file contains entries, such as IDC (CCU slot) number, IPI port number, device type, and unit type. A typical */ioconfig* file including entries for an IDC with three 3-Mbyte/second 1-Gbyte drives (one master drive and two slave drives) and three 6-Mbyte/second 1-Gbyte drives (one master drive and three slave drives) is shown in Figure 3–1, "Example */ioconfig* File":

Figure 3–1, Example */ioconfig* File

```
iop 3
    mbus 0
      ctrl DKC-001 csr 0x3f0 int 2
        unit 0 type DKD-005
viop 4
    vme 0
      ctrl DKC-203 csr 0x800 int 3
        unit 0 DKD-214
    vme 1
      ctrl DKC-203 csr 0x800 int 3
        unit 0 DKD-214
        unit 1 DKD-214
idc z
    ipi 0
      drvr DKC-IP2
          unit 0 type DKD-502 master
          unit 1 type DKD-501
    ipi 1
      drvr DKC-IP2
          unit 0 type DKD-502
          unit 1 type DKD-501
    ipi 2
      drvr DKC-IP2
          unit 0 type DKD-502
          unit 1 type DKD-501
    ipi 3
      drvr DKC-IP2
          unit 0 type DKD-502
```

NOTE

In addition to the IDC information in an */ioconfig* file, spindle synchronization cables must be installed between the drives in a spindle-synchronized chain.

Whenever an IDC configuration is changed, the information in the hardware section of the configuration file (*/ioconfig)* must be changed, otherwise system operation problems will occur.

NOTE

The booting order of the IDC disk units determines the names of the units in the filesystem */dev* directory.

The *CONVEX System Manager's Guide* should be consulted when making changes to the configuration file.

3.4 Testing the IDC

The IDC and related hardware are tested by the *idc4000* diagnostic program. This program verifies the operation of the IDC.

3.4.1 Using the *idc4000* Diagnostic Program

The *idc4000* diagnostic program is an offline program that must be executed on the SPU while the CPU is halted. The procedures for executing this test are beyond the scope of this manual. However, this information is contained in the *CONVEX PBUS I/O System Diagnostics Manual.* This manual should be consulted before running this test.

Chapter 4
Maintenance Procedures and IPB

4.1 Overview

Guidelines for obtaining technical assistance, and maintenance procedures for the IDC and related peripheral equipment are contained in this chapter. Also, an Illustrated Parts Breakdown (IPB) for all Field Replaceable Units (FRUs) is included.

4.2 CONVEX Technical Assistance

CONVEX offers two sources of help if problems arise:

- CONVEX Technical Assistance Center (TAC)
- CONVEX *contact* utility

4.2.1 CONVEX Technical Assistance Center

Contact the CONVEX Technical Assistance Center (TAC) for real time support on urgent hardware and software problems. The TAC can be reached from all locations in the continental United States by calling 1(800)952-0379, or by calling 1(214)952-4379 from other locations in Alaska, Hawaii, or Canada. From all other locations, contact the nearest CONVEX office.

4.2.2 CONVEX *contact* Utility

Use the CONVEX *contact* utility for reporting minor hardware and software problems. Refer to Appendix A for an example of the CONVEX *contact* utility.

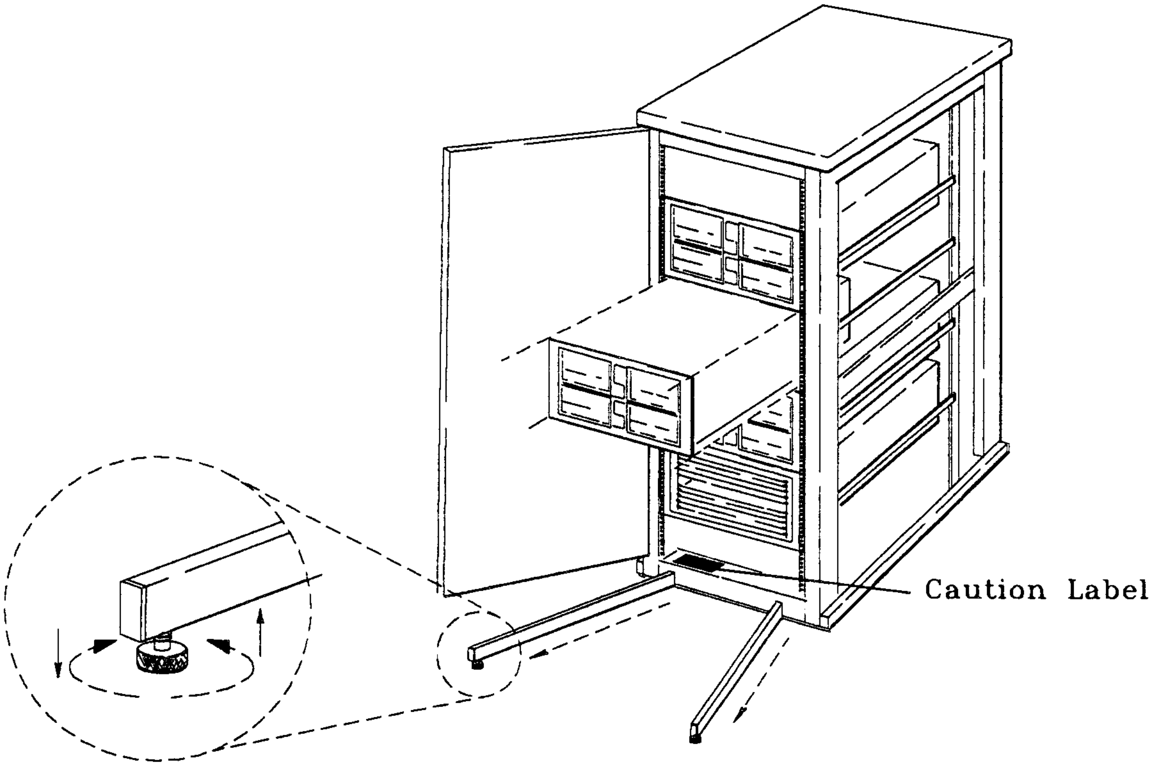
4.3 Maintenance Safety Procedures

Maintenance safety procedures for the IDC and peripheral cabinet apply to most servicing operations. For example, the cabinet stabilizer bars must be extended during most servicing operations. These safety procedures must be used in the maintenance procedures described in Section 4.4.

> **WARNING**
>
> Peripheral cabinet stabilizer bars must be extended prior to extending a drive assembly forward on its slide rails for service. Failure to do so will make the peripheral cabinet unstable, increasing the possibility of it falling forward. This can cause injury to personnel and will cause damage to equipment.

1. Extend the peripheral cabinet stabilizer bars, and adjust the adjustable feet until they are in firm contact with the floor as shown in Figure 4–1, "Peripheral Cabinet Stabilizer Bars":

Figure 4–1, Peripheral Cabinet Stabilizer Bars



CAUTION LABEL

ENGLISH/FRENCH

CAUTION

TO REDUCE RISK OF POSSIBLE INJURY DUE TO UNSTABLE UNIT, ACTUATE STABILIZER BEFORE ANY PERIPHERAL IS EXTENDED.

1. TO ACTUATE STABILIZER, FULLY EXTEND ANTITILT CHANNELS AND LOWER CHANNEL SUPPORT FEET FIRMLY TO THE FLOOR.
2. INSURE THAT LOCKING MECHANISMS ARE INSTALLED IN ALL OTHER EXTENDABLE UNITS.
3. NEVER EXTEND MORE THAN ONE UNIT AT A TIME.

ATTENTION

POUR REDUIRE LE RISQUE D'ACCIDENT ATTRIBUABLE A L'INSTABILITE DE L'UNITE, DEPLOYER LES STABILISATEURS AVANT DE SORTIR LES PERIPHERIQUES.

1. POUR DEPLOYER LES STABILISATEURS, TIRER COMPLETEMENT LES BRAS ANTI-BASCULEMENT ET ABAISER LES PATTES DE FACON QU'ELLES REPOSENT SOLIDEMENT SUR LE SOL.
2. S'ASSURER QUE TOUS LES PERIPHERIQUES SON MUNIS DE VIS DE BLOCAGE.
3. NE JAMAIS SORTIR PLUS D'UN PERIPHERIQUE A UN MOMMENT DONNE.

ENGLISH/GERMAN

CAUTION

TO REDUCE RISK OF POSSIBLE INJURY DUE TO UNSTABLE UNIT, ACTUATE STABILIZER BEFORE ANY PERIPHERAL IS EXTENDED.

1. TO ACTUATE STABILIZER, FULLY EXTEND ANTITILT CHANNELS AND LOWER CHANNEL SUPPORT FEET FIRMLY TO THE FLOOR.
2. INSURE THAT LOCKING MECHANISMS ARE INSTALLED IN ALL OTHER EXTENDABLE UNITS.
3. NEVER EXTEND MORE THAN ONE UNIT AT A TIME.

ACHTUNG

ZUR VERMEIDUNG VON GEFAHRDUNG DURCH EIN INSTABILES GERAT SIND VOR DER HERAUSNAHME VON PERIPHERALS DER STABILISIERUNGSMECHANISMUS BETATIGT WERDEN.

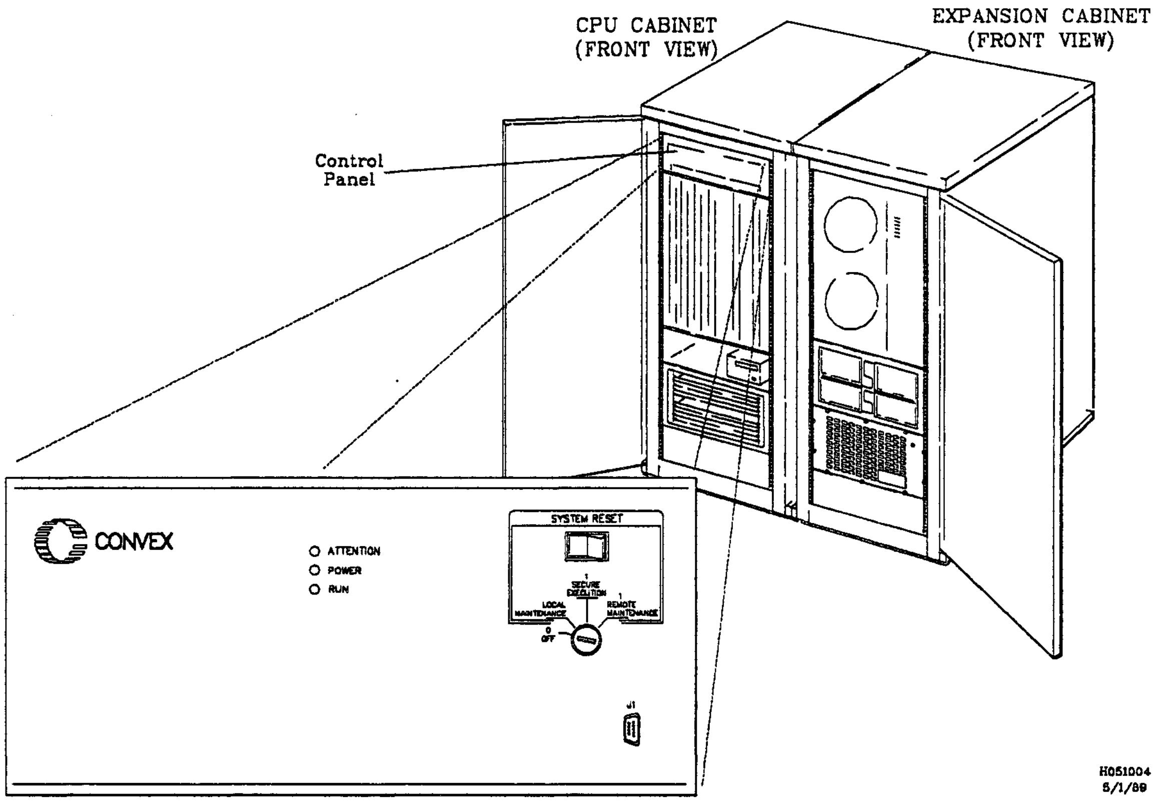
1. UM DIE STABILISIERUNGSEINRICHTUNG ZU BETATIGEN, SIND DER "ANTITILT KANAL" GANZ HERAUS ZU ZIEHEN UND DER UNTERE STUTZFUSS AUF DEN BODEN ZU FUHREN.
2. OBERPRUFEN SIE, OB IN ALLEN ANDEREN VERSCHIEBBAREN GERATEN DER SICHERUNGSMECHANISMUS BETATIGT IST.
3. ZIEHEN SIE NIE MEHR ALS EIN GERAT HERAUS.

H015020
2/7/90

> **CAUTION**
>
> Failure to remove power before installing or removing equipment from the computer card cage will damage electronic components. Refer to the *CONVEX Processor Operation Guide (C100 Series, C200 Series)* for power down procedures on a CONVEX computer.

2. Turn the power control switch on the computer's front control panel to the **OFF** position as shown in Figure 4–2, "Typical Front Panel Power Control Switch":

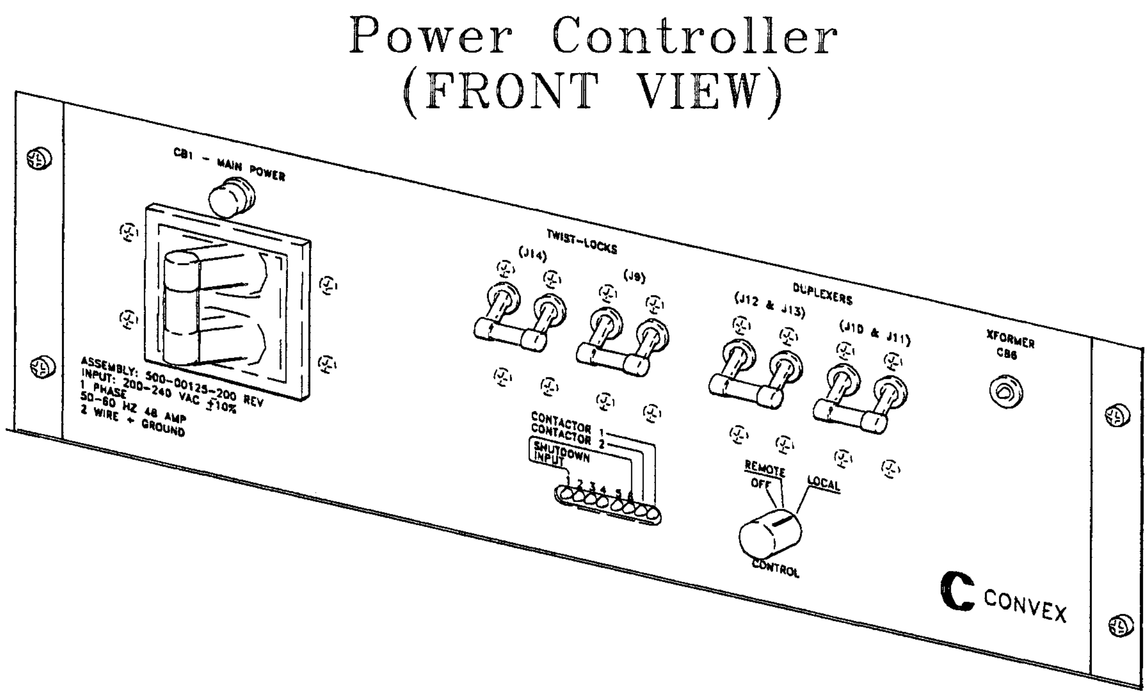
Figure 4–2, Typical Front Panel Power Control Switch



> **CAUTION**
>
> Failure to remove power before installing or removing equipment will damage electronic components.

3. Set the power controller's main circuit breaker to the **OFF** position and set the LOCAL/REMOTE switch to the **OFF** position as shown in Figure 4–3, "Power Controller Switches":

Figure 4–3, Power Controller Switches



H025034
11/22/89

4.4 Maintenance Procedures

This section defines the maintenance procedures for the IDC and equipment in the peripheral cabinet.

4.4.1 IDC

Follow these procedures to remove and replace the IDC.

4.4.1.1 Removal

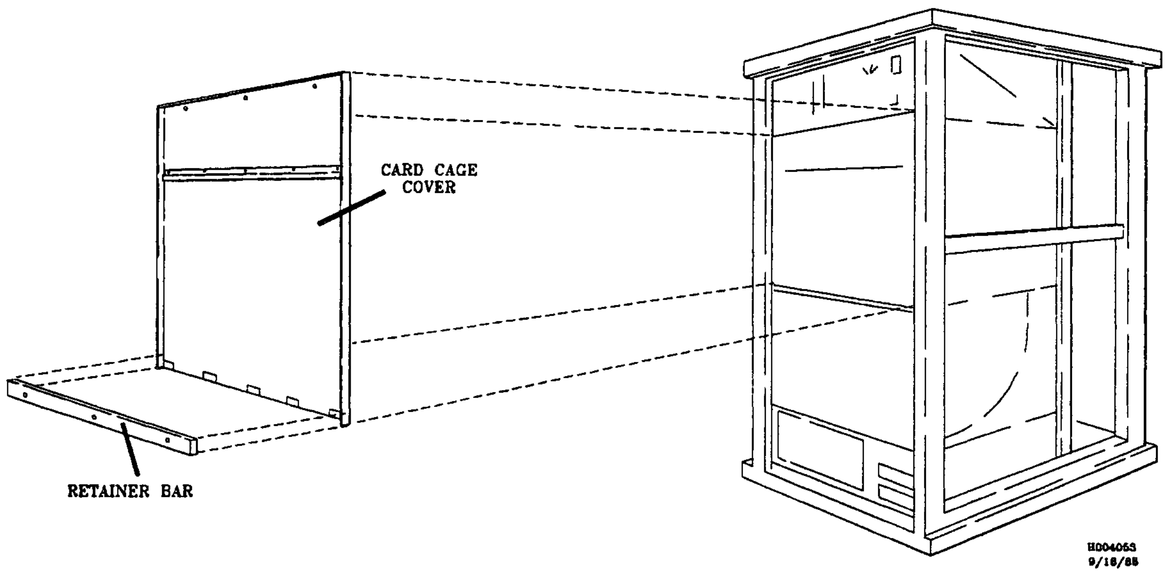
> **CAUTION**
>
> Failure to remove power before removing the IDC will damage electronic components on the IDC. Refer to the *CONVEX Processor Operation Guide (C100 Series, C200 Series)* for the shutdown procedures for a CONVEX computer.

1. Complete steps 1, 2, and 3 in Section 4.3, "Maintenance Safety Procedures."

> **CAUTION**
>
> The IDC can be damaged by Electrostatic Discharge (ESD). A grounded wrist strap (or other grounding method) must be used when handling the IDC to prevent ESD damage.

2. Remove the cover plate on the front of the processor card cage as shown in Figure 4-4, "Card Cage Cover Plate":

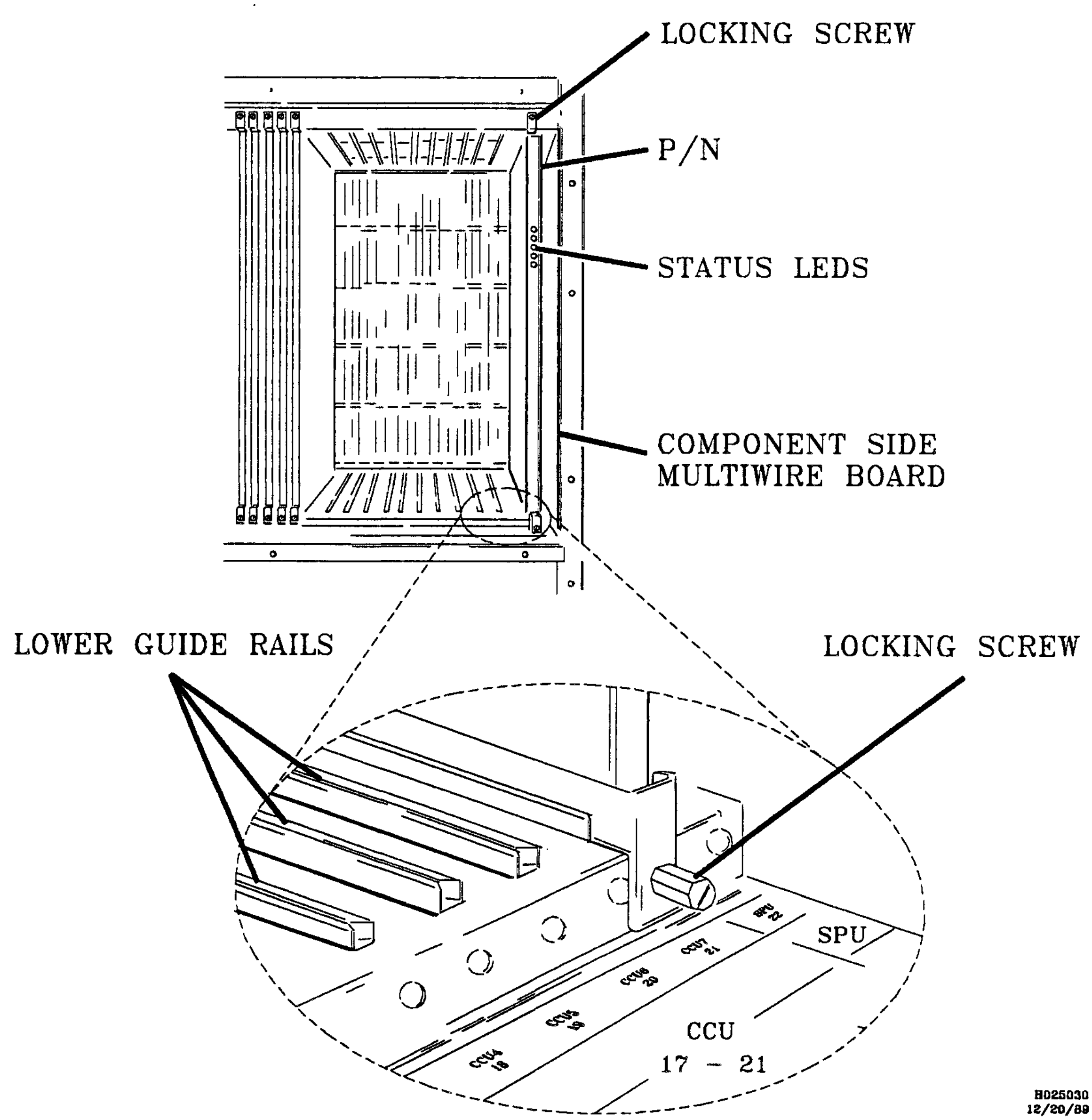
Figure 4–4, Card Cage Cover Plate



> **CAUTION**
>
> Failure to loosen the two chassis locking screws simultaneously may damage a connector.

3. Using 2 nut drivers, simultaneously loosen the 2 chassis locking screws on the ends of the IDC as shown in Figure 4–5, "CCU Slots and Mounting Hardware":

Figure 4–5, CCU Slots and Mounting Hardware



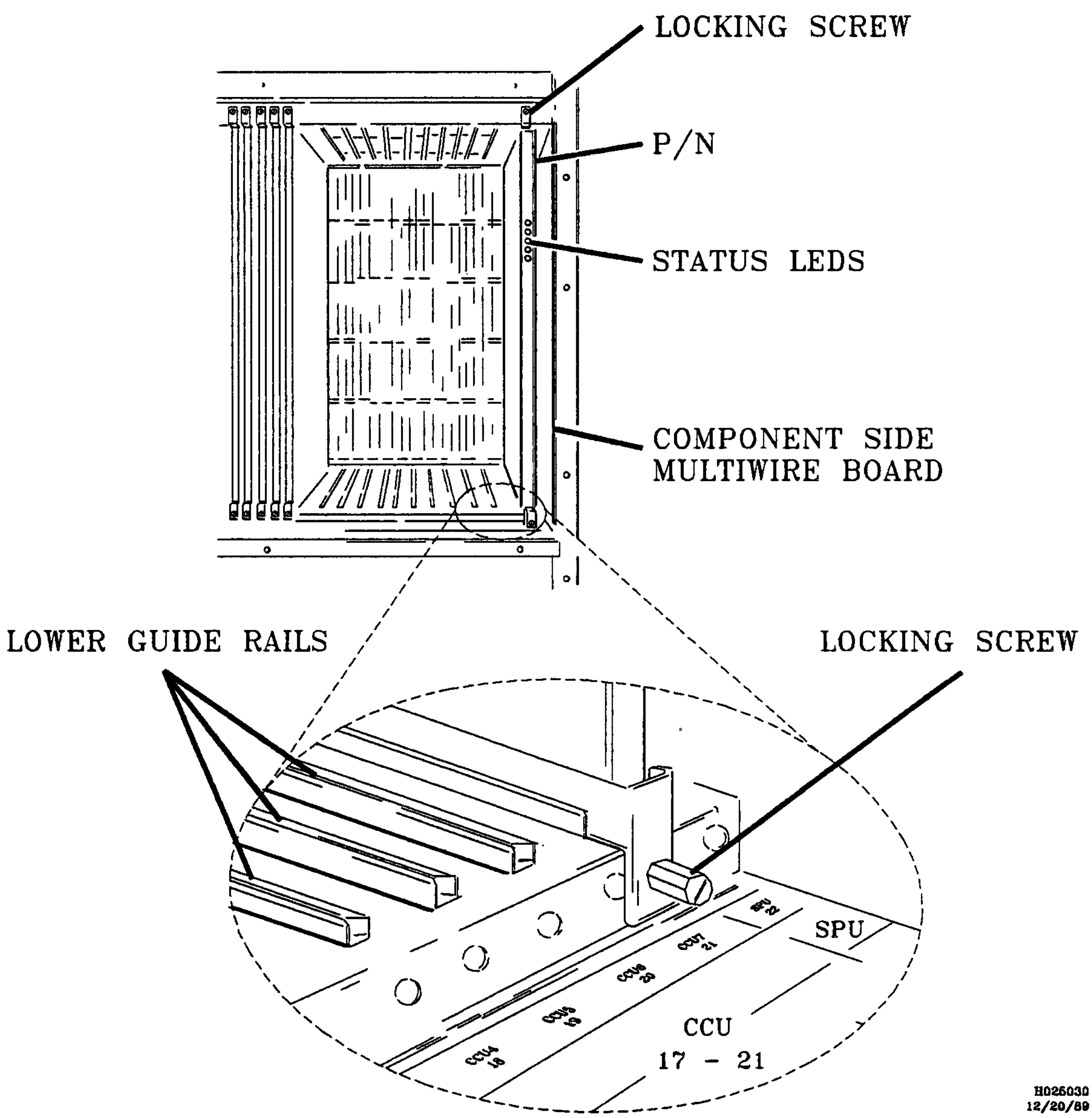
4. Remove the IDC board assembly from the **CCU** slot in the processor cabinet's logic rack.

4.4.1.2 Replacement

> **CAUTION**
>
> The IDC can be damaged by Electrostatic Discharge (ESD). A grounded wrist strap (or other grounding method) must be used when handling the IDC.

1. Ensure that steps 1, 2, and 3 in Section 4.3, "Maintenance Safety Procedures," are completed.
2. Install the IDC in the **CCU** slot in the processor cabinet's logic rack as shown in Figure 4–6, "CCU Slots and Mounting Hardware":

Figure 4–6, CCU Slots and Mounting Hardware



CAUTION

Failure to tighten the two chassis locking screws simultaneously may damage a connector or result in a faulty connection.

3. Using 2 nut drivers, simultaneously tighten the 2 chassis locking screws on the ends of the IDC board assembly as shown in the previous figure.
4. Replace the cover plate on the front of the processor card cage.
5. Set the power controller's main circuit breaker to the **ON** position and select local or remote operation with the LOCAL/REMOTE switch.
6. Turn the processor's front control panel key switch to the **ON** position.
7. Return the peripheral cabinet stabilizer bars to their retracted positions.

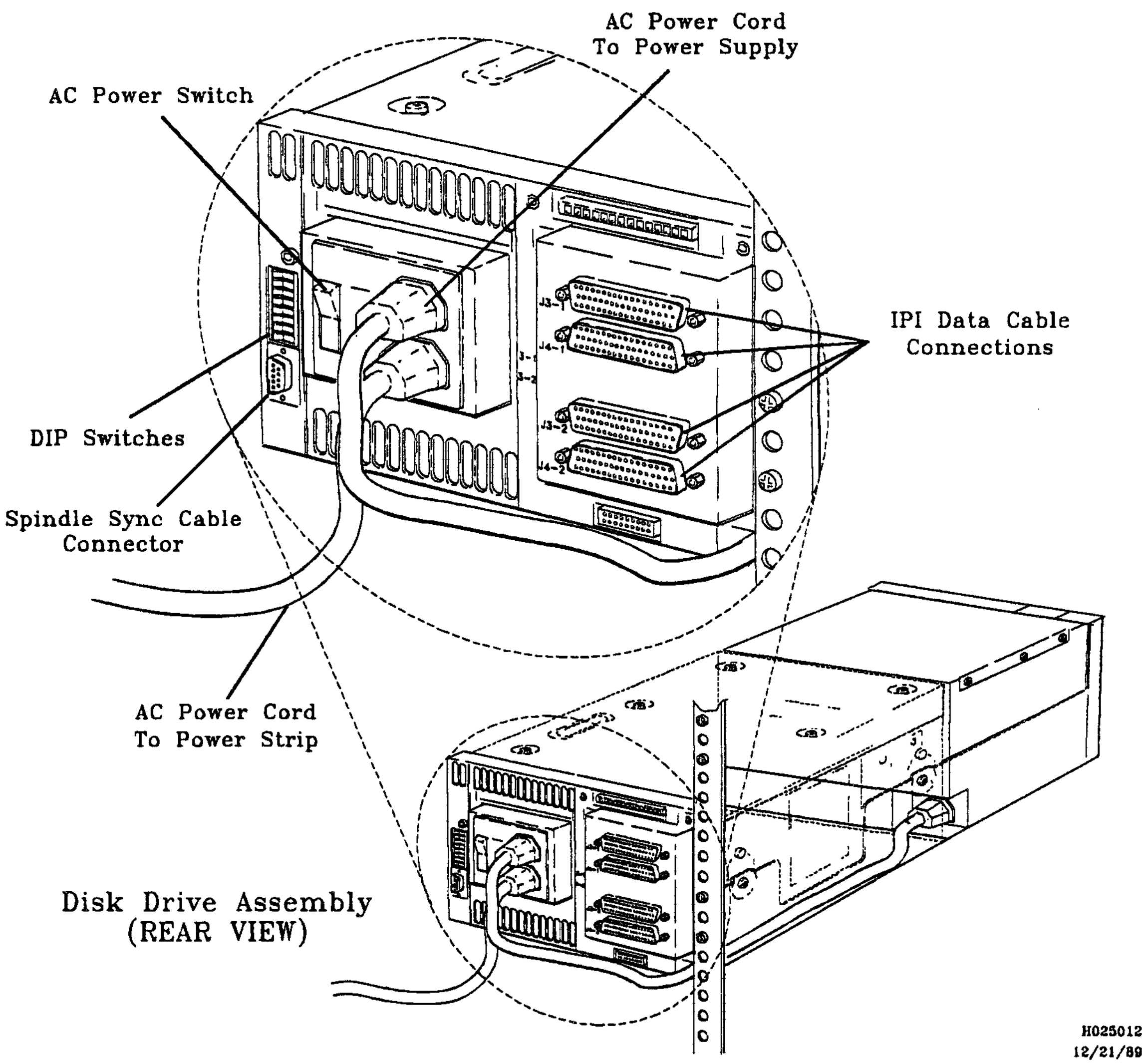
4.4.2 Disk Drive Module

Follow these procedures to remove and replace a disk drive module.

4.4.2.1 Removal

1. Complete steps 1, 2, and 3 in Section 4.3, "Maintenance Safety Procedures."
2. Remove the peripheral cabinet rear panel.
3. Turn the AC power switch on the rear of the right drive to the **OFF** position. Then disconnect the AC power cord from the AC power cord connector on the rear of the right drive as shown in Figure 4–7, "Drive Connections".
4. Turn the AC power switch on the rear of the left drive to the **OFF** position. Then disconnect the AC power cord from the AC power cord connector on the rear of the left drive.

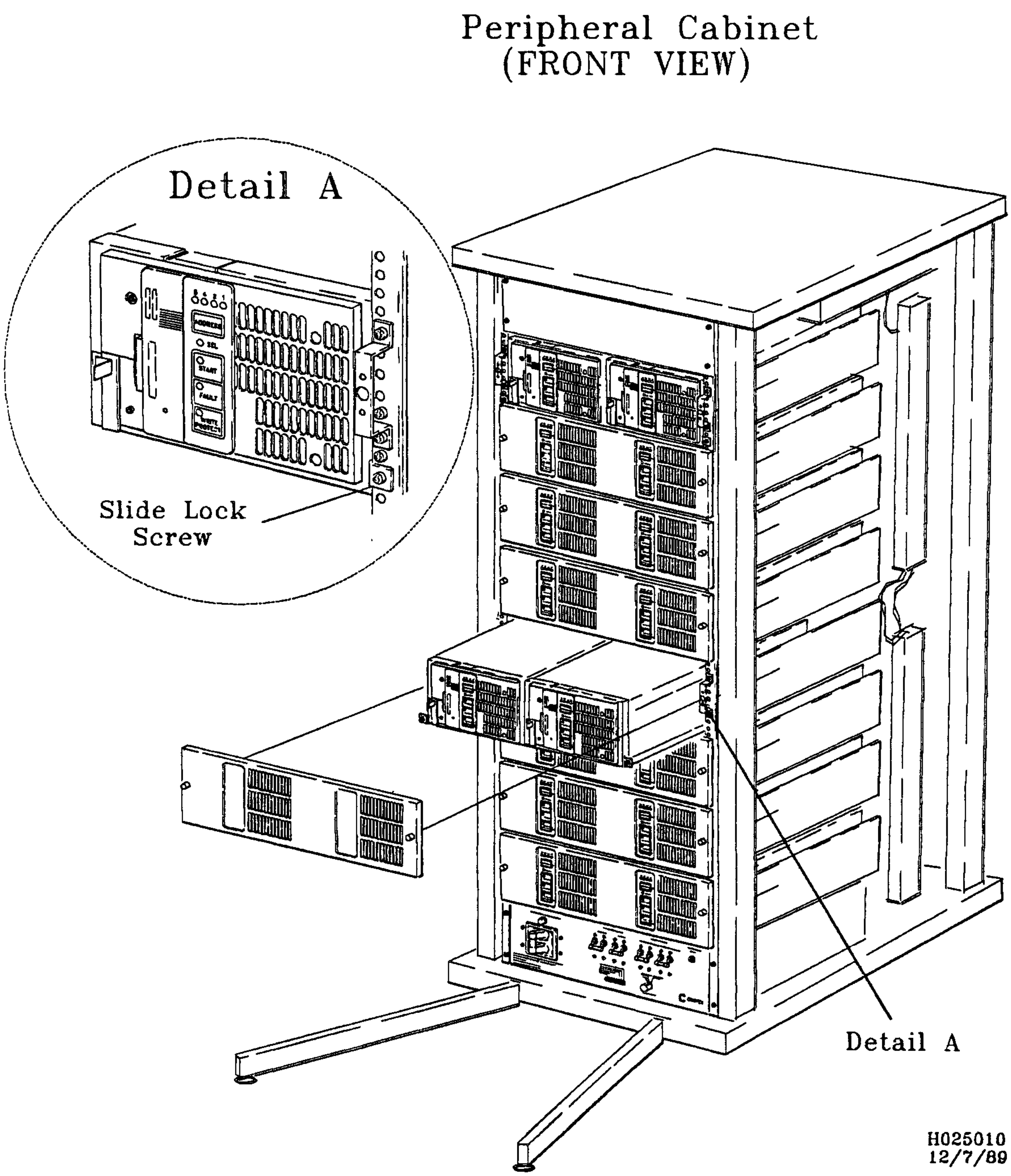
Figure 4–7, Drive connections



5. Disconnect the input data cable from the **J4-1** connector and the output data cable from the **J3-1** connector on the rear of the right drive. Also disconnect the spindle sync cable from the spindle sync connector on the rear of the right drive as shown in Figure 4–7, "Drive Connections".
6. Disconnect the input data cable from the **J4-1** connector and the output data cable from the **J3-1** connector on the rear of the left drive. Also disconnect the spindle sync cable from the spindle sync connector on the rear of the left drive.

7. Release the 2 front cover captive-lock screws and remove the front cover as shown in Figure 4–8, "Front Cover and Slide Lock":

Figure 4–8, Front Cover and Slide Lock



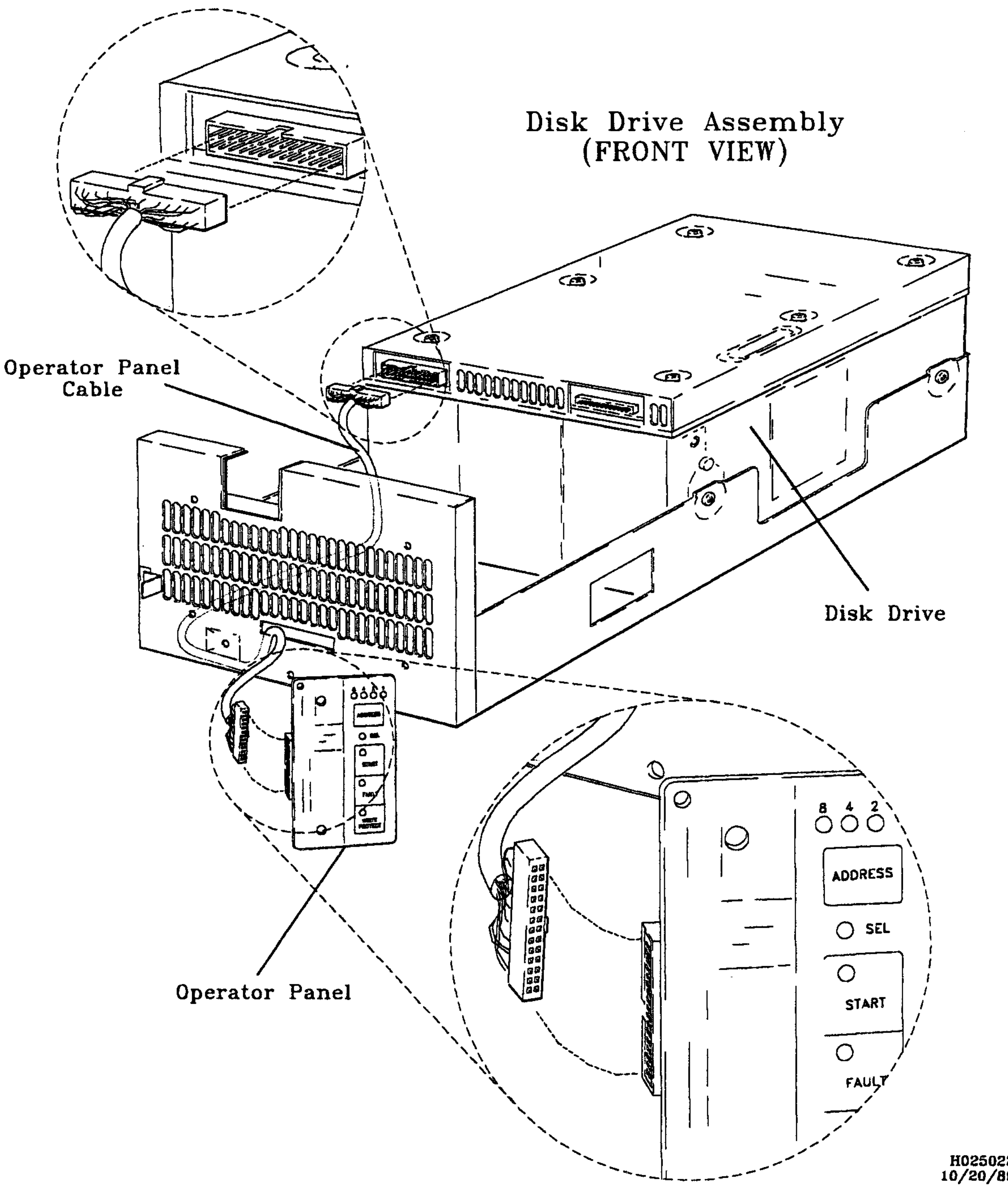
8. Release the 2 front chassis captive-lock screws and extend the assembly on its slides as shown in the previous figure.

> **NOTE**
>
> To remove the drive assembly locking screw located behind the bottom edge of the operator panel, the operator panel must be removed.

> **CAUTION**
>
> The operator panel can be damaged by Electrostatic Discharge (ESD). A grounded wrist strap (or other grounding method) must be used when handling the operator panel.

9. Remove the 2 screws in the operator panel and disconnect the operator panel cable from the operator panel as shown in Figure 4–9, "Operator Panel Cable":

Figure 4–9, Operator Panel Cable



10. Remove the drive assembly locking screw located on the front of the tray.

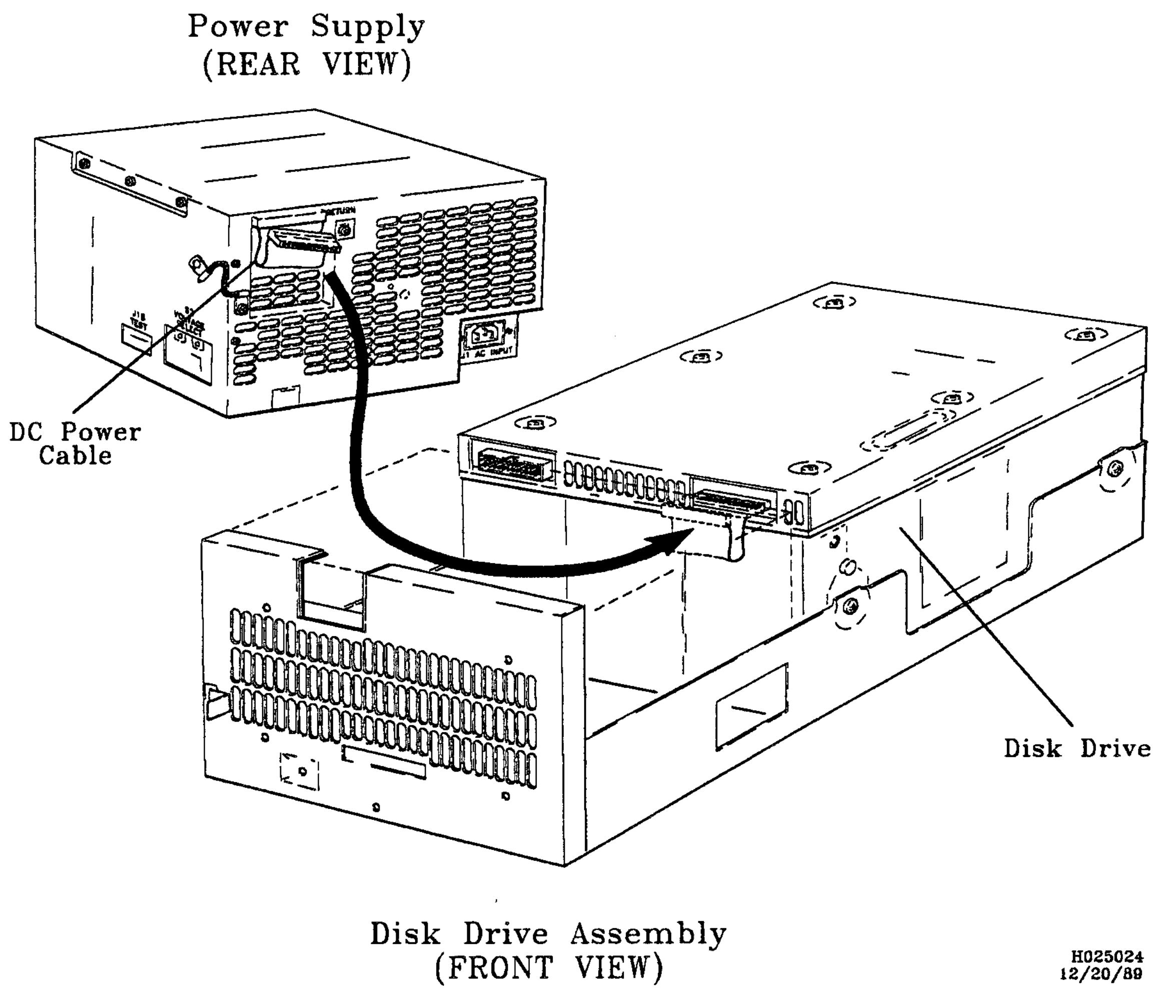
> **WARNING**
>
> Because a drive assembly weighs 60 lbs, personnel injury or equipment damage can occur when a drive assembly is installed or removed. Two people are required to install or remove a drive assembly.

11. Press the drive assembly tray lock to the right, then lift the drive assembly tray free of the slide tray.
12. Disconnect the AC power cord from the assembly AC jack at the rear of the disk drive module.
13. Remove the 2 locking screws on the connection cover plate and remove the plate.
14. Loosen the four disk drive module mounting screws and remove the ground strap locking screw located above the disk drive module mounting screw.

> **CAUTION**
>
> The DC power and operator panel cables must be disconnected from the disk module while removing the disk drive module from the drive assembly tray. If the disk drive module is moved too far away from the power supply without disconnecting the cables, the cables or cable connections may be damaged.

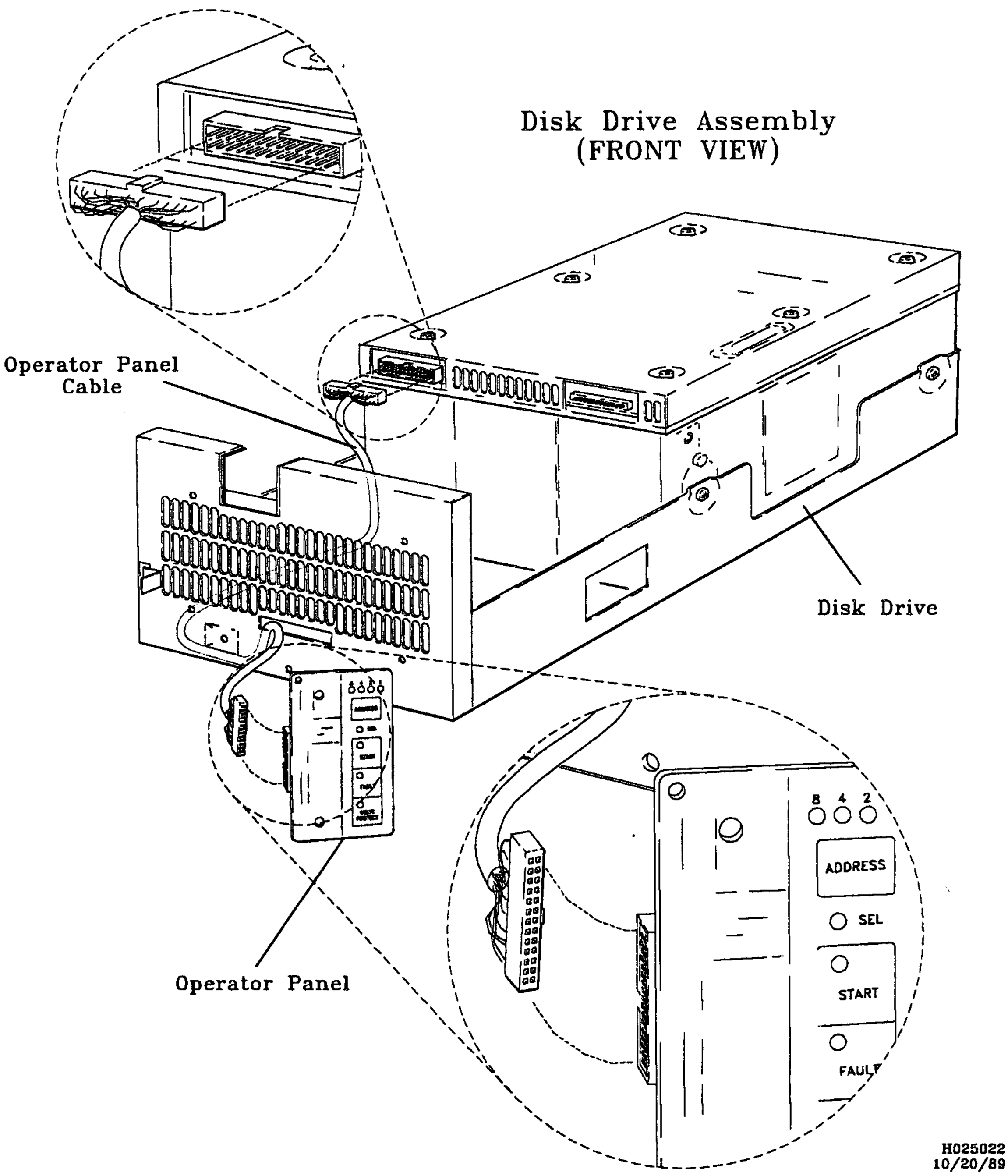
15. Lift the disk drive module until the disk drive module mounting screws are above the sides of the drive assembly tray. Then move the disk drive module to the rear of the disk assembly tray to allow access to the DC power and the operator panel cable connectors. Lower the disk drive module in the drive assembly tray until the disk drive module mounting screws are supported by the sides of the drive assembly tray.
16. Disconnect the DC power cable from the disk drive module as shown in Figure 4–10, "DC Power Cable":

Figure 4–10, DC Power Cable



17. Disconnect the operator panel cable from the disk drive module as shown in Figure 4–11, "Operator Panel Cable":

Figure 4–11, Operator Panel Cable



18. Remove the disk drive module from the drive assembly tray.

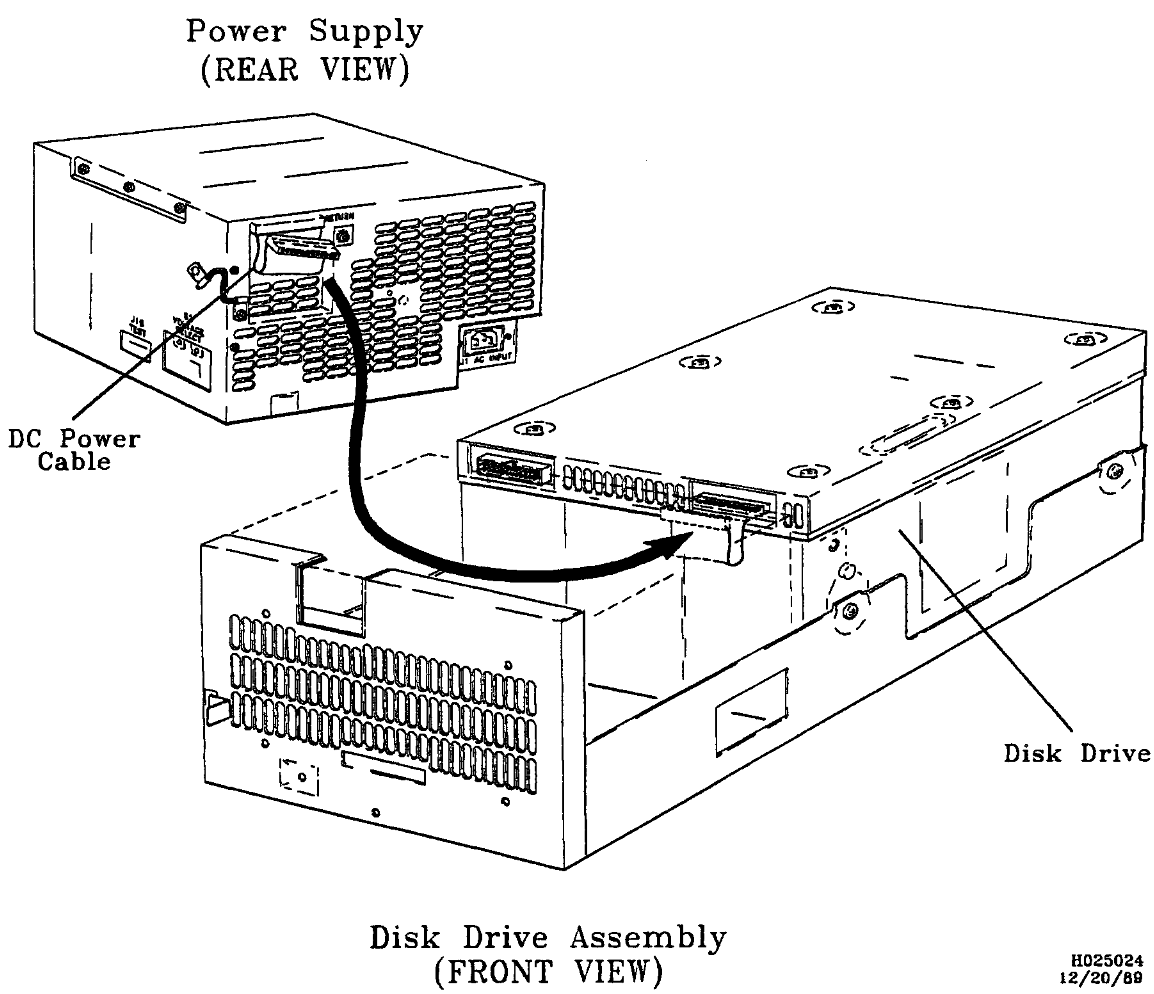
4.4.2.2 Replacement

1. Ensure steps 1, 2, and 3 in Section 4.3, "Maintenance Safety Procedures" are completed.

> **NOTE**
>
> When lowering the disk drive module into the drive assembly tray, place the disk drive module to the rear of the disk assembly tray to allow access to the DC power and the operator panel cable connectors.

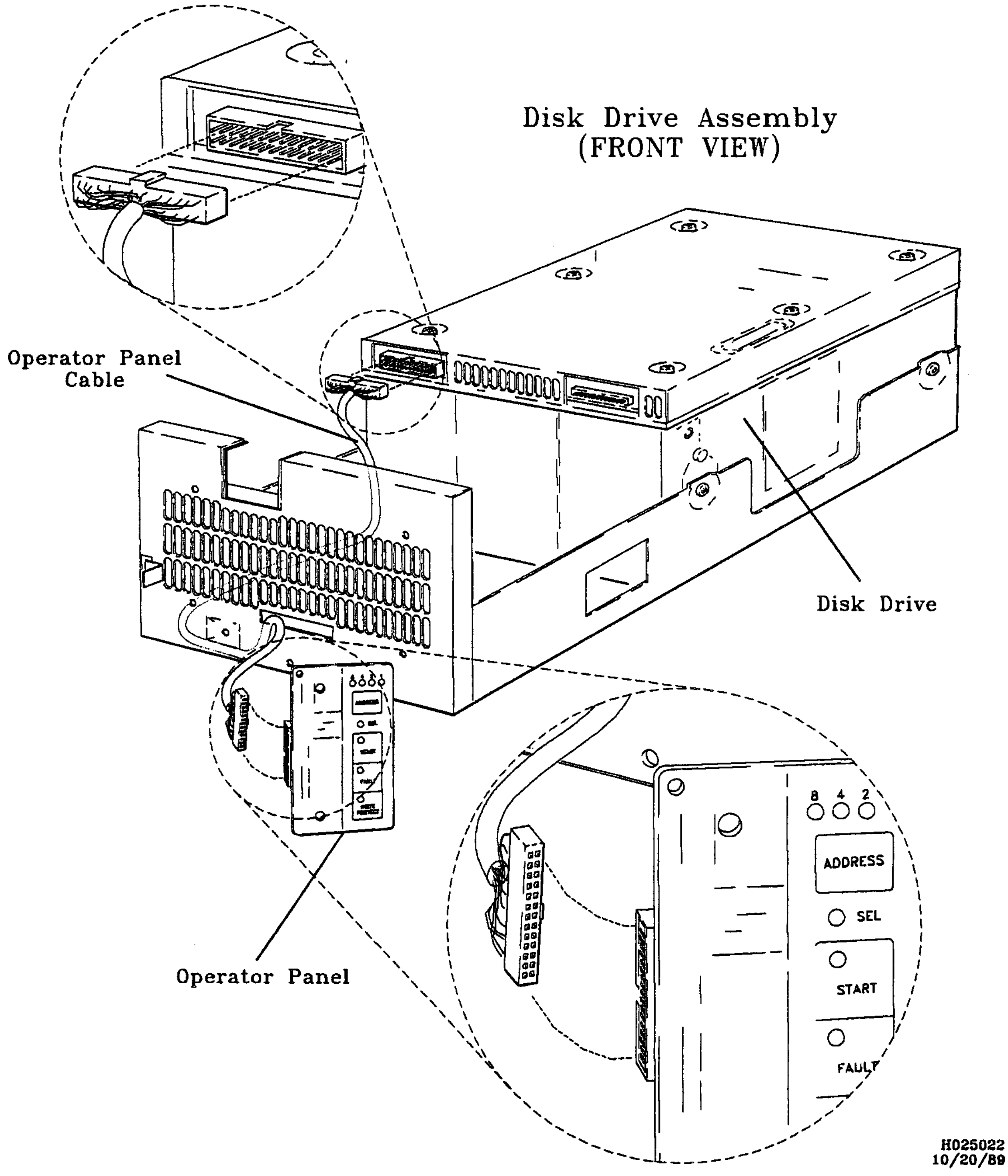
2. Lower the disk drive module into the drive assembly tray until the disk drive module mounting screws are supported by the sides of the drive assembly tray.
3. Connect the DC power cable to the disk drive module as shown in Figure 4–12, "DC Power Cable":

Figure 4–12, DC Power Cable



4. Connect the operator panel cable to the disk drive module as shown in Figure 4–13, "Operator Panel Cable":

Figure 4–13, Operator Panel Cable



5. Move the disk drive module forward in the drive assembly tray until the disk module mounting screws are aligned with the slots in the sides of the drive assembly tray.
6. Lower the the disk drive module and secure the disk drive module with the four disk drive module mounting screws.
7. Secure the ground strap located above the disk drive module mounting screw with the locking screw.
8. Install the connection cover plate and secure with 2 locking screws.
9. Connect the AC power cord to the assembly AC jack at the rear of the disk drive module.

WARNING

Because a drive assembly weighs 60 lbs, personnel injury or equipment damage can occur when a drive assembly is removed or installed. Two people are required to remove or install a drive assembly.

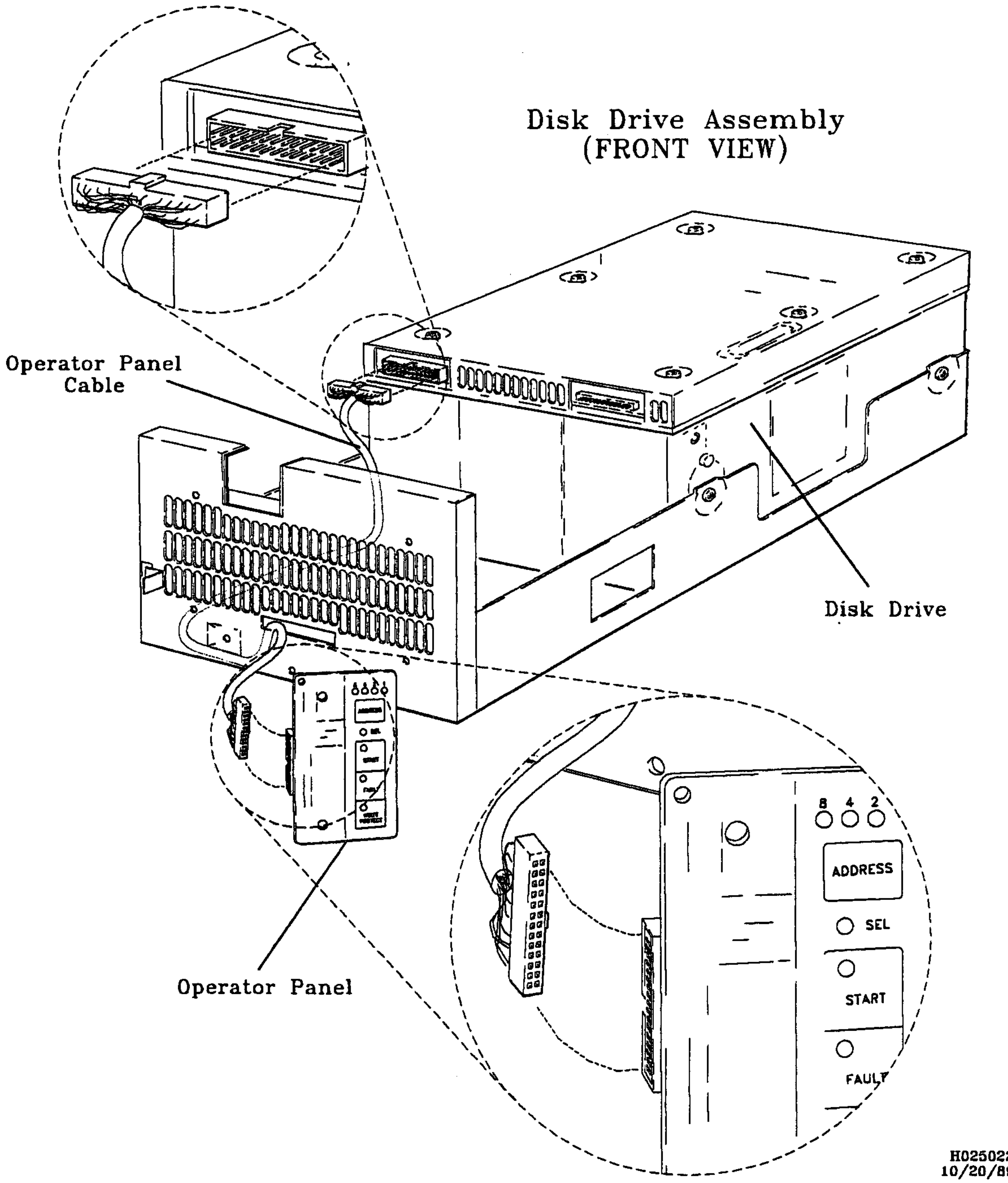
10. Press the drive assembly tray lock to the right, then lower the drive assembly tray into the slide tray.
11. Install the drive assembly locking screw located on the front of the tray.

CAUTION

The operator panel can be damaged by Electrostatic Discharge (ESD). A grounded wrist strap (or other grounding method) must be used when handling the operator panel.

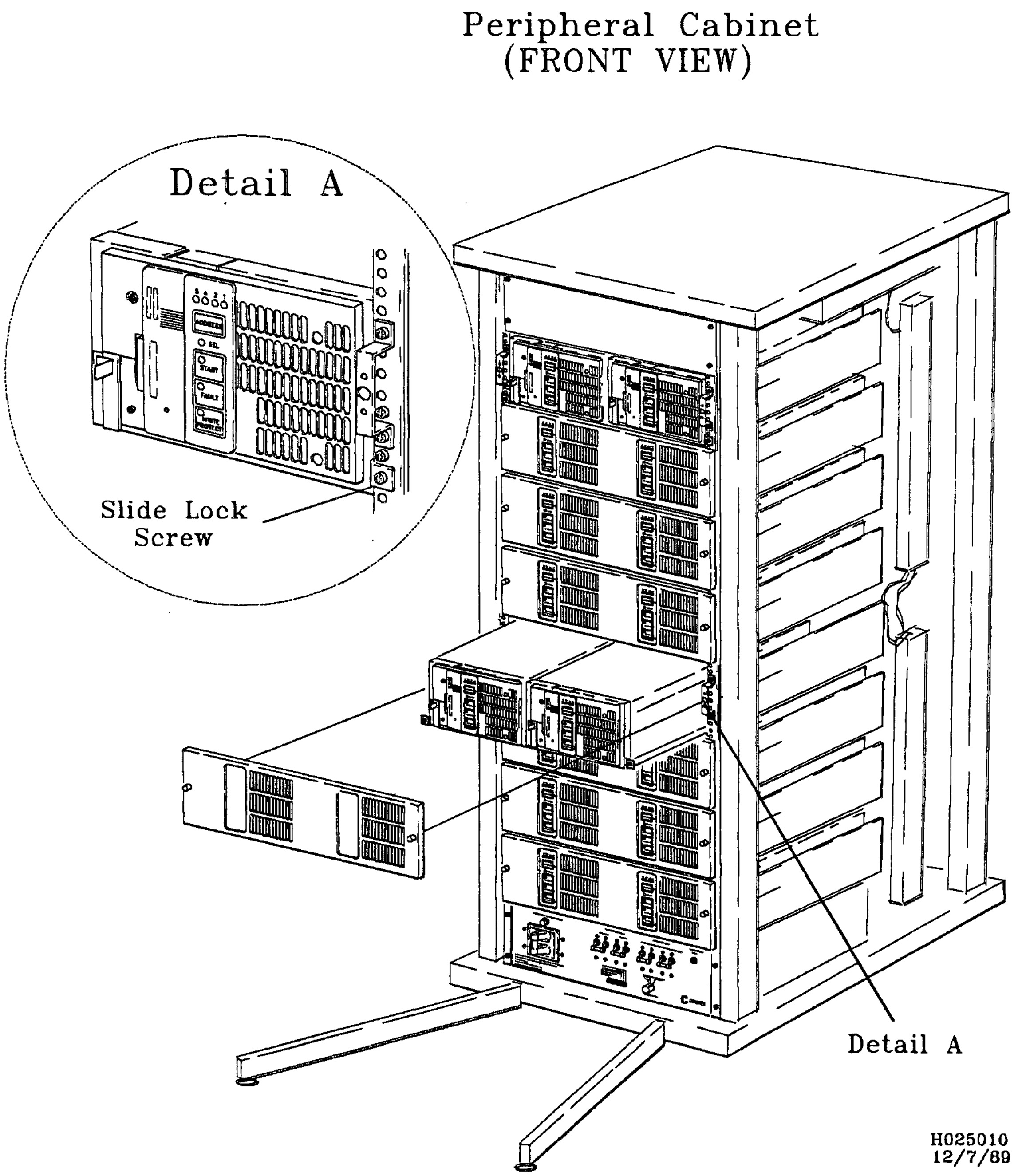
12. Connect the operator panel cable to the operator panel as shown in Figure 4–14, "Operator Panel Cable":

Figure 4–14, Operator Panel Cable



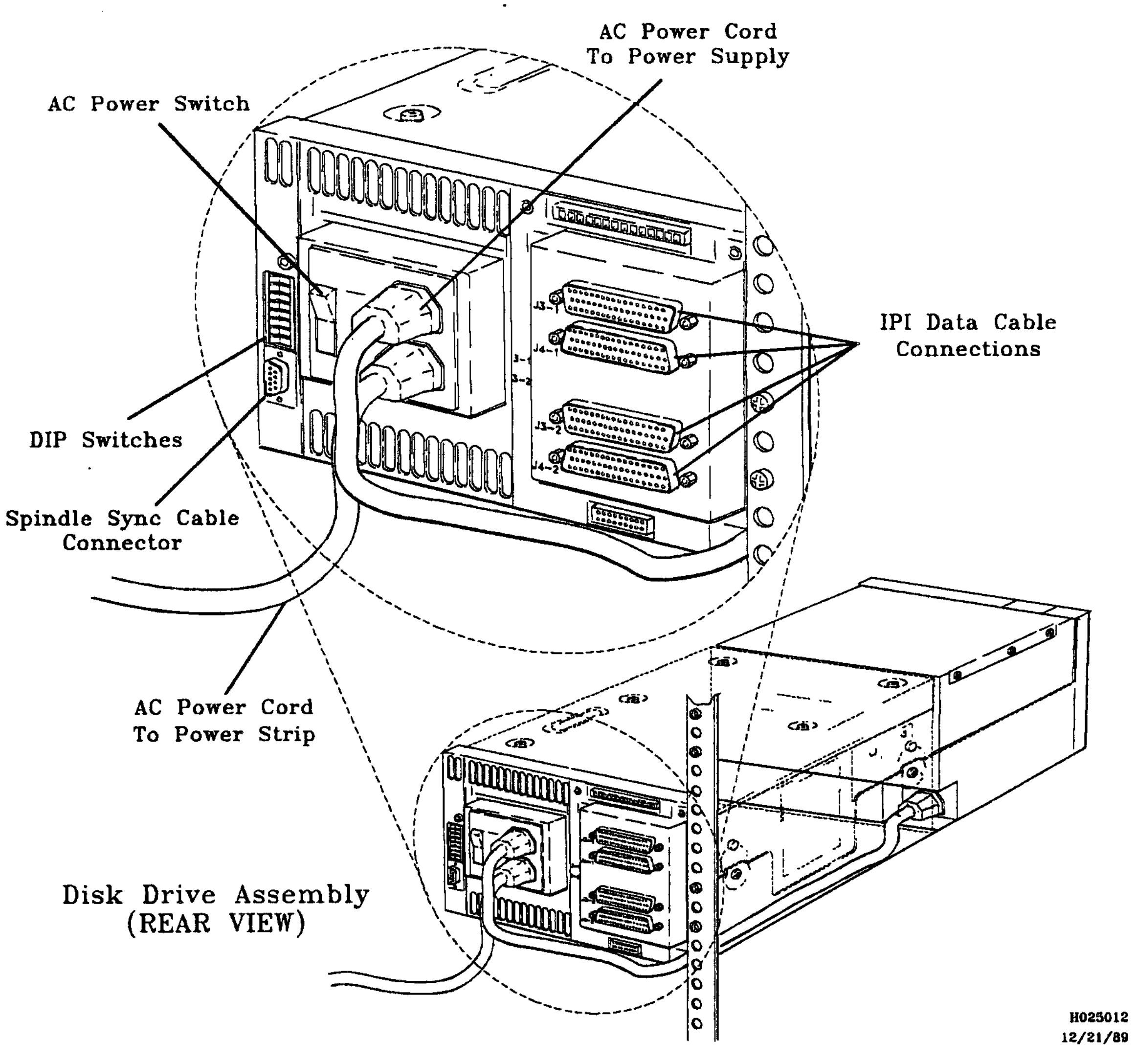
13. Install the 2 screws in the operator panel.
14. Retract the assembly on its slides and secure with the 2 front chassis captive-lock screws.
15. Install the front cover and secure with the 2 front cover captive-lock screws as shown in Figure 4–15, "Front Cover and Slide Lock":

Figure 4–15, Front Cover and Slide Lock



16. Connect the input data cable to the **J4-1** connector and the output data cable to the **J3-1** connector on the rear of the right drive. Also connect the spindle sync cable to the spindle sync connector on the rear of the right drive as shown in Figure 4-16, "Drive Connections".
17. Connect the input data cable to the **J4-1** connector and the output data cable to the **J3-1** connector on the rear of the left drive. Also connect the spindle sync cable to the spindle sync connector on the rear of the left drive.

Figure 4-16, Drive Connections



18. Connect the AC power cord to the AC power cord connector on the rear of the right drive. Then turn the AC power switch on the rear of the right drive to the **ON** position as shown in Figure 4–16, "Drive Connections".
19. Connect the AC power cord to the AC power cord connector on the rear of the left drive. Then turn the AC power switch on the rear of the left drive to the **ON** position.
20. Install the peripheral cabinet rear panel.
21. Set the power controller's main circuit breaker to the **ON** position and select local or remote operation with the LOCAL/REMOTE switch.
22. Turn the processor's front control panel key switch to the **ON** position.
23. Return the peripheral cabinet stabilizer bars to their retracted positions.

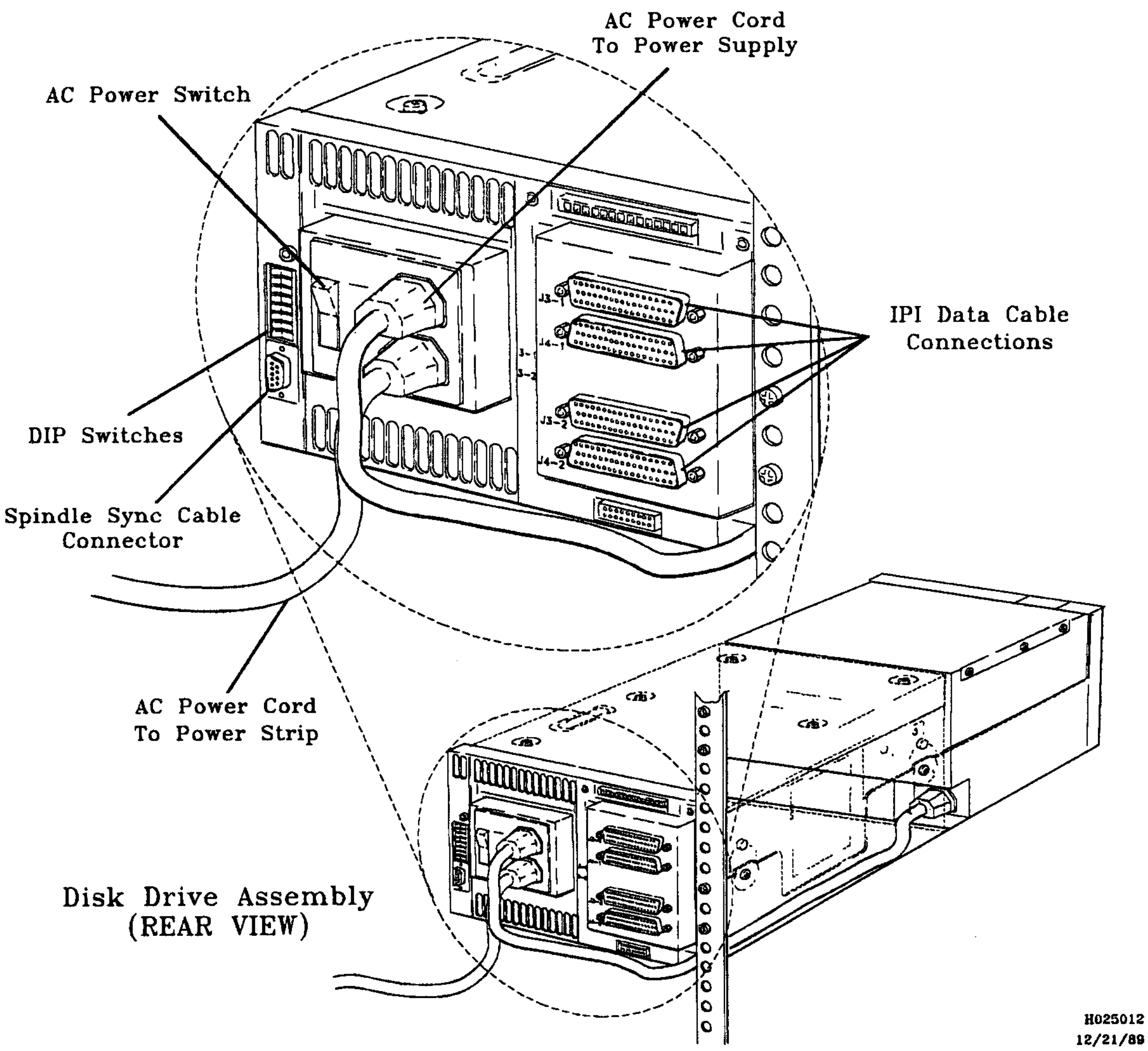
4.4.3 Power Supply

Follow these procedures to remove and replace a power supply.

4.4.3.1 Removal

1. Complete steps 1, 2, and 3 in Section 4.3, "Maintenance Safety Procedures."
2. Remove the peripheral cabinet rear panel.
3. Turn the AC power switch on the rear of the right drive to the **OFF** position. Then disconnect the AC power cord from the AC power cord connector on the rear of the right drive as shown in Figure 4–17, "Drive Connections".
4. Turn the AC power switch on the rear of the left drive to the **OFF** position. Then disconnect the AC power cord from the AC power cord connector on the rear of the left drive.

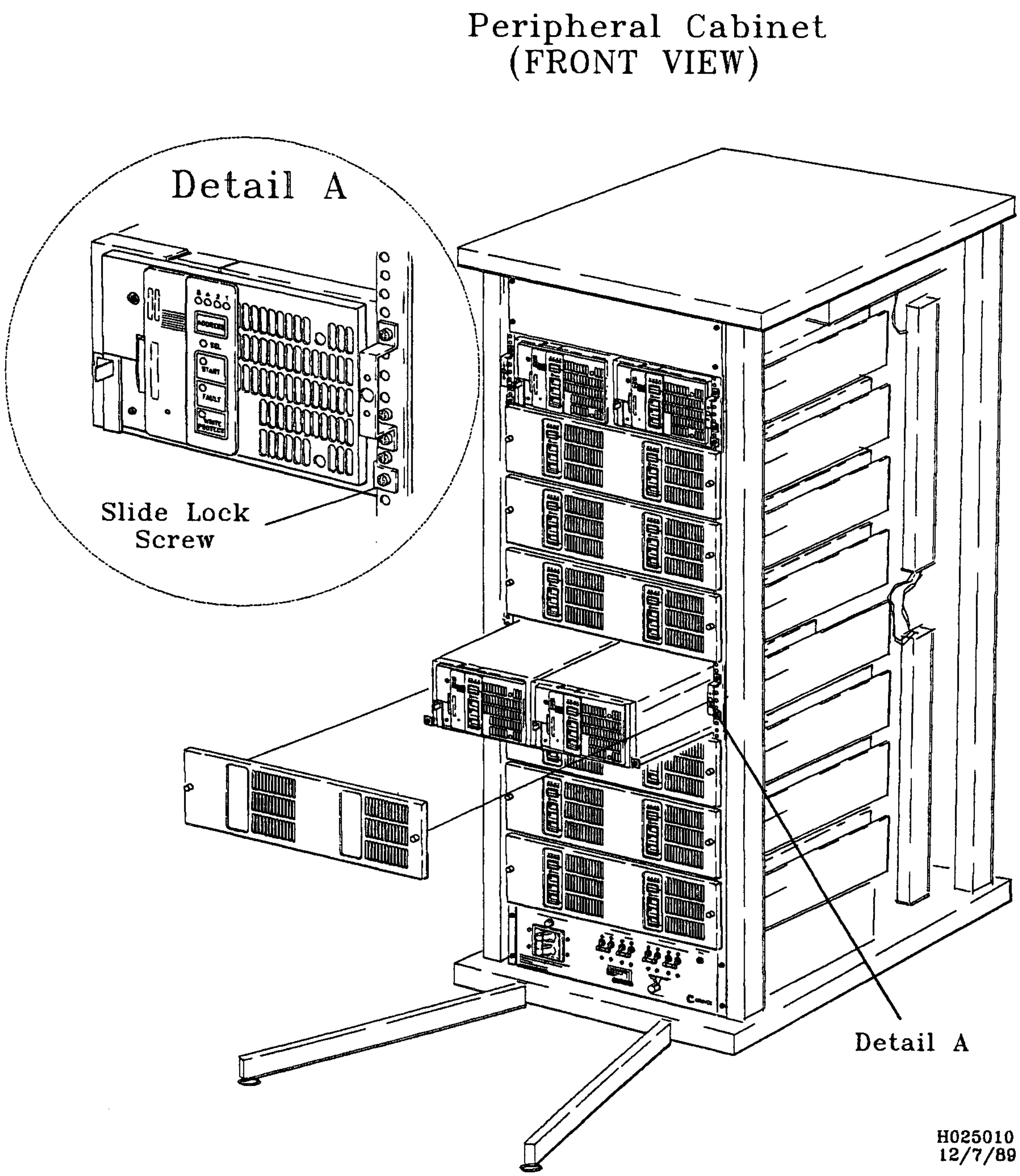
Figure 4–17, Drive Connections



5. Disconnect the input data cable from the **J4-1** connector and the output data cable from the **J3-1** connector on the rear of the right drive. Also disconnect the spindle sync cable from the spindle sync connector on the rear of the right drive as shown in Figure 4–17, "Drive Connections".
6. Disconnect the input data cable from the **J4-1** connector and the output data cable from the **J3-1** connector on the rear of the left drive. Also disconnect the spindle sync cable from the spindle sync connector on the rear of the left drive.

7. Release the 2 front cover captive-lock screws and remove the front cover as shown in Figure 4–18, "Front Cover and Slide Lock":

Figure 4–18, Front Cover and Slide Lock



8. Release the 2 front chassis captive-lock screws and extend the assembly on its slides as shown in the previous figure.

NOTE

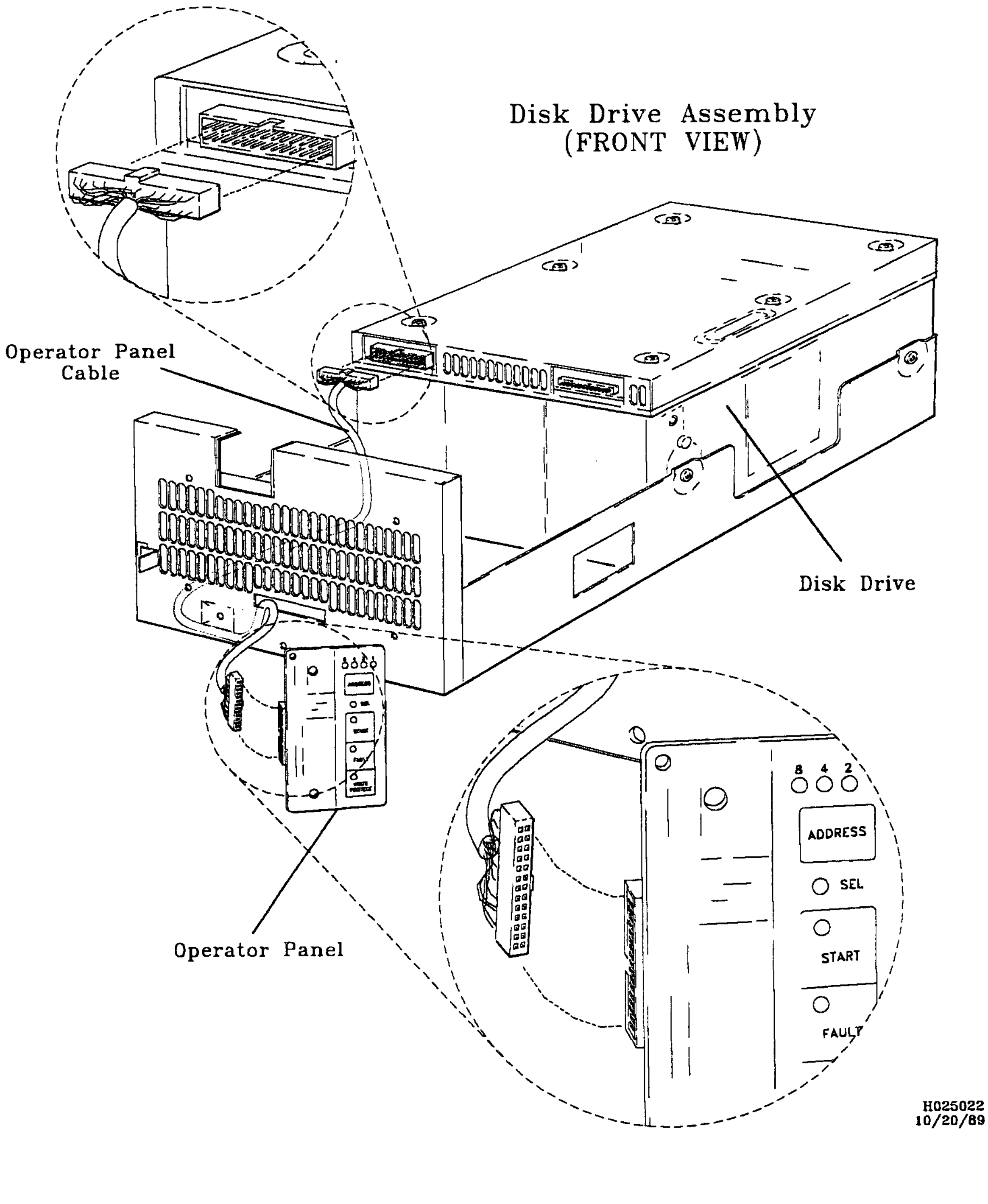
To remove the drive assembly locking screw located behind the bottom edge of the operator panel, the operator panel must be removed.

CAUTION

The operator panel can be damaged by Electrostatic Discharge (ESD). A grounded wrist strap (or other grounding method) must be used when handling the operator panel.

9. Remove the 2 screws in the operator panel and disconnect the operator panel cable from the operator panel as shown in Figure 4–19, "Operator Panel Cable":

Figure 4–19, Operator Panel Cable



10. Remove the drive assembly locking screw located on the front of the tray.

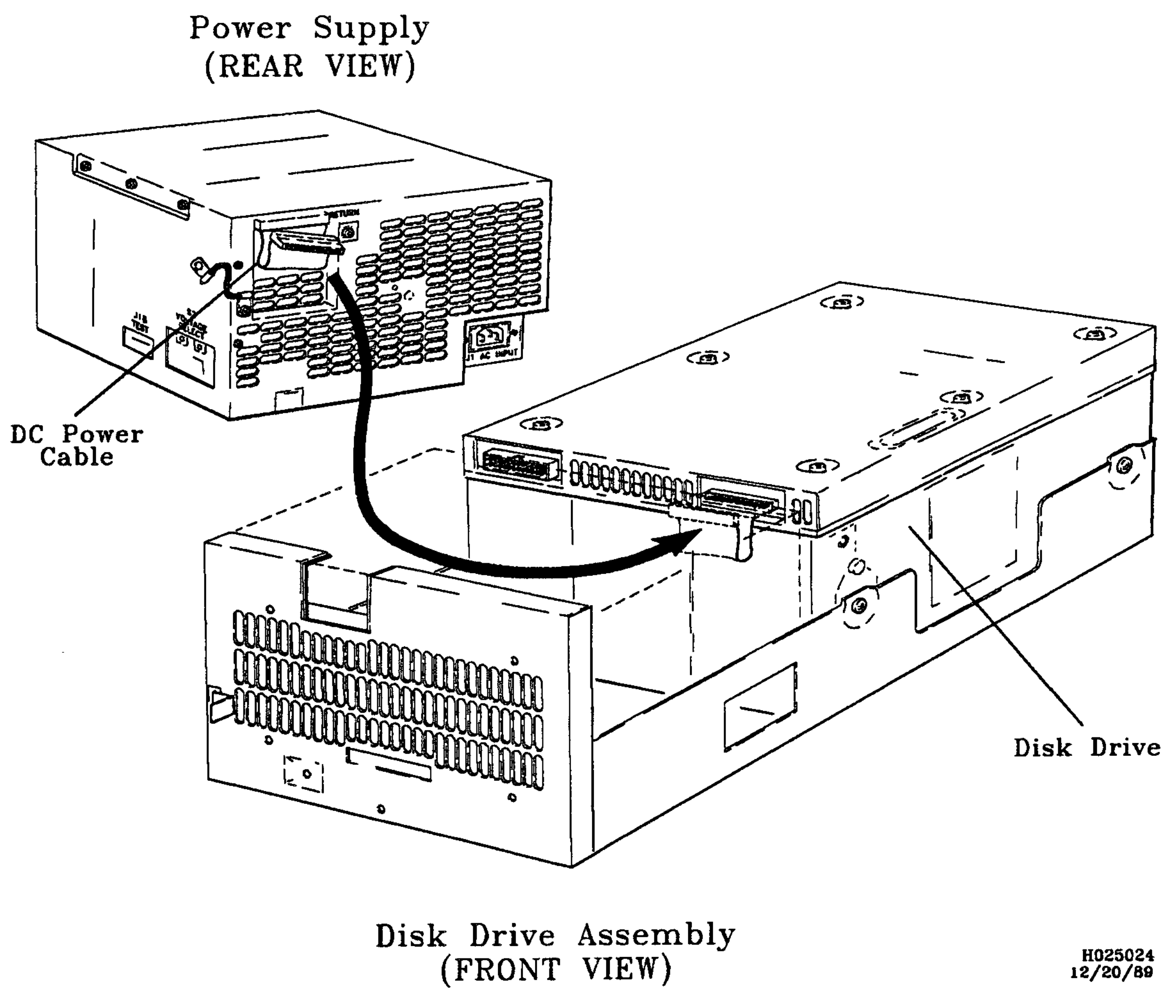
> **WARNING**
>
> Because a drive assembly weighs 60 lbs, personnel injury or equipment damage can occur when a drive assembly is installed or removed. Two people are required to install or remove a drive assembly.

11. Press the drive assembly tray lock to the right, then lift the drive assembly tray free of the slide tray.
12. Disconnect the AC power cord from the assembly AC jack at the rear of the disk drive module.
13. Remove the 2 locking screws on the connection cover plate and remove the plate.
14. Loosen the four disk drive module mounting screws and remove the ground strap locking screw located above the disk drive module mounting screw.

> **CAUTION**
>
> The DC power and operator panel cables must be disconnected from the disk module while removing the disk drive module from the drive assembly tray. If the disk drive module is moved too far away from the power supply without disconnecting the cables, the cables or cable connections may be damaged.

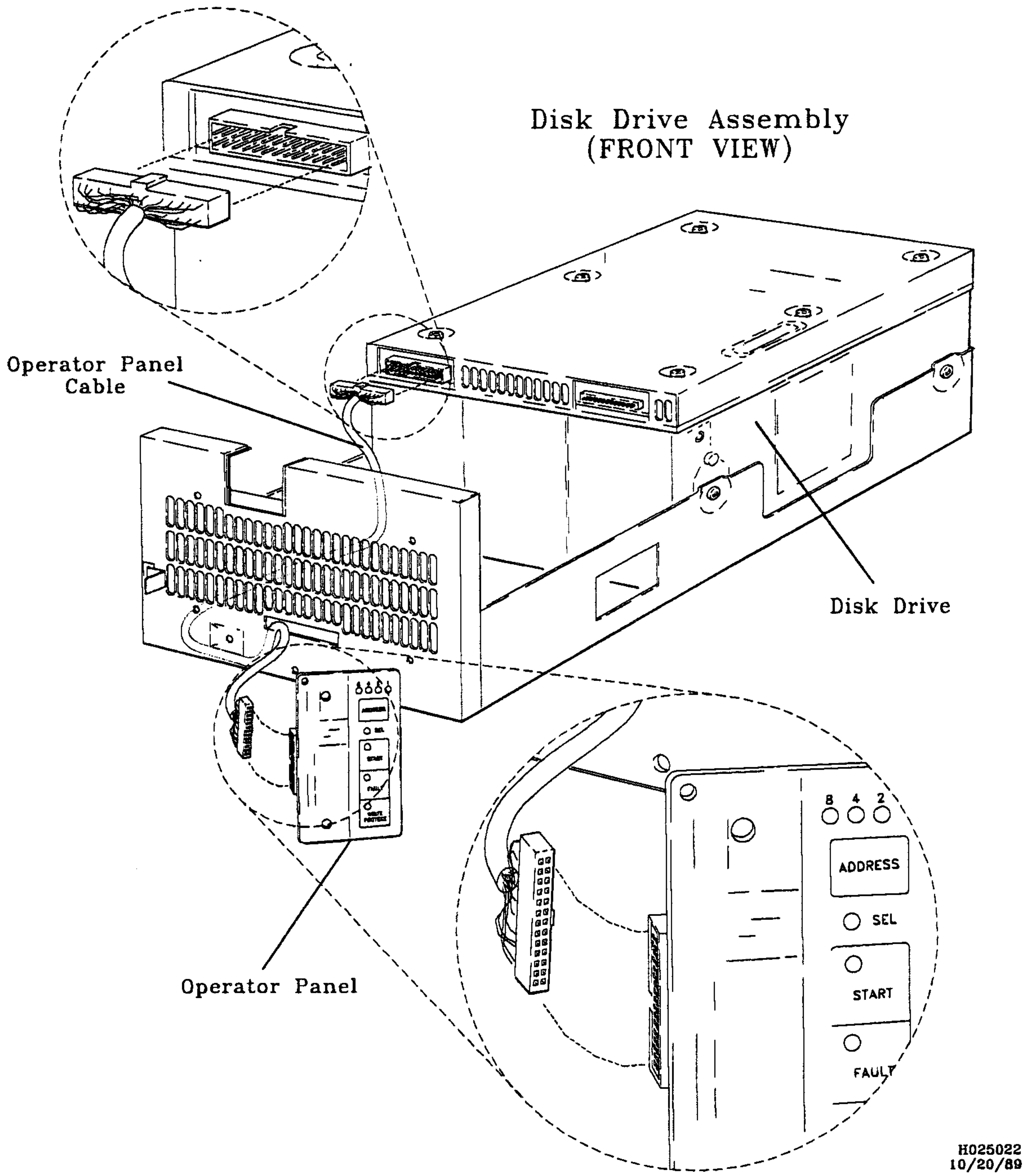
15. Lift the disk drive module until the disk drive module mounting screws are above the sides of the drive assembly tray. Then move the disk drive module to the rear of the disk assembly tray to allow access to the DC power and the operator panel cable connectors. Lower the disk drive module in the drive assembly tray until the disk drive module mounting screws are supported by the sides of the drive assembly tray.
16. Disconnect the DC power cable from the disk drive module as shown in Figure 4-20, "DC Power Cable":

Figure 4–20, DC Power Cable



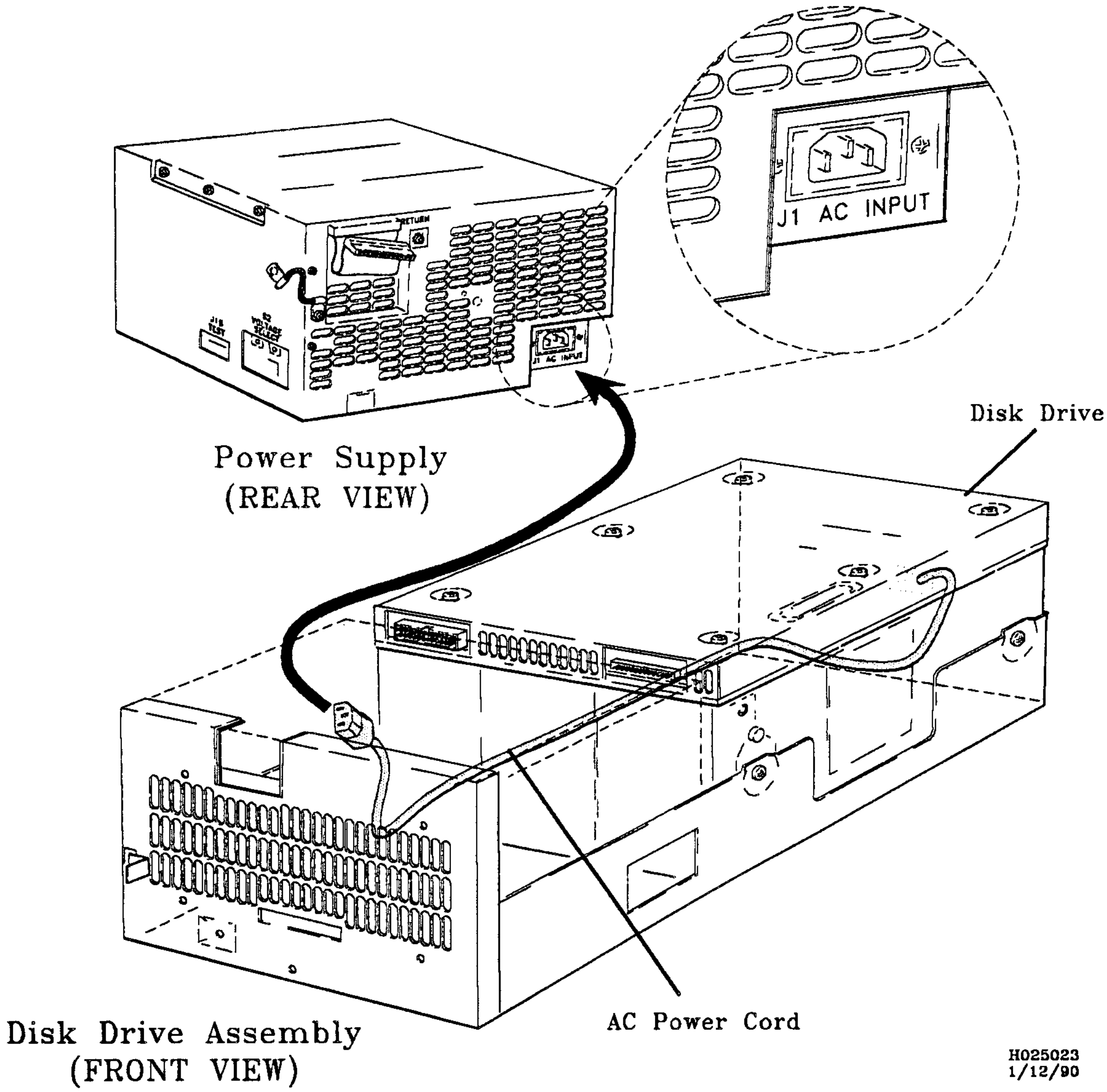
17. Disconnect the operator panel cable from the disk drive module as shown in Figure 4–21, “Operator Panel Cable”:

Figure 4–21, Operator Panel Cable



18. Remove the disk drive module from the drive assembly tray.
19. Disconnect the AC power cord from the power supply as shown in Figure 4–22, "Power Supply AC Power Cord":

Figure 4–22, Power Supply AC Power Cord



> **NOTE**
>
> The drive assembly tray should be turned on its side to allow access to the power supply mounting screws.

20. Turn the drive assembly tray on its side.
21. Remove the four power supply mounting screws and remove the power supply from the drive assembly tray.
22. Disconnect the DC power cable from the power supply.
23. If the replacement power supply does not have a ground strap, remove the ground strap located below the DC power cable connection on the power supply.

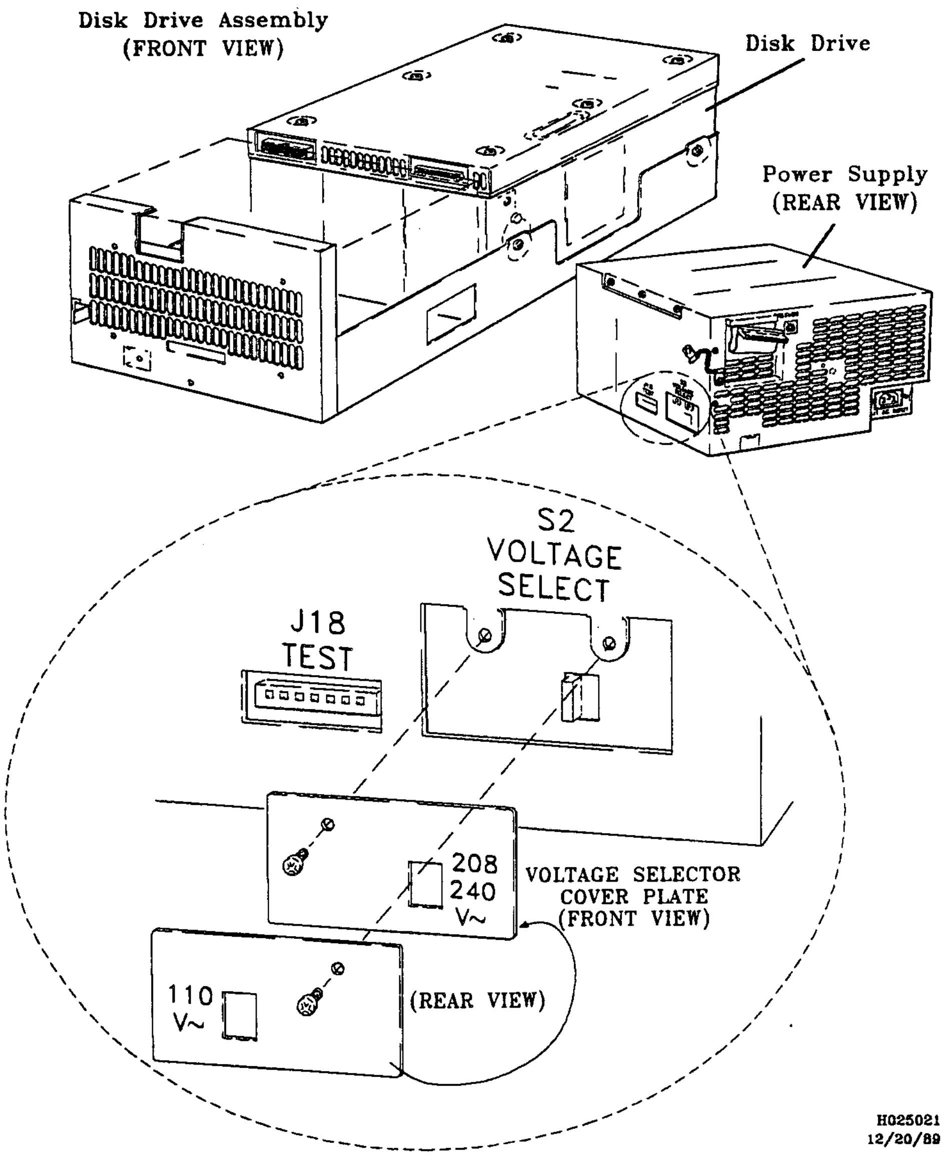
4.4.3.2 Replacement

1. Ensure that steps 1, 2, and 3 in Section 4.3, "Maintenance Safety Procedures," are completed.

> **CAUTION**
>
> Applying an incorrect voltage level to a power supply can damage electronic components.

2. Verify the drive power supply voltage setting is correct. If the voltage setting is incorrect, set the voltage as shown in Figure 4–23, "Power Supply Voltage Setting":

Figure 4–23, Power Supply Voltage Setting



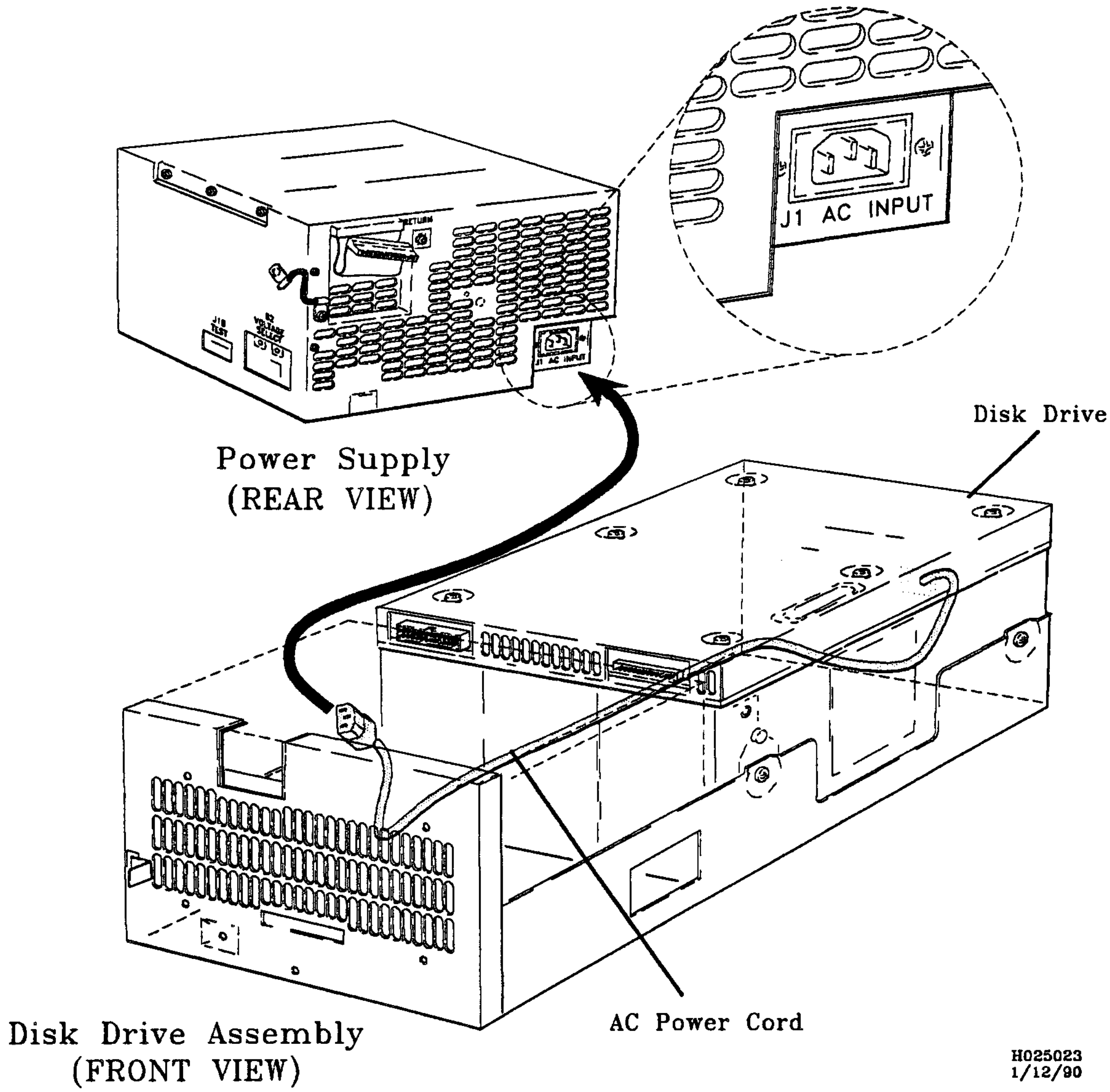
3. Set the ON/STANDBY switch on the front of the power supply to the **ON** position.
4. If the replacement power supply does not have a ground strap, install a ground strap below the DC power cable connection on the power supply.
5. Connect the DC power cable to the power supply.

NOTE

The drive assembly tray should be turned on its side to allow access to the power supply mounting screws.

6. Place the power supply in the drive assembly tray.
7. Move the power supply forward in the disk assembly tray until the mounting holes in the power supply are aligned with the holes in the disk assembly tray.
8. Secure the power supply with the four power supply mounting screws.
9. Return the drive assembly tray to an upright position.
10. Connect the AC power cord to the power supply as shown in Figure 4–24, "Power Supply AC Power Cord":

Figure 4–24, Power Supply AC Power Cord

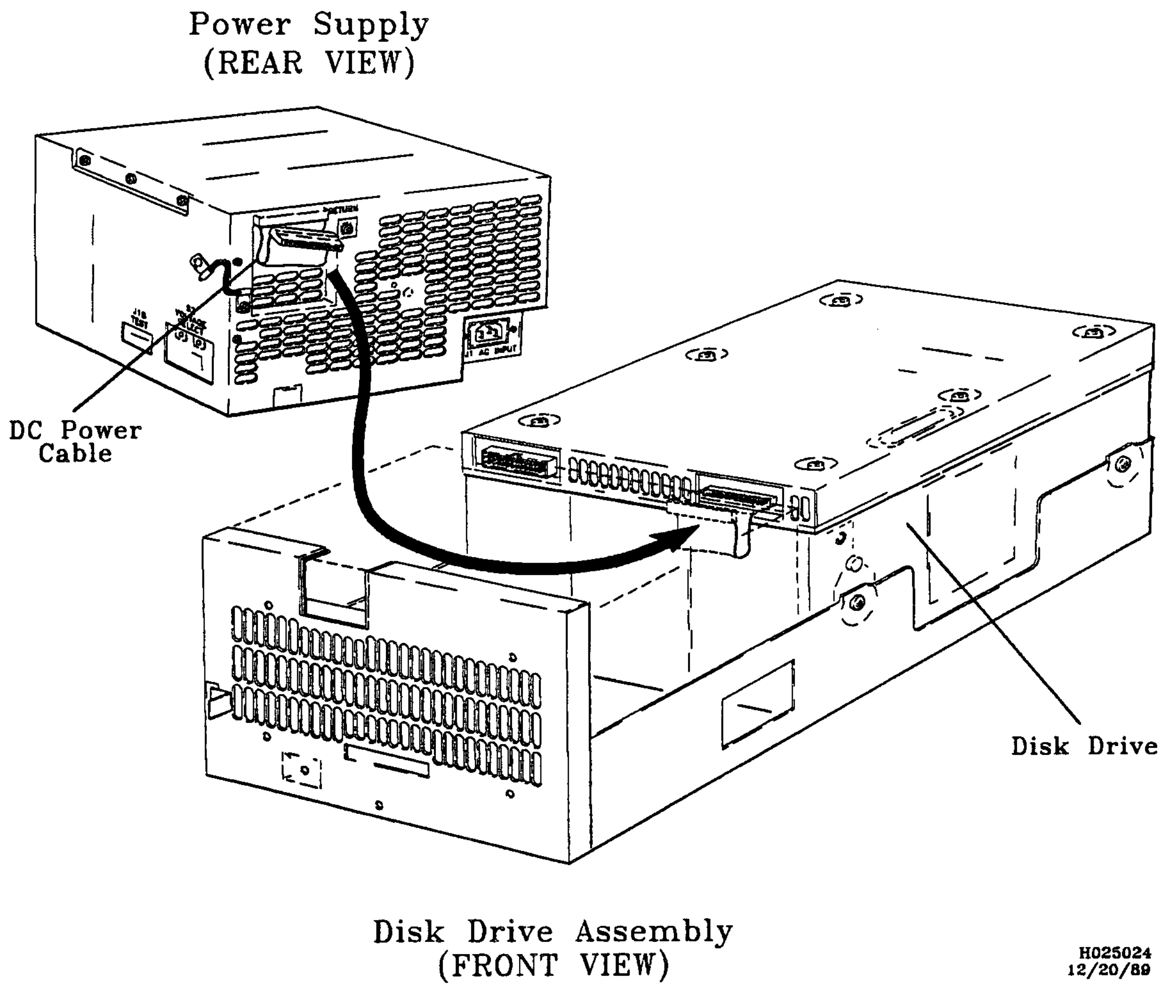


NOTE

When lowering the disk drive module into the drive assembly tray, place the disk drive module to the rear of the disk assembly tray to allow access to the DC power and the operator panel cable connectors.

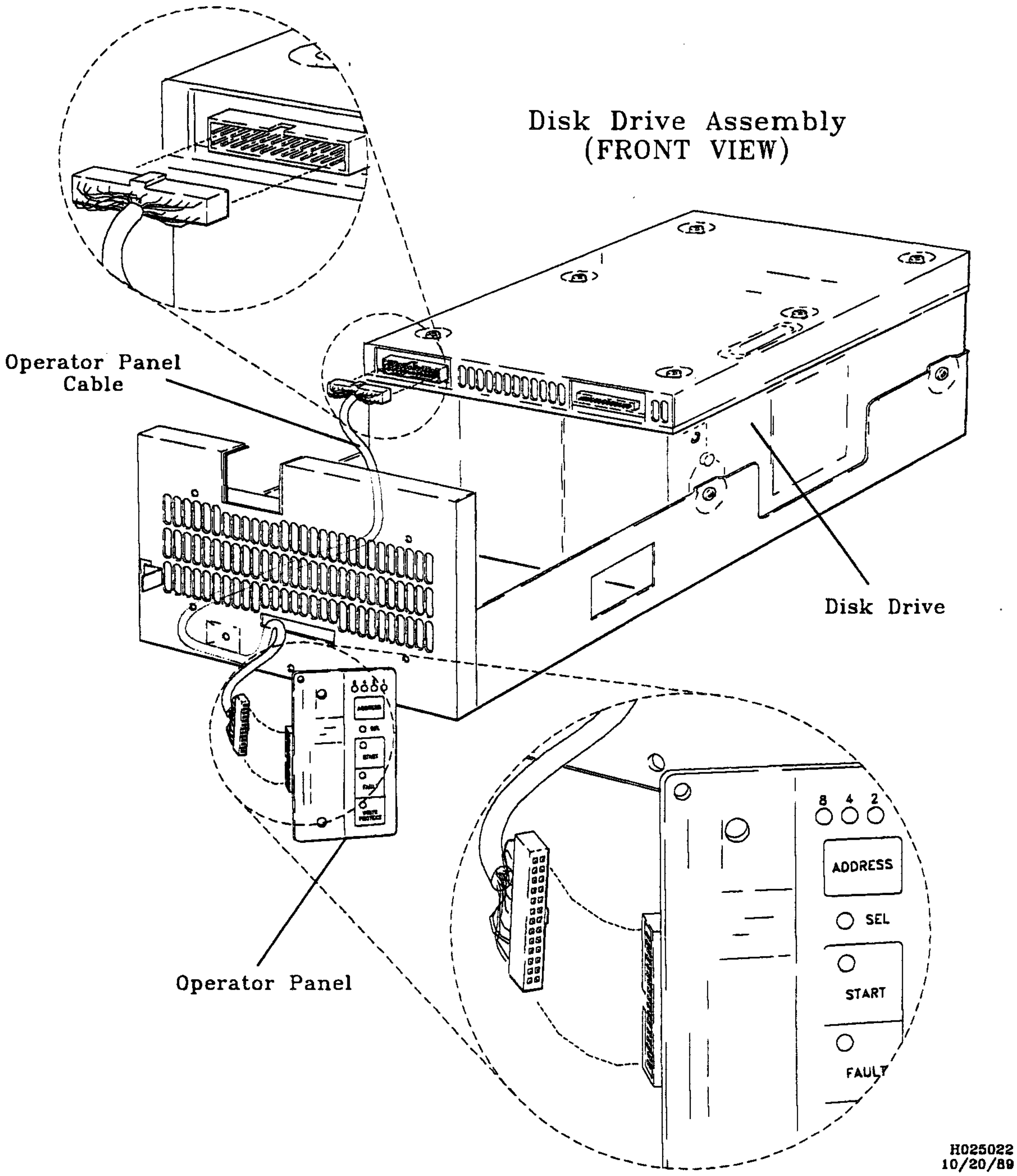
11. Lower the disk drive module into the drive assembly tray until the disk drive module mounting screws are supported by the sides of the drive assembly tray.
12. Connect the DC power cable to the disk drive module as shown in Figure 4–25, "DC Power Cable":

Figure 4–25, DC Power Cable



13. Connect the operator panel cable to the disk drive module as shown in Figure 4–26, "Operator Panel Cable":

Figure 4–26, Operator Panel Cable



14. Move the disk drive module forward in the drive assembly tray until the disk module mounting screws are aligned with the slots in the sides of the drive assembly tray.
15. Lower the disk drive module and secure the disk drive module with the four disk drive module mounting screws.
16. Secure the ground strap located above the disk drive module mounting screw with the locking screw.
17. Install the connection cover plate and secure with 2 locking screws.
18. Connect the AC power cord to the assembly AC jack at the rear of the disk drive module.

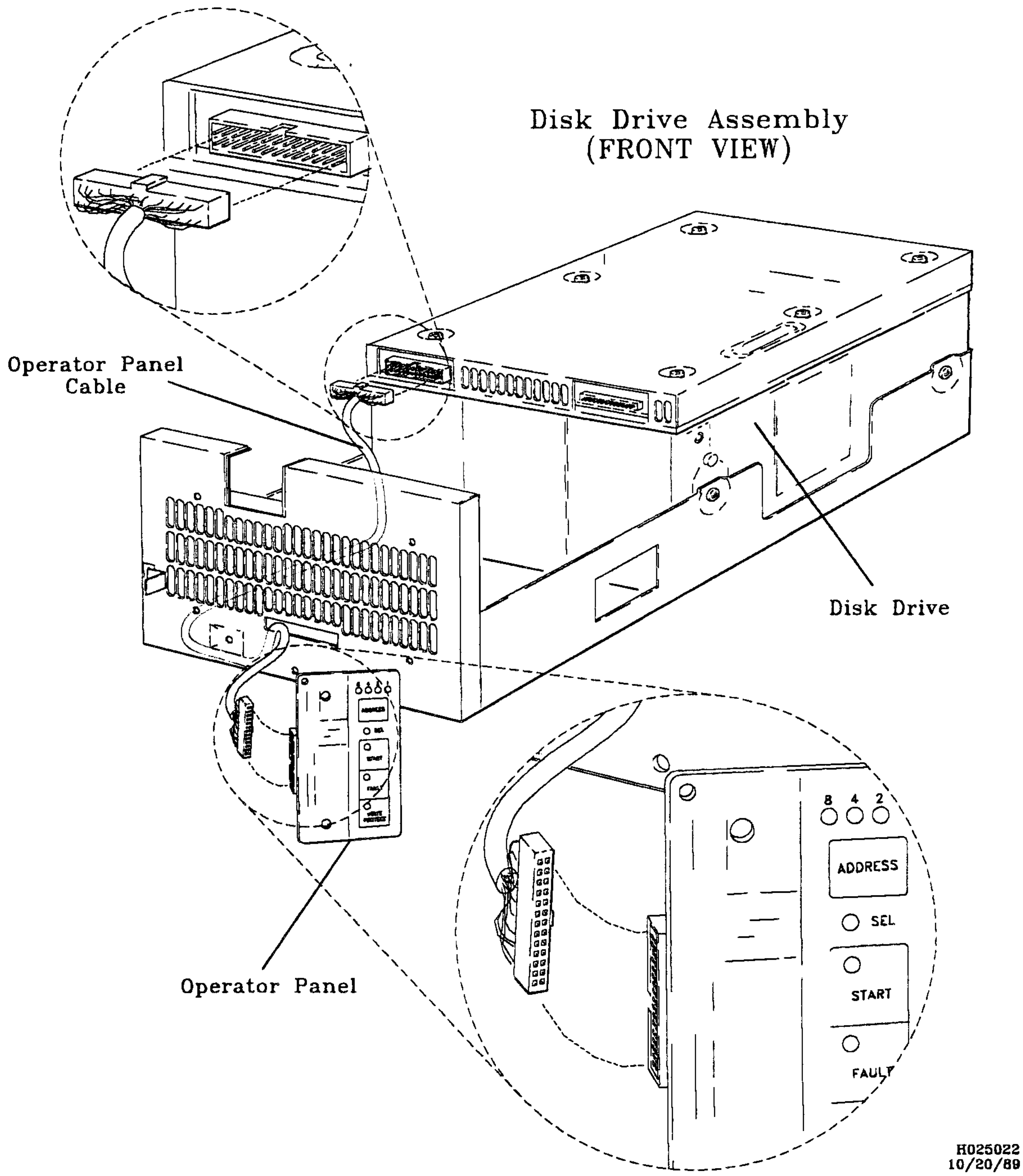
> **WARNING**
>
> Because a drive assembly weighs 60 lbs, personnel injury or equipment damage can occur when a drive assembly is removed or installed. Two people are required to remove or install a drive assembly.

19. Press the drive assembly tray lock to the right, then lower the drive assembly tray into the slide tray.
20. Install the drive assembly locking screw located on the front of the tray.

> **CAUTION**
>
> The operator panel can be damaged by Electrostatic Discharge (ESD). A grounded wrist strap (or other grounding method) must be used when handling the operator panel.

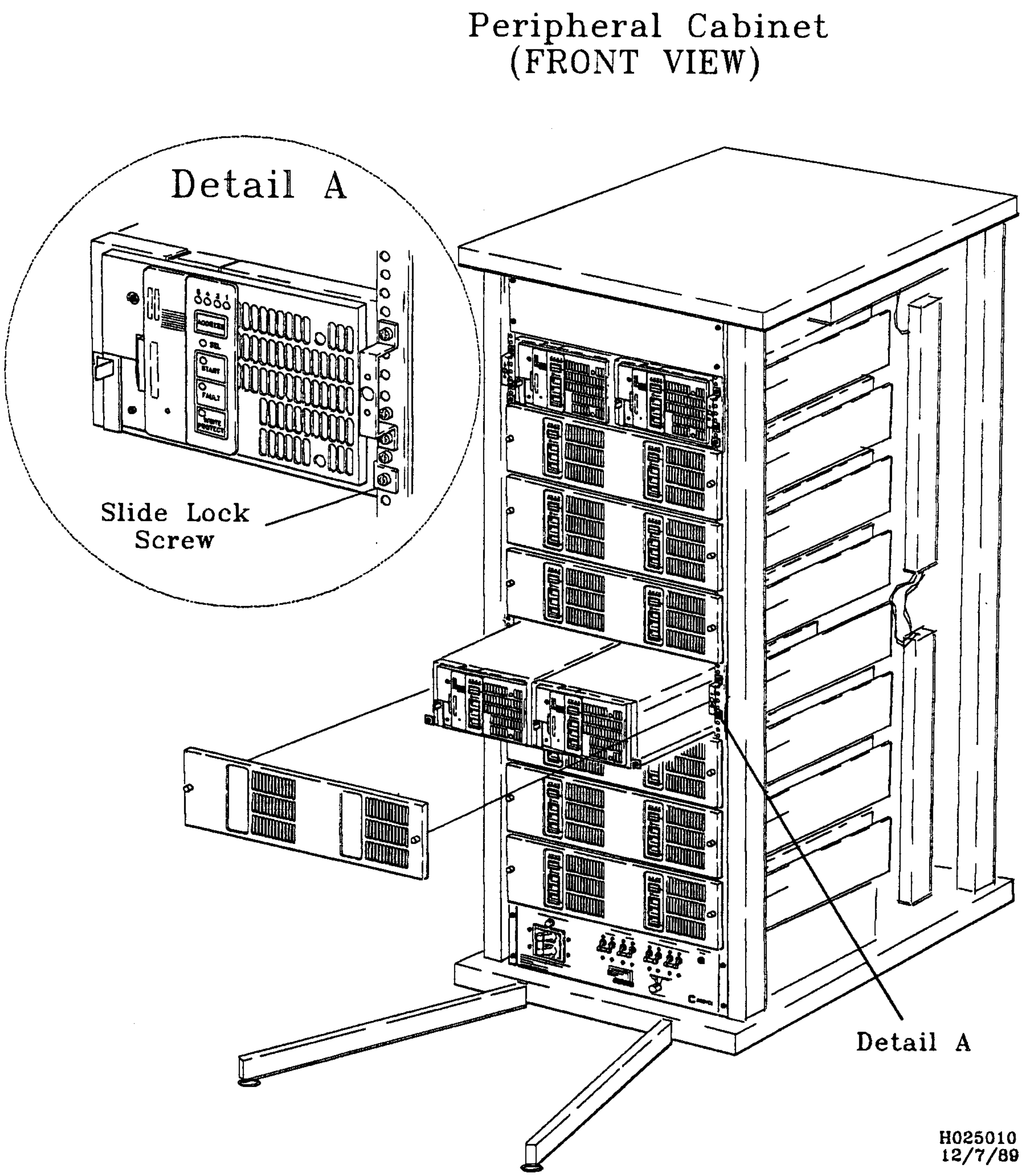
21. Connect the operator panel cable to the operator panel as shown in Figure 4–27, "Operator Panel Cable":

Figure 4–27, Operator Panel Cable



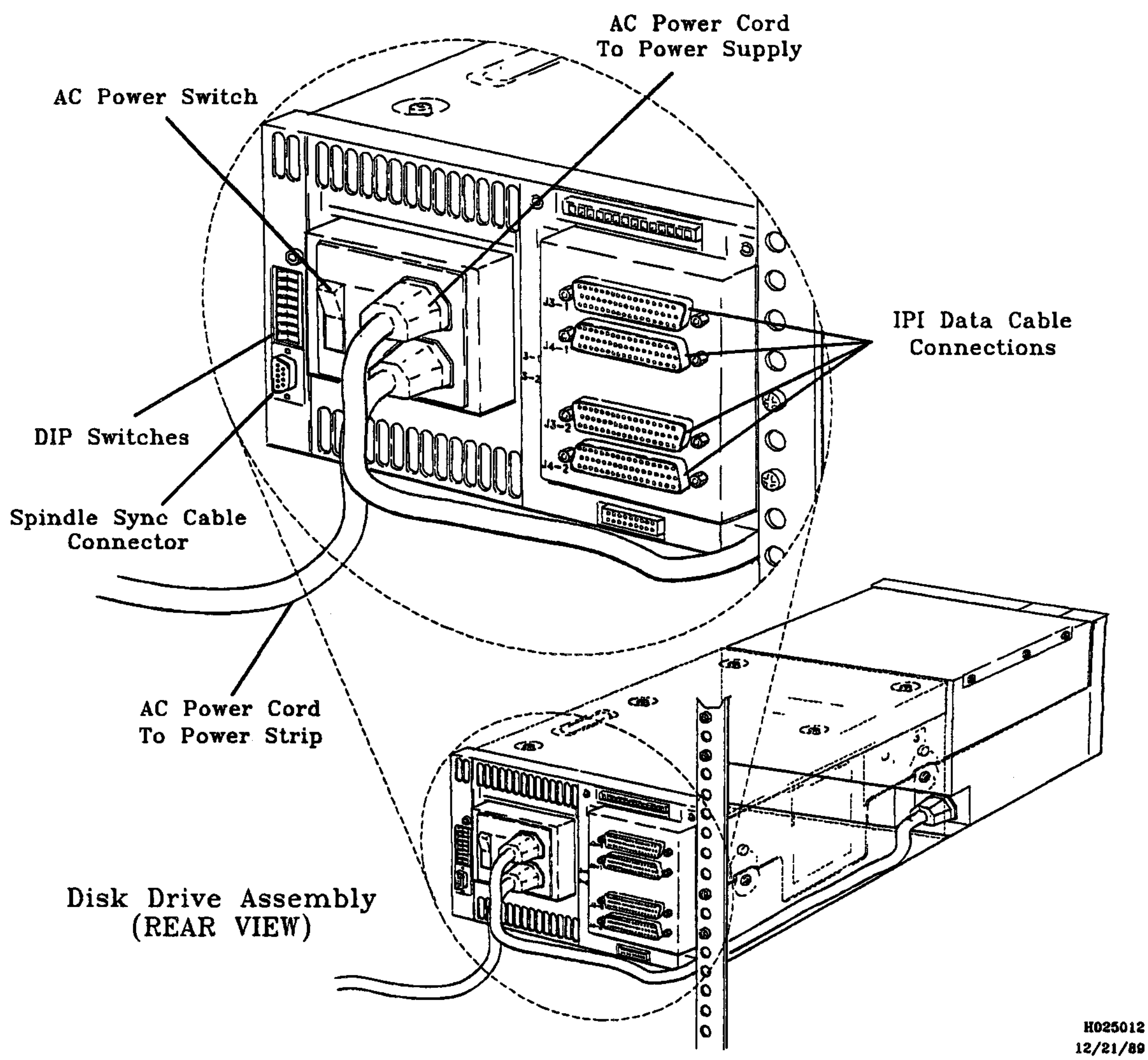
22. Install the 2 screws in the operator panel.
23. Retract the assembly on its slides and secure with the 2 front chassis captive-lock screws.
24. Install the front cover and secure with the 2 front cover captive-lock screws as shown in Figure 4–28, "Front Cover and Slide Lock":

Figure 4–28, Front Cover and Slide Lock



25. Connect the input data cable to the **J4-1** connector and the output data cable to the **J3-1** connector on the rear of the right drive. Also connect the spindle sync cable to the spindle sync connector on the rear of the right drive as shown in Figure 4–29, "Drive Connections".
26. Connect the input data cable to the **J4-1** connector and the output data cable to the **J3-1** connector on the rear of the left drive. Also connect the spindle sync cable to the spindle sync connector on the rear of the left drive.

Figure 4–29, Drive Connections



27. Connect the AC power cord to the AC power cord connector on the rear of the right drive. Then turn the AC power switch on the rear of the right drive to the **ON** position as shown in Figure 4–29, "Drive Connections".
28. Connect the AC power cord to the AC power cord connector on the rear of the left drive. Then turn the AC power switch on the rear of the left drive to the **ON** position.
29. Install the peripheral cabinet rear panel.
30. Set the power controller's main circuit breaker to the **ON** position and select local or remote operation with the LOCAL/REMOTE switch.
31. Turn the processor's front control panel key switch to the **ON** position.
32. Return the peripheral cabinet stabilizer bars to their retracted positions.

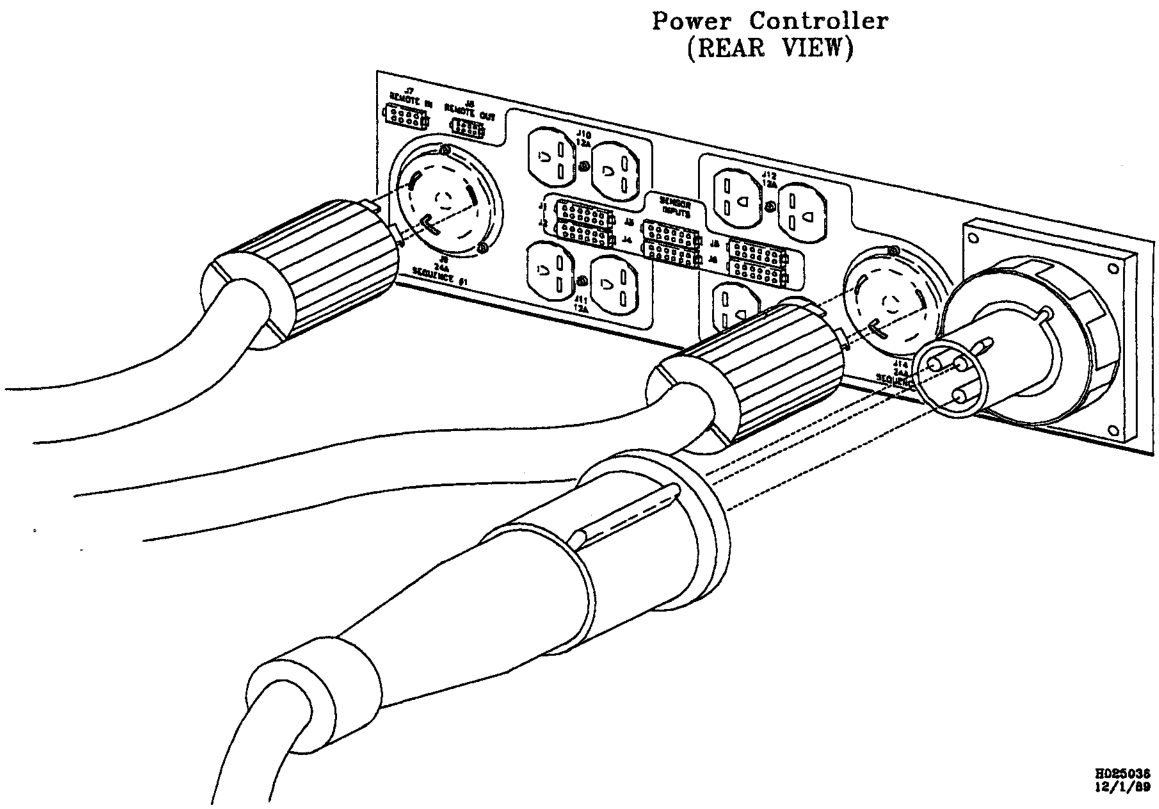
4.4.4 Power Controller

Follow these procedures to remove and replace the power controller.

4.4.4.1 Removal

1. Complete steps 1, 2, and 3 in Section 4.3, "Maintenance Safety Procedures."
2. Ensure that the power controller main circuit breaker is set to the **OFF** position.
3. Remove the peripheral cabinet rear panel.
4. Disconnect the AC power cable from the power controller AC-input jack as shown in Figure 4–30, "Power Controller":

Figure 4–30, Power Controller

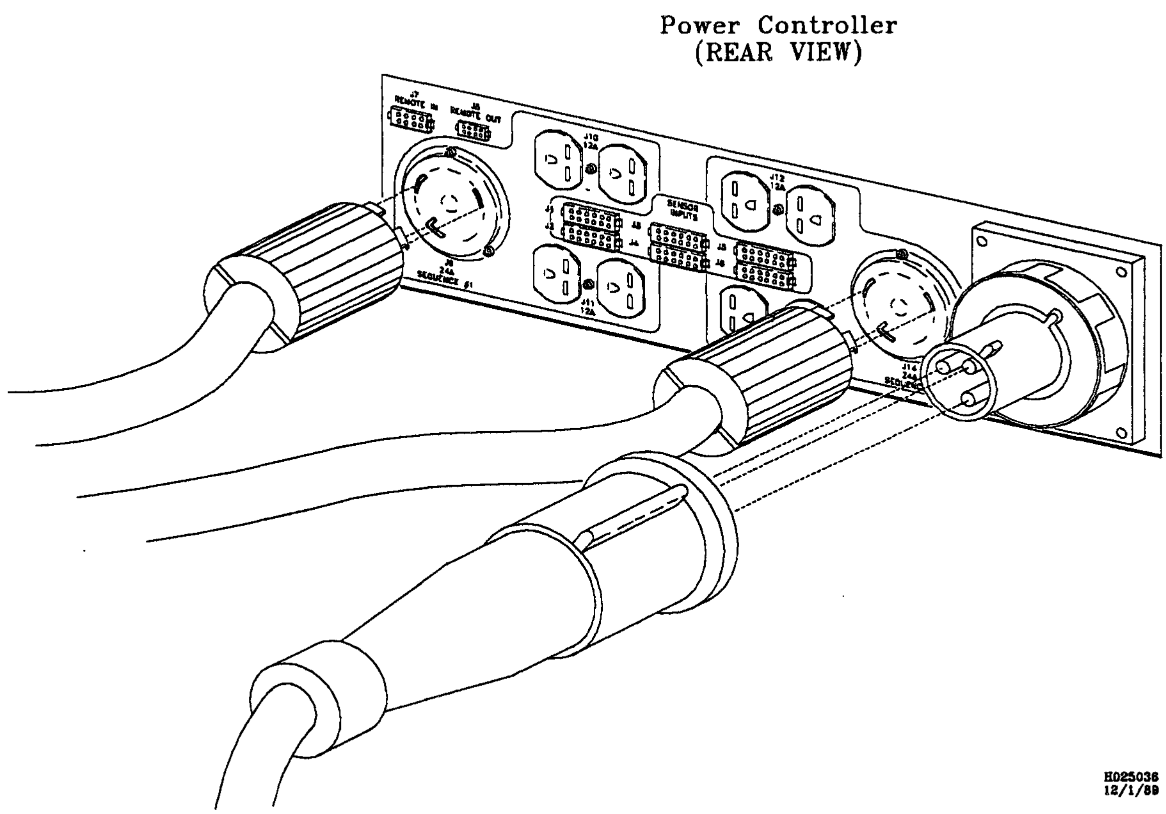


5. Disconnect the power strip twist-lock plugs from the twist-lock connectors.
6. Disconnect the fan sensor plugs from the **J1** and **J2** connectors.
7. Disconnect the thermostat plug from the **J3** connector.
8. Unplug and label all remaining cables and cords.
9. Remove the four screws on the front panel of the power controller.
10. Remove the power controller from the cabinet.

4.4.4.2 Replacement

1. Ensure that steps 1, 2, and 3 in Section 4.3, "Maintenance Safety Procedures," are completed.
2. Ensure that the power controller main circuit breaker is set to the **OFF** position.
3. Install the power controller in the cabinet.
4. Install the four screws on the front panel of the power controller.
5. Connect the AC power cable to the power controller AC-input jack as shown in Figure 4–31, "Power Controller":

Figure 4–31, Power Controller



6. Connect the power strip twist-lock plugs to the twist-lock connectors.
7. Connect the fan sensor plugs to the **J1** and **J2** connectors.
8. Connect the thermostat plug to the **J3** connector.
9. Plug in all remaining cables and cords.
10. Install the peripheral cabinet rear panel.
11. Set the power controller's main circuit breaker to the **ON** position and select local or remote operation with the LOCAL/REMOTE switch.
12. Turn the processor's front control panel key switch to the **ON** position.
13. Return the peripheral cabinet stabilizer bars to their retracted positions.

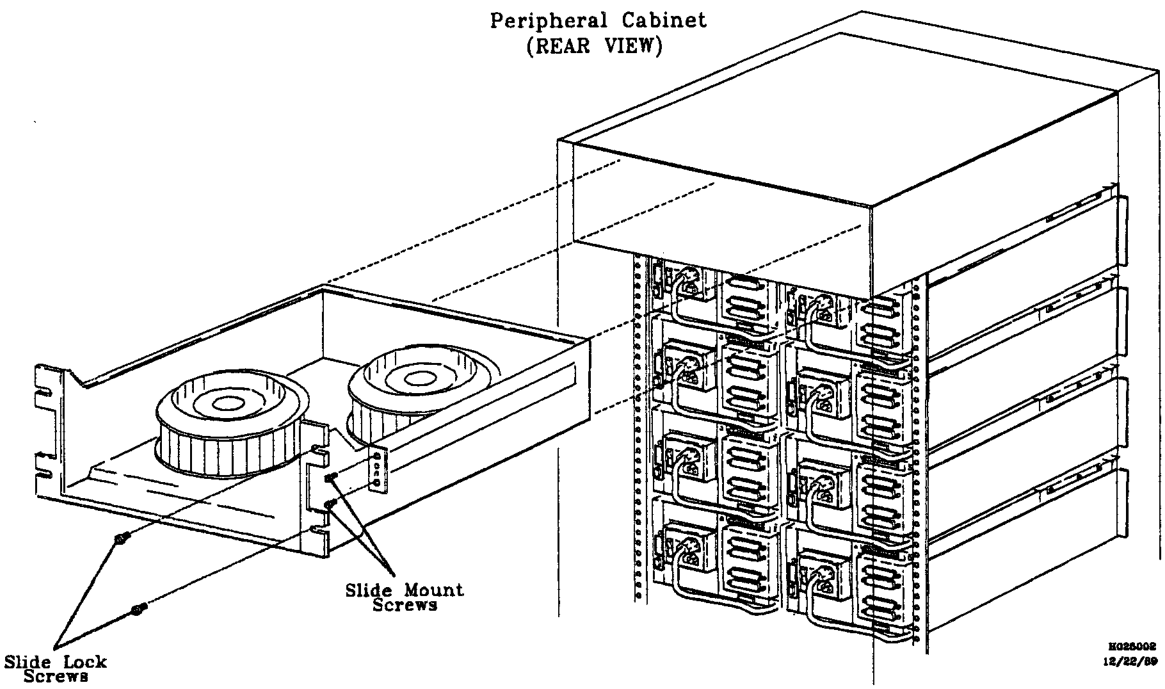
4.4.5 Fan Assembly

Follow these procedures to remove and replace the fan assembly.

4.4.5.1 Removal

1. Complete step 3 in Section 4.3, "Maintenance Safety Procedures."
2. Remove the rear panel of the peripheral cabinet.
3. Disconnect the AC input plug underneath the fan assembly.
4. Disconnect the fan sensor plug underneath the fan assembly.
5. Remove the 2 slide lock screws from the cabinet as shown in Figure 4–32, "Peripheral Cabinet Fan Assembly":

Figure 4–32, Peripheral Cabinet Fan Assembly



6. Remove the fan assembly from the rear of the cabinet.

4.4.5.2 Replacement

1. Ensure that step 3 in Section 4.3, "Maintenance Safety Procedures," is completed.
2. Install the fan assembly into the rear of the cabinet and secure it with 2 slide lock screws.

3. Connect the fan sensor plug underneath the fan assembly.
4. Connect the AC input plug underneath the fan assembly.

> **CAUTION**
>
> Do not operate the cabinet with its panels removed. The panels must be installed to obtain proper airflow inside the cabinet.

5. Install the peripheral cabinet rear panel.
6. Set the power controller's main circuit breaker to the **ON** position and select local or remote operation with the LOCAL/REMOTE switch.

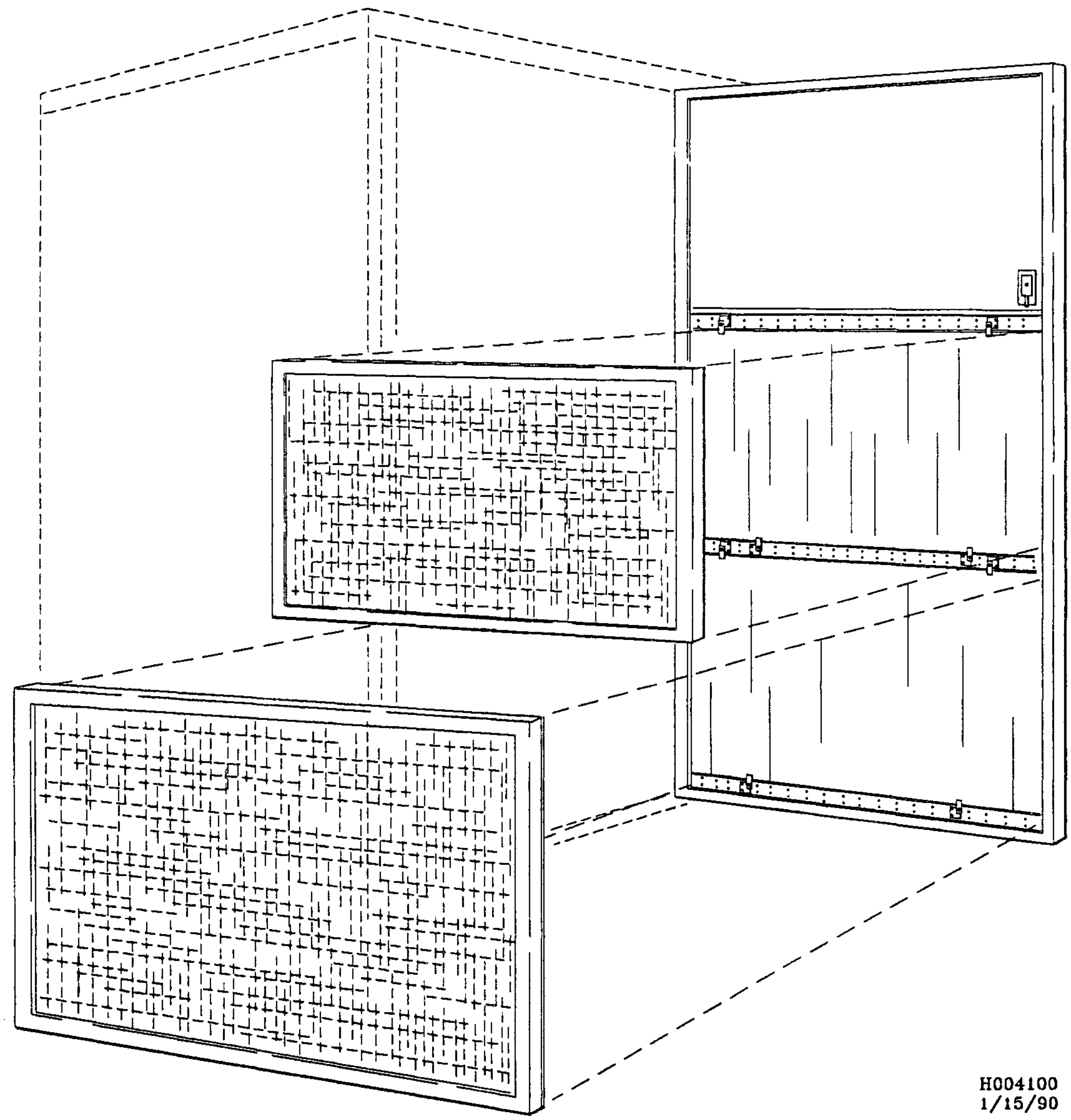
4.4.6 Peripheral Cabinet Air Filter

Follow these procedures to remove and replace the cabinet door air filters.

4.4.6.1 Removal

1. Release the 8 lock screws and remove the cabinet door air filter panels as shown in Figure 4–33, "Air Filter":

Figure 4–33, Air Filter



4.4.6.2 Replacement

1. Install the cabinet door air filter panels into the front door of the cabinet and secure with the 8 lock screws.

4.4.7 Disk Drive Air Filter

Follow these procedures to remove and replace a disk drive air filter.

4.4.7.1 Removal

1. Complete step 1 in Section 4.3, "Maintenance Safety Procedures."
2. Release the 2 front cover lock screws and remove the cover.
3. Release the 2 front assembly lock screws and extend the assembly one inch on its slides.

> **CAUTION**
>
> The operator panel can be damaged by Electrostatic Discharge (ESD). A grounded wrist strap (or other grounding method) must be used when handling the operator panel.

4. Remove the disk drive air filter from the air filter slot.

4.4.7.2 Replacement

1. Ensure that step 1 in Section 4.3, "Maintenance Safety Procedures," is completed.

> **CAUTION**
>
> The operator panel can be damaged by Electrostatic Discharge (ESD). A grounded wrist strap (or other grounding method) must be used when handling the operator panel.

2. Install the disk drive air filter into the air filter slot.
3. Return the assembly to its retracted position and secure it with the 2 lock screws.

> **CAUTION**
>
> Do not operate the cabinet with its panels removed. The panels must be installed to obtain proper airflow inside the cabinet.

4. Install the front cover and secure the cover with the 2 captive screws.
5. Return the peripheral cabinet stabilizer bars to their retracted positions.

4.5 Illustrated Parts List

This section is the Illustrated Parts Breakdowns (IPBs) for the IDC and peripheral equipment. Table 4–1, "Assembly List" lists the CONVEX part numbers, descriptions, and quantities of the assemblies for an IDC and a fully configured peripheral cabinet:

Table 4–1, Assembly List

Part Number	Description	Quantity
550-000295-200	Subsystem, IPI-2 Interface	1
550-000296-200	Subsystem, Disc 1.15 Gbyte IPI, 1st Disc	8
550-000296-201	Subsystem, Disk 1.15 Gbyte IPI, 2nd Disc	8
550-000245-200[1]	Disc Cabinet, Domestic	1
550-000244-200[1]	Disc Cabinet, International	1

[1] The domestic and international cabinets are the same except for the external power cord.

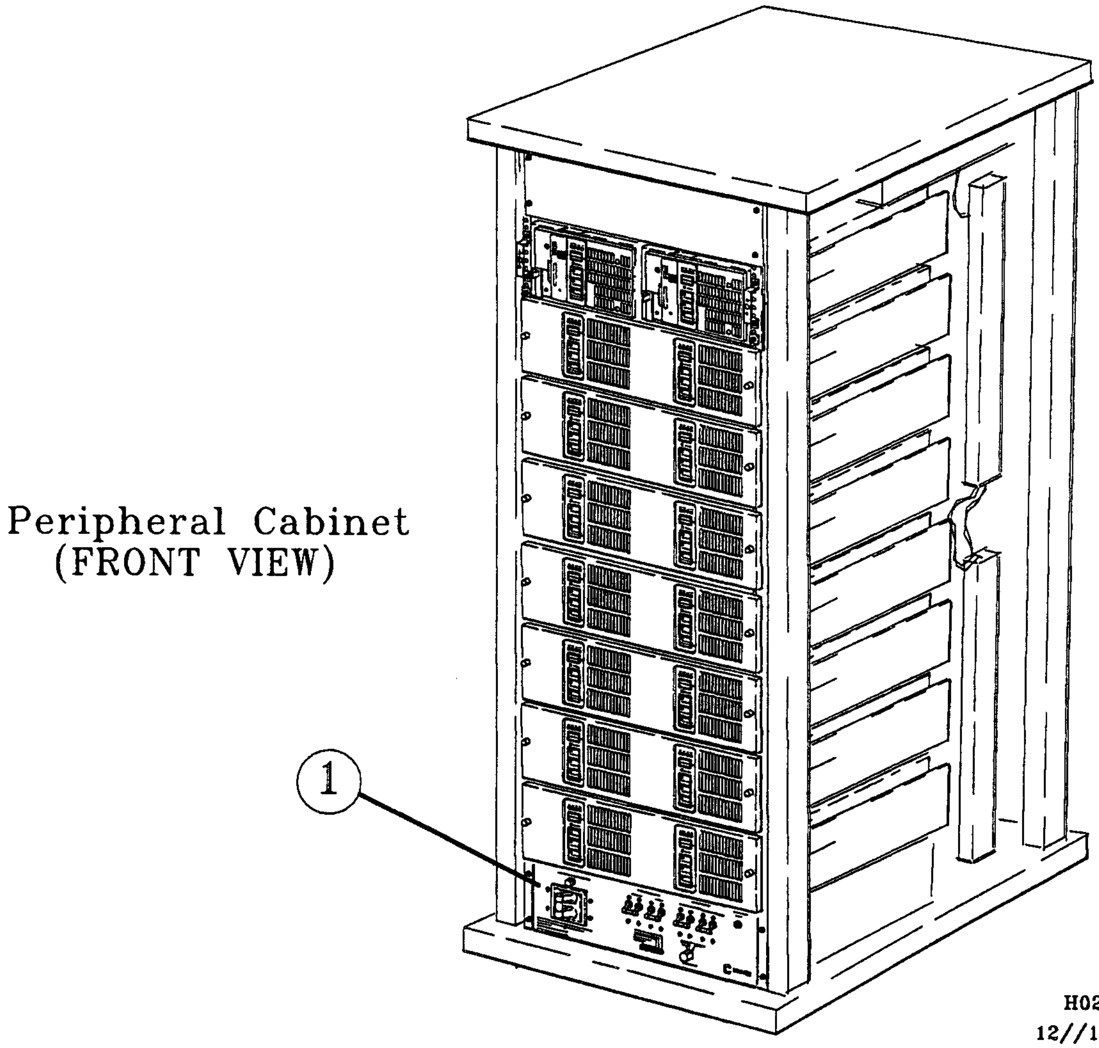
The following table lists the part numbers and descriptions for peripheral cabinet equipment:

Table 4–2, Peripheral Cabinet Parts List

Part Number	Description	Figure Number	Reference Number
500-001025-200	Power Controller Assembly	4-34	1
253-000100-001	Fuse MDA-1/2A 250V	-	-
253-000100-004	Fuse MDA-2A 250V	-	-
500-000297-200	Fan Assembly , 230V, Peripheral Cabinet	4-35	1
230-000012-001	Fan, 365 CFM 230V, W/Hall Effect	4-35	2
500-000296-200	Power Strip Assembly, 230V	4-36	1
614-000001-001	Cord, Power, 3 Conductor, 10 AWG, UL/CSA	4-36	2
500-000295-200	Thermostat Assembly, 220V Peripheral Cabinet	4-37	1
605-030005-200	Cord Assembly, 48A Power Controller, Domestic	-	-
614-000004-002	Cord, Power, 3 Conductor, 6AWG, UL/CSA	-	-
304-000036-001	Connector, Body, 60A, 2 Pole 3 Wire, IEC309	-	-
304-000038-001	Connector, Plug, 60A, 2 Pole 3 Wire, 250V	-	-
605-030006-200	Cord Assembly, 48A Power Controller, International[1]	-	-
614-000005-002	Cord, Power, 3 Cond, 10SQMM, Harmon	-	-
304-000036-001	Connector, Body, 60A, 2 Pole 3 Wire, IEC309	-	-

[1] The domestic and international peripheral cabinets are the same except for the power cord. The power cord for the international cabinet does not contain a plug on the site end of the cord.

Figure 4–34, Power Controller



H025050
12//18/89

Figure 4–35, Peripheral Cabinet Fan Assembly

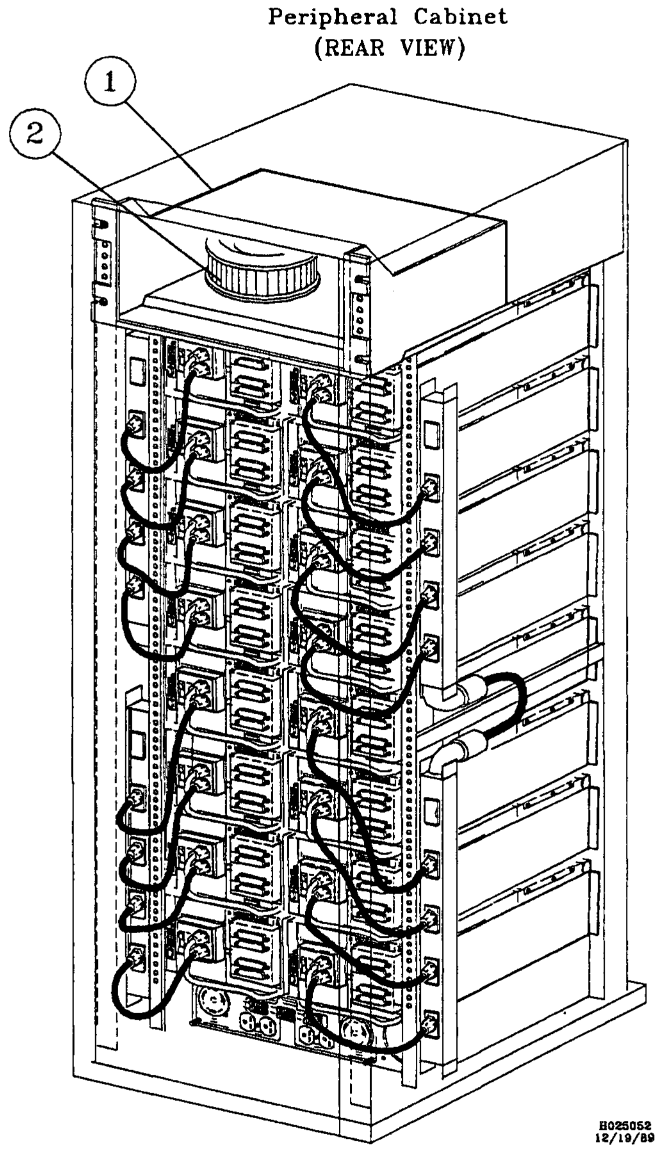


Figure 4–36, Peripheral Cabinet AC Cabling

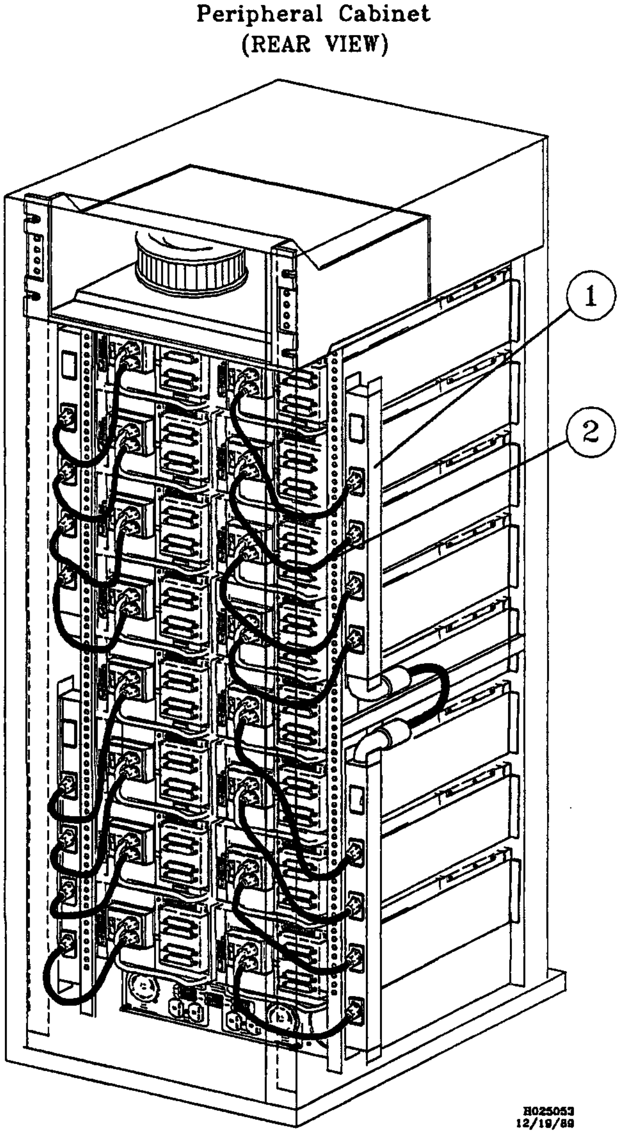
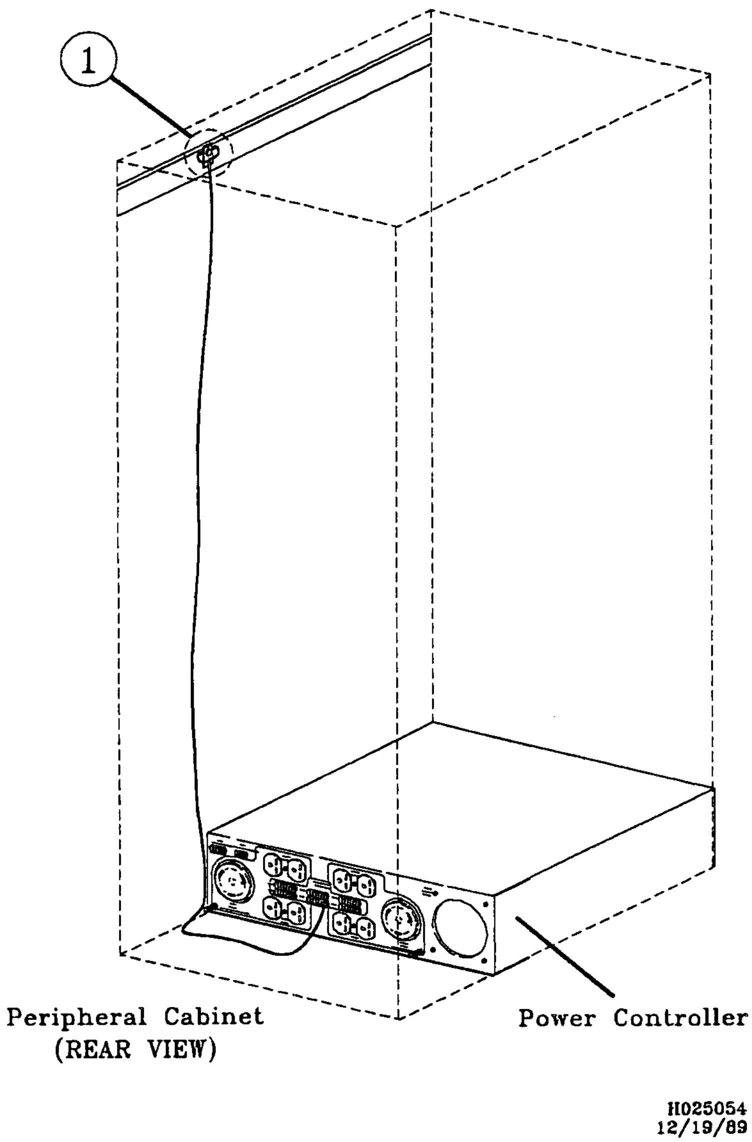


Figure 4–37, Peripheral Cabinet Thermostat Assembly



The following table lists the part numbers and descriptions for IPI-2 interface equipment:

Table 4–3, IPI-2 Interface Parts List

Part Number	Description	Figure Number	Reference Number
410-001228-200	Board Assembly, IDC	4-38	1
411-000202-200	Board Assembly, IPI Interface, Top	4-39	1
411-000201-200	Board Assembly, IPI Interface, Standard	4-39	2
601-640007-200	Cable Assembly, IPI Interface, 60/60	4-39	3
601-640006-200	Cable Assembly, IPI Interface, 60/40	4-39	4
601-640005-200	Cable Assembly, IPI Interface, Standard	4-39	5
601-640005-201	Cable Assembly, IPI Interface, 3 Key	4-39	6
320-000299-500	Connection Plate, 50 POS, D Shell	4-39	7
310-002301-001	Screw, 6-32 x 3/8, PHH, PNH, SEM	-	-
312-000124-002	Screw Lock, 4-40 x 1/4, D Connector, Female	-	-
204-000015-006	Cable, Sync Spindle, 1.5 ft.	4-40	1
204-000015-005	Cable, Sync Spindle, 3 ft.	4-40	1
204-000015-001	Cable, Sync Spindle, 6 ft.	4-40	1
204-000015-009	Cable, Sync Spindle, 10 ft.	4-40	1
204-000015-010	Cable, Sync Spindle, 25 ft.	4-40	1
604-500007-001	Cable, IPI Data, 1.5 ft.	4-40	2
604-500007-010	Cable, IPI Data, 10 ft.	4-40	3
604-500007-015	Cable, IPI Data, 15 ft.	4-40	3
604-500007-025	Cable, IPI Data, 25 ft.	4-40	3
604-500007-050	Cable, IPI Data, 50 ft.	4-40	3
204-000016-015	Terminator, Sync Spindle	4-40	4
204-000016-012	Terminator, IPI Data	4-40	5

Figure 4–38, IDC Board Assembly

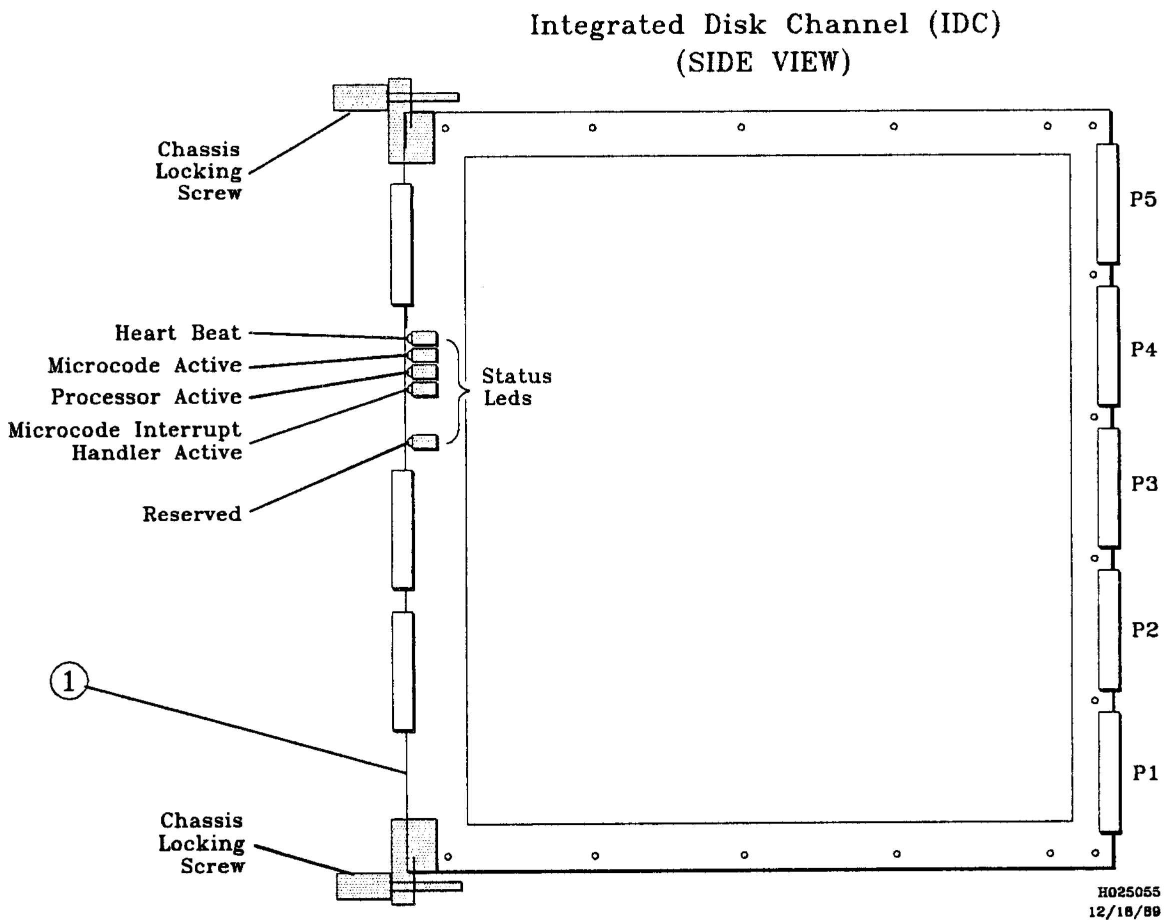
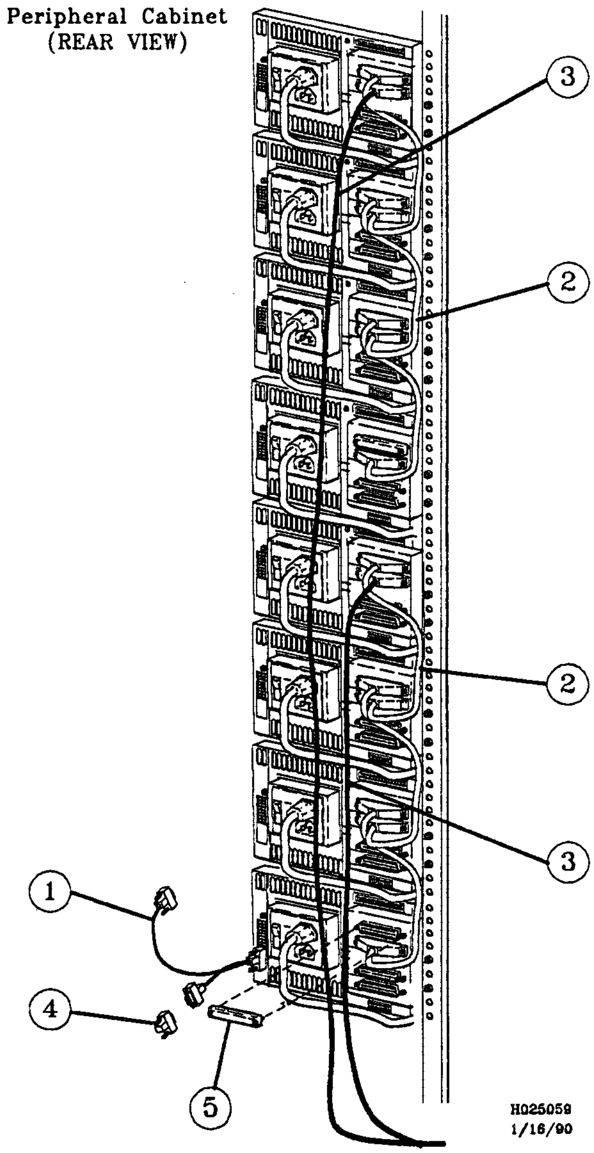


Figure 4–40, Peripheral Cabinet Drive Cabling



The following table lists the part numbers and descriptions for a disk drive assembly:

Table 4–4, Disk Drive Assembly Parts List

Part Number	Description	Figure Number	Reference Number
204-000016-200	Disk Drive 1.15 GB 2HP 6MB/Sec	4-41	1
204-000018-200	Disk Drive, IPI-2 1.23 Gbyte 3MB/Sec	4-41	1
204-000015-200	Disk Drive, IPI-2 1.15 GB 6MB/Sec	4-42	1
204-000017-200	Disk Drive, 1.23 GB 3MB/Sec	4-42	1
204-000016-009	Drawer, Inner, 1.15 Gbyte Disk Drive	4-42	2
204-000016-010	Power Supply, IPI-2HP Disk Drive	4-42	3
204-000016-008	Cable, DC Power, 2.5 in.	4-42	4
204-000016-007	Cable, AC Power, 3 Conductor Shield	4-42	5
204-000016-004	Air Baffle, 1.15 Gbyte Disk Drive	4-42	6
900-000340-001	Screw, 10-32 x 3/8, Shock Mount	4-42	7
900-000342-001	Filter, Primary, Disk Drive	4-42	8
900-000343-001	Panel, Operator, Disk Drive	4-42	9
900-000345-001	Fan, 24 VDC, Disk Drive	-	-
204-000016-005	Clamp, Power Cable	-	-
204-000016-002	Screw, 6-32 x 1/4, Hex	-	-
204-000016-003	Screw, 8-32 x 1/2, Hex	-	-
204-000016-006	Washer, Lock #6	-	-
204-000016-001	Cable, DC Ground	-	-
204-000016-014	Ground Straps Kit, Shock Mount	-	-
250-000011-001	Switch, Power Connector	-	-
900-000341-001	Panel, Status/control, Disk Drive (optional)	-	-
900-000344-001	Input/Output Board, Disk Drive	-	-
900-000346-001	Control Board, Disk Drive	-	-
900-000421-001	Manual, Service, 8 in. Disk Drive	-	-

Figure 4–41, Disk Drive Subsystem Assembly

Disk Drive Subsystem
(FRONT VIEW)

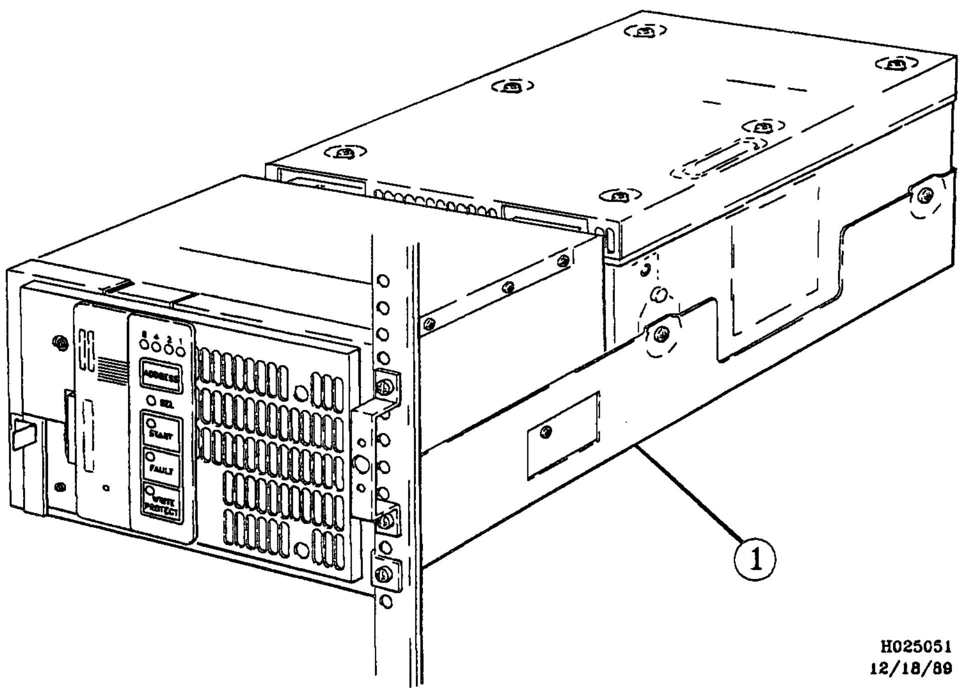
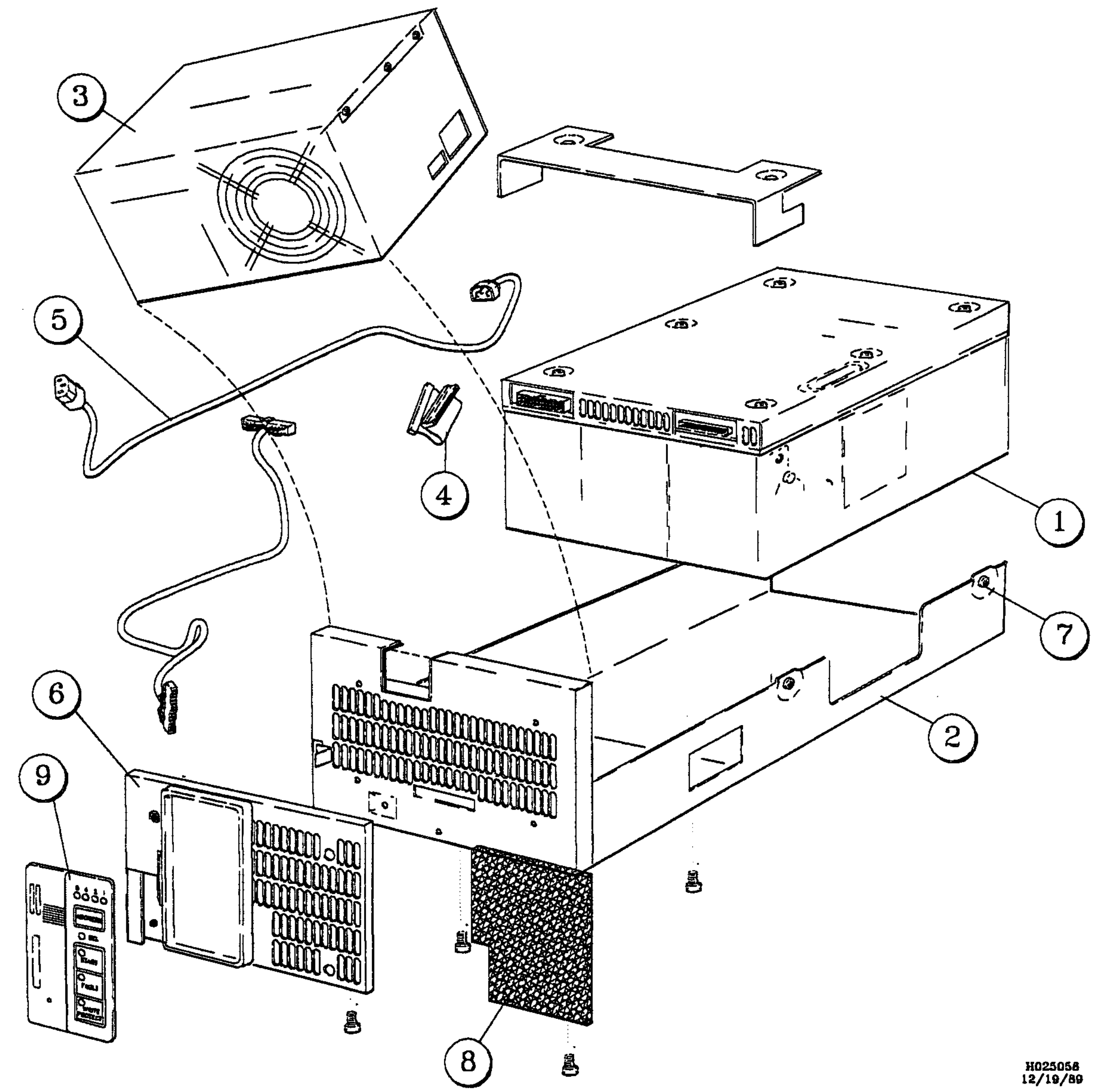


Figure 4–42, Disk Drive Assembly



The following table lists the part numbers and descriptions for a disk drive mount assembly:

Table 4–5, Drive Mount Assembly List

Part Number	Description
204-000014-009	Tray Assembly, Rack Mount, Double
204-000014-001	Tray, Double, Imprimis
204-000014-002	Slide, Guide, Right
204-000014-003	Slide, Guide, Left
204-000014-004	Bracket, Slide Mounting, Right Rear
204-000014-005	Bracket, Slide Mounting, Left Rear
204-000014-006	Nut Plate, Imprimis
204-000014-007	Bracket, Clamp, Imprimis
204-000014-008	Screw, 10-32 x 1/2 Hex, Socket

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A
Reporting Problems

A.1 Overview

This appendix introduces the CONVEX Technical Assistance Center (TAC) and the *contact* utility. The *contact* utility is an online system for reporting problems to the TAC. To learn *contact* by using it, enter **contact** at the system prompt and then answer the questions as they appear on the screen. To find out more about using *contact*, read through this appendix. It describes prerequisites and tips for using *contact* and the step-by-step process *contact* takes you through.

A.2 Technical Assistance Center

The CONVEX Technical Assistance Center (TAC) is staffed by technical specialists who can address the diverse questions and problems that arise in a supercomputing environment. If you have a hardware, software, or documentation problem, contact the TAC. This group stands ready to solve such problems.

A.3 The *contact* Utility

The TAC recommends using the *contact* utility to report a hardware, software, or documentation problem. The *contact* utility is an interactive utility that helps the TAC track reports and route them to the the CONVEX personnel most qualified to fix them.

After invoking *contact*, it prompts for information about the problem. When you finish your report, *contact* electronically mails it to the TAC. You are notified within 48 hours that the TAC has received your report.

A.4 Prerequisites

To use *contact* requires

- a UNIX-to-UNIX Communication Protocol (UUCP) connection to the TAC
- the full path name of the program or utility in question
- the version number of the program or utility in question

A.4.1 UUCP Connection

Before using *contact*, check with your system administrator to be sure there is a UUCP connection to the TAC. A UUCP connection allows files to be copied from one UNIX system to another. The *uucp* (UNIX-to-UNIX copy) command relies on either a dial-up or hard-wired UUCP communication line.

A.4.2 Finding the Program Path Name

To determine the full path name of the program or utility in question, use the *which* command. The following screen illustrates using the *which* command to find the full path name of the loader (*ld*) utility:

```
>which ld
/bin/ld
>
```

In this example, the full path name of the loader is */bin/ld*.

For more information on the *which* command, refer to the *which*(1) man page. You can also use the *info* online information system. Enter **info which** at the system prompt. If you use the C shell (*csh*), you can also use the *whence* command to find the program path name. The *whence* command works like *which*, only faster.

A.4.3 Finding the Program Version Number

To determine the version number of the program or utility in question, use the *vers* command. The following screen illustrates using the *vers* command (enter **vers**, then the path name of the program or utility) to find the version number of the loader (*ld*) utility.

```
>vers /bin/ld
/bin/ld: 7.0
>
```

In this example, the loader utility version number is 7.0.

For more information on the *vers* command, refer to the *vers*(1) man page. You can also use the *info* online information system. To do so, enter **info vers** at the system prompt.

A.5 Tips on Using the *contact* Utility

The *contact* utility is interactive and easy to use. This section lists tips to help use it efficiently. In particular, this section tells how to

- use a *.contact* file
- abort a contact session
- resubmit an aborted report
- suspend a contact session
- move from one prompt to another
- use tilde-escape sequences in the *contact* utility

A.5.1 Using a *.contact* File

When invoked, *contact* prompts for information regarding the problem. The first prompt is for your name, title, phone number, and company name. You can, however, create a *.contact* file to skip this first prompt. Follow these steps:

1. Create a *.contact* file in your home directory.
2. Enter your name, job title, phone number, and company name, each on a new line.

When you invoke *contact*, it automatically includes the *.contact* file as input for the first prompt and proceeds to the next prompt.

A.5.2 Aborting the Report

To abort a contact report, either enter the interrupt key (usually CTRL-C) or choose the abort option when prompted by the *contact* utility. Using CTRL-C to abort does not save the contents of the report. Using the abort option saves the contents of the report in a file named *dead.report* in your home directory.

A.5.3 Submitting the *dead.report* File

When aborting a contact session, the *contact* utility saves the report in a file named *dead.report* in your home directory. Using the *contact* command with the *-r* option automatically merges the contents of the *dead.report* file into the new contact session. Enter

contact -r

and *contact* finds the *dead.report* file in your home directory and merges it into the contact report. You can then edit the report. When you end the editing session, *contact* returns to the final prompt, which asks you to review, edit, submit, or abort the report.

A.5.4 Suspending a Report

Sometimes it is necessary to stop in the middle of a contact report and return to the shell (for instance, to suspend the contact session to find the program path name or version number). To suspend the contact session, press CTRL-Z. To return to the contact session, enter **fg**. Using CTRL-Z and the *fg* (foreground) command lets you switch back and forth between the *contact* utility and the shell. You cannot, however, use CTRL-Z and *fg* to switch back and forth if you are using a Bourne shell (*sh*).

A.5.5 Ending a Response

The *contact* utility prompts for information pertinent to your hardware, software, or documentation question. Some prompts require one-line responses; to move to the next prompt, press RETURN. Other prompts require more than a one-line response; to move to the next prompt, press CTRL-D.

A.5.6 Tilde-Escape Sequences

The *contact* utility treats input beginning with a tilde (~) as a special sequence. The character following the tilde is considered a request for a special function. The following tilde sequences are recognized by *contact*:

~e Start the text editor (defined in your EDITOR environment variable).

~h Display a list of available tilde-escape sequences.

~p Print the contact report to the terminal screen.

~r *filename* Read the contents of *filename* as a response to the current prompt. Some prompts require only a one-line response. This tilde-escape sequence only works for prompts that allow more a than one-line response.

~~ Insert a single tilde as the first character in the line.

A.6 Using the *contact* Utility

The *contact* utility prompts for the following information:

- your name, title, phone number, and corporate name
- the name and version of the product involved
- a one-line summary of the problem
- a detailed description of the problem
- the priority of the problem
- instructions on how to reproduce the problem
- comments about the problem
- comments about the documentation supporting the problem
- files to include in the contact report

The following is a step-by-step discussion of these prompts:

1a. To invoke the *contact* utility, enter **contact** at the system prompt. The system responds with a welcome message and a series of questions regarding your hardware, software, or documentation question. The following screen illustrates the *contact* command and the system response:

```
>contact
Welcome to contact version 0.11 ()

Enter your name, title, phone number, and corporate name (^D to terminate)
>
```

1b. If there is a *.contact* file in your home directory, *contact* skips the first prompt. The following screen illustrates the *contact* command and the system response when a *.contact* file is in your home directory:

```
>contact
Welcome to contact version 0.11 ()

Enter the name of the product involved
>
```

2. The *contact* utility prompts for the version number of the product. If you do not know the version number, use CTRL-Z to suspend the session. Use the *which* (or *whence* if using *csh*) and *vers* commands to find the version number of the product. Use the *fg* command to return to the session and enter the version number in the form X.X or X.X.X.X.
3. The *contact* utility prompts for a one-line summary of the problem. This summary is the subject header in any further correspondence regarding the problem. Make this summary as descriptive as possible in one line.
4. The *contact* utility prompts for a detailed description of the problem. Make this description as complete as possible. Include source code and a stack backtrace whenever possible. (Refer to the *adb*(1) or *csd*(1) man page for information on obtaining a stack backtrace.) The more information provided, the quicker the TAC can isolate and solve the problem.
5. The *contact* utility prompts for the priority of the problem. The following screen illustrates this prompt and the priority levels from which to choose; you must enter a priority number.

```
Enter a problem priority, based on the following:
1) Critical    - work cannot proceed until the problem is resolved.
2) Serious     - work can proceed around the problem, with difficulty.
3) Necessary   - problem has to be fixed.
4) Annoying    - problem is bothersome.
5) Enhancement - requested enhancement.
6) Informative - for informational purposes only.
>
```

6. The *contact* utility prompts for an explanation of how to reproduce the problem. Include the command syntax and options you used and anything else you did to make your program run.
7. The *contact* utility prompts for any other pertinent comments. Include any relevant information.
8. The *contact* utility prompts for suggestions regarding the documentation supporting the product. Indicate if the documentation could be revised to address the question.
9. The *contact* utility asks for the names of files necessary to reproduce the problem. The following screen illustrates the *contact* prompt and sample user response:

```
Are there any files that should be included in this report (yes | no)?
>yes
Please enter the names of the files, one to a line (^D to terminate)
>test.f
>~/subroutines/sub.f
>
```

> **NOTE**
>
> Tilde-escape sequences are not recognized in responses to this prompt. Instead, *contact* treats a tilde in this section to mean your home directory. This convention is based on use of the tilde for expanding file names in *csh*.

If the files specified are small text files, they are automatically included in the contact report. If the files are too big to be included in this report, *contact* gives further instructions on how to submit these files.

To specify a directory, combine the directory files into a single file using the *tar* command (refer to the *tar*(1) man page for further information) or enter each file name in the directory on a single line in the contact report.

10. The *contact* utility prompts you to review, edit, submit, or abort the contact report. The following screen illustrates this prompt:

```
Please select one of the following options:
1) Review the problem report.
2) Edit the problem report.
3) Submit the problem report.
4) Abort the problem report.
>
```

Choose the number of the option you want to select. These options let you do the following:

Review — Review the text of your contact report. You are then prompted again to select an option.

Edit — Edit the text of the contact report. If you choose to edit the report, *contact* puts you in your default text editor.

Submit — Send the report to the CONVEX TAC. You are notified within 48 hours that the TAC has received the report. This option exits the *contact* utility and returns you to the shell environment.

Abort — Save the text of your report in a file named *dead.report* in your home directory. This option exits the *contact* utility and returns you to the shell environment.

Index

A

B

C

D

E

F

H

I

J

L

M

N

O

P

R

S

T

U

V

W

CONVEX Integrated Disk Channel Service Guide
Document No. 081-001530-001, Second Edition

Electronic Mail

Using Electronic Mail

The Hardware Documentation Group has an email address for documentation comments. Use this service to give us a quick response mechanism if you have special documentation questions that you would like addressed immediately. If you have a technical question, you should still contact the Technical Assistance Center, as described previously. To use email response service, just send mail addressed to:

`cnvxhwdoc@convex.COM`

We will read your comments and give you a personal reply.

What to Include in an Email Message

When you use the electronic mail service, please provide the following information:

- The reader's name and company name
- A return email address in INTERNET notation or UUCP (bang) notation
- The manual that is being critiqued
- The chapter and page number in question
- The comment

Reader's Forum

If you wish to mail your comments to us, please use the form on the next page and list the document page number with your questions and comments. Thank you.

THIS PAGE INTENTIONALLY LEFT BLANK

CONVEX Integrated Disk Channel Service Guide
Document No. 081-001530-001, Second Edition

Reader's Forum

Please use this form to submit comments or questions concerning the clarity and service of this manual. Constructive critical comments are most welcome and help us continue in our efforts to generate quality customer documentation. Please list the page number for questions or comments.

____

____

____

____

____

____

____

____

From:

Name________________________ Title________________________

Company________________________ Date________________________

Address and Phone No.________________________

FOR ADDITIONAL INFORMATION OR DOCUMENTATION:

Location	Phone Number
From all locations in continental U.S.	1(800)952-0379
From locations in Alaska, Hawaii, & Canada	1(214)497-4379
From all other locations	Contact nearest CONVEX office

Direct mail orders to:

CONVEX Computer Corporation
Customer Service
PO Box 833851
Richardson TX 75083-3851 USA

(Fold Here First)

CONVEX

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1046 RICHARDSON, TEXAS

POSTAGE WILL BE PAID BY ADDRESSEE

CONVEX Computer Corporation
Customer Service
PO Box 833851
Richardson TX 75083-3851

(Fold Here Second)

(Tape or Staple)