CONVEX Integrated Disk/Tape Channel Service Guide

Order No. DHW-025

Third Edition
March 1992

CONVEX Press
Richardson, Texas
United States of America

CONVEX Integrated Disk/Tape Channel
Service Guide
Order No. DHW-025

Printed in the United States of America

Revision history

CONVEX Integrated Disk/Tape Channel Service Guide

Edition	Document No.	Description
Third	081-001530-002	Released March 1992. This document was formerly titled *CONVEX Integrated Disk Channel Service Guide.* Added information on integrated tape channel (ITC). Removed Appendix B, "Reporting Problems." Corrected errors on Figures 1-10, 2-11, 2-14, 4-9 (operator panel cable), 4-30 (power controller), and 4-38 (IDC board assembly). Minor corrections on shutdown input indicators, pin settings, voltage specifications, and /ioconfig entries.
Second	081-001530-001	Released June 1990.
First	081-001530-000	Released January 1990.

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, it may cause harmful interference to radio communications.

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment.

When this equipment is operated in a residential area, it is likely to cause interference. In this case, the interference must be corrected at the operator's expense.

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

Table of Contents

List of Tables

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

Purpose and audience

The *CONVEX Integrated Disk/Tape Channel Service Guide* provides a general overview of the CONVEX integrated disk/tape channel (IDC/ITC) subsystem and related hardware and how to:

- Install a CONVEX IDC/ITC and related hardware.
- Integrate a CONVEX IDC/ITC and related hardware into the CONVEX operating system (ConvexOS).
- Test a CONVEX IDC/ITC and related hardware.
- Remove and replace a CONVEX IDC/ITC and related hardware.

This document is intended for:

- CONVEX customer support engineers and CONVEX manufacturing personnel
- Customers who install or maintain their own CONVEX supercomputer systems

Organization

This document consists of the following sections:

- **Chapter 1**, "**Description and specifications**"—Describes the CONVEX IDC/ITC 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 peripheral cabinet, the IDC/ITC board assembly, and related hardware.
- **Chapter 3**, "**Software and test**"—Explains how to integrate a CONVEX IDC/ITC and related hardware into ConvexOS and how to test the CONVEX IDC/ITC.
- **Chapter 4**, "**Maintenance procedures and IPB**"—Provides removal and replacement instructions for a CONVEX IDC/ITC and related hardware.

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

ITC—integrated tape channel

MPU—main processing unit

PBI—PBUS interface

PIGA—PBUS interface gate array

Notational conventions

Notational conventions are systems of characters, symbols, terms, or abbreviated expressions used to express technical facts or quantities as established by this guide. The following notational conventions are used in this document:

- **Boldface** indicates user-entered information for a computer program that should be entered exactly as it appears.
- *Italic* is used to define new terms, for user-supplied variables, for emphasis, and to indicate titles of publications.
- `Constant-width` is used for code examples, command names and options, error messages, screen output, and system calls.
- Bit numbering is left to right, N1 through 0. The most significant numerical bit is N-1, the least significant 0. The bit numbering represents the binary weight of a position.
- Bit fields are specified using the following convention: *name*<*x..y*> where the bit field is *name* from bits *x* through *y*.

All CONVEX illustrations have an illustration file number at the bottom right-hand corner that is for CONVEX use only.

Examples of typical warnings, cautions and notes as used in CONVEX documents follow:

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

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

> **NOTE**
>
> A note highlights information of a supplemental nature. The note immediately precedes or follows the highlighted information.

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:

- C100/200 Series—*CONVEX Site Preparation Guide*, Order No. DHW-009
- C3400 Series—*CONVEX C3410 thru C3440 Series Computers Site Preparation Guide*, Order No. DHW-221
- C3800 Series—*CONVEX Site Preparation Guide (C3800 Series)*, Order No. DHW-211
- *CONVEX I/O Site Preparation Guide*, Order No. DHW-010
- *Managing ConvexOS: Configuration Guide*, Order No. DSW-030
- *CONVEX Integrated Tape Channel (itc4000) Diagnostics Manual*, Order No. DHW-285
- *CONVEX Integrated Disk Channel (idc4010) Diagnostics Manual*, Order No. DHW-286

Ordering documentation

To order the current edition of this or any other CONVEX document, send requests to:

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

Include the order number with the request. The order number is on the title page of the manual and begins with the letters "DHW" or "DSW."

Technical assistance

Hardware, software, and documentation 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 Canada, call 1(800)345-2384.
- From all other locations, contact the nearest CONVEX office.

The TAC recommends using the `contact` utility to report a hardware, software, or documentation problem. The `contact` utility is an interactive program that helps the TAC track reports and route them to the CONVEX personnel most qualified to fix a problem. After you invoke `contact`, it prompts you for information about the problem. When you finish your report, `contact` mails it to the TAC electronically.

The TAC notifies you within 48 hours that your report has been received. Using `contact` requires:

- UNIX-to-UNIX Communications Protocol (UUCP) connection to the TAC
- Full path name of the program or utility in question
- Version number of the program or utility in question

Refer to the contact(1) man page for complete details.

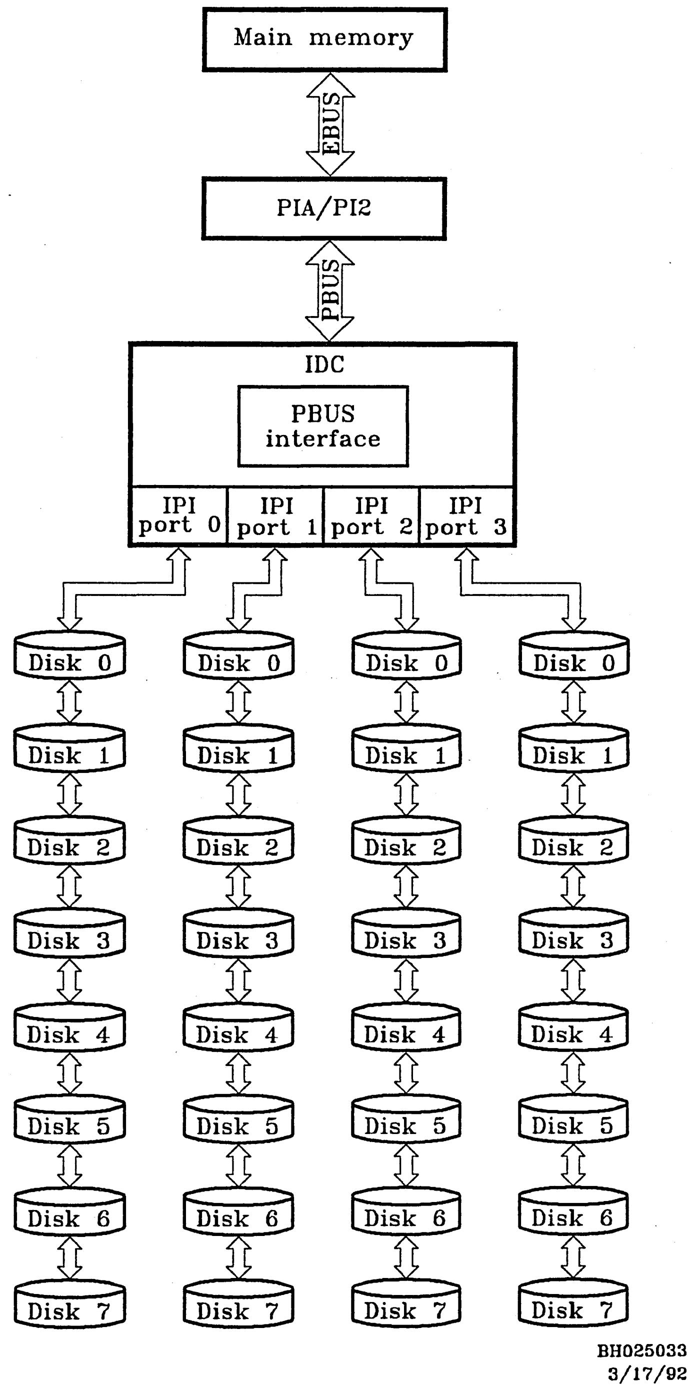
Chapter 1
Description and specifications

1.1 Overview

The integrated disk/tape channel (IDC/ITC) is a high-capacity, high-performance channel control unit (CCU). The IDC/ITC includes four peripheral interface ports that conform to the ANSI intelligent peripheral interface (IPI) physical interface specification. Each port is capable of supporting eight disk drives or tape drives.

The IDC/ITC communicates with the operating system kernel via the message based system (MBS) and interrupts. The IPI-2 logical command set for magnetic disk or IPI-3 logical command set for tape is implemented with a device driver in main processing unit (MPU) software. Physically, an IDC/ITC consists of a PBUS interface to main memory, an onboard MPU, a high-bandwidth multiplexing data buffer, and four IPI ports. An IDC/ITC subsystem block diagram is illustrated in Figure 1–1.

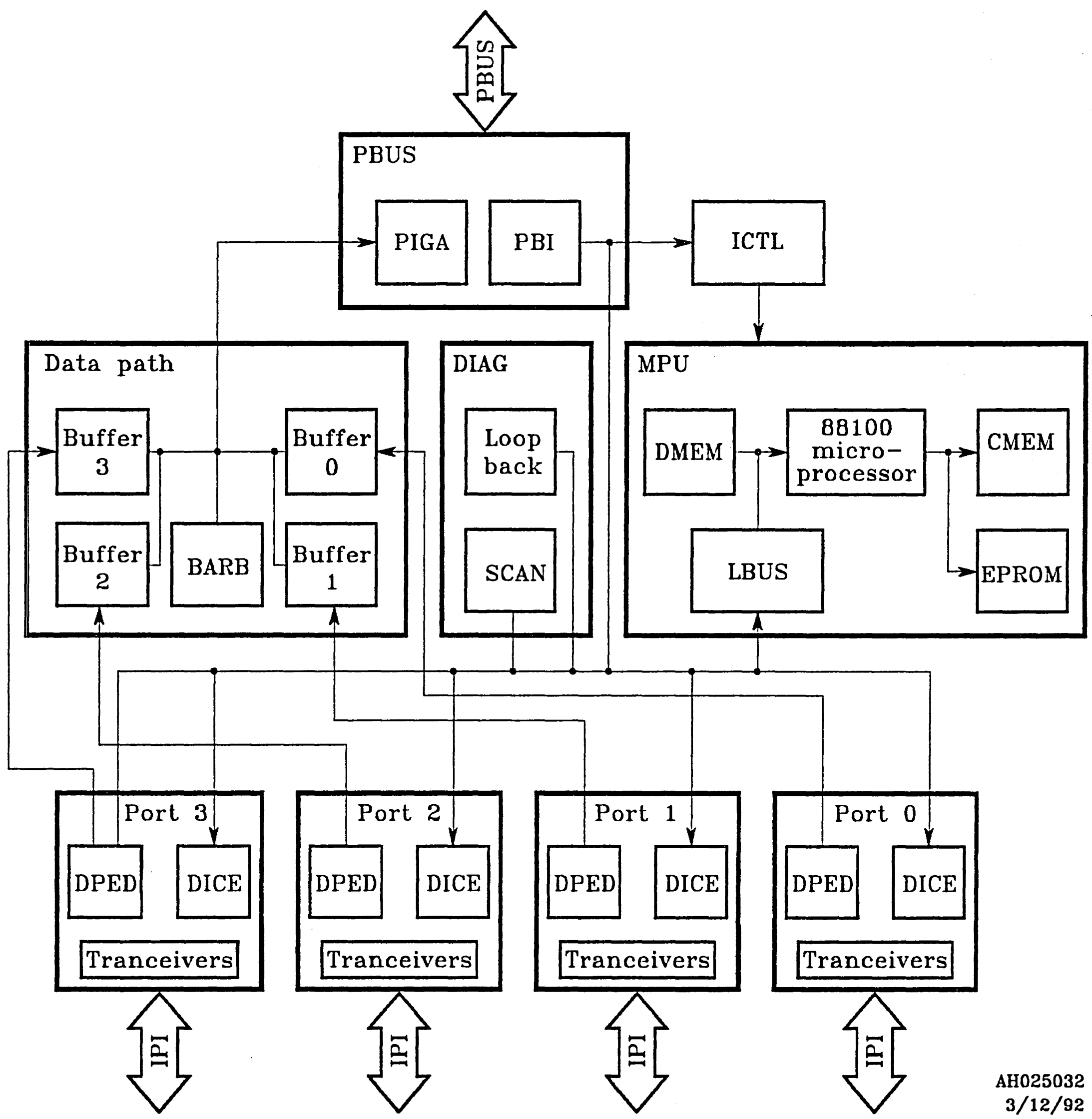
Figure 1–1, IDC/ITC block diagram



1.2 Logical organization

Since many IDC/ITC operations are distributed among multiple hardware units, the IDC/ITC is organized as several logical units and subsystems according to functionality. Figure 1–2 shows a functional block diagram of the programming and data flow through the IDC/ITC discussed in the rest of this chapter.

Figure 1–2, IDC/ITC functional block diagram



1.2.1 Data path

The primary function of the IDC/ITC is to transfer data between external storage devices and main memory. The IDC/ITC 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. All elements of the IDC/ITC 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/ITC 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/ITC 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/ITC functions as the physical interface to system level units:

- Central processing unit (CPU)
- Service processor unit (SPU)
- Peripheral interface adapter (PIA)
- Memory

It is also 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:

- A PBUS sequencer
- A micromachine for data buffer management and address translation
- 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/ITC 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 line clock 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/ITC. MPU software, comprising a realtime kernel and device driver(s), governs all IDC/ITC operations. The MPU communicates with other elements of the IDC/ITC 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/ITC 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 PBI logic maps MPU LBUS reads and writes to physical main memory with a 32-by-1,024 RAM look-up table used to generate PBUS headers. The mapping RAM is initialized at boot time.

MPU software via the LBUS initiates, configures, controls, and monitors IDC/ITC operations. Control registers in the IDC/ITC gate arrays and board level hardware logic are loaded with parameters that are specific to the application and individual operations. Status pertaining to IDC/ITC 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 interrupts to service main memory queues, schedule processes, and detect external events such as operation completion and errors. The 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/ITC 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/ITC contains four independent I/O ports; 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 IDC/ITC supports the IPI level 3 command set for magnetic disk and tape drives.

The MPU code initiates operations with peripherals by programming a set of registers in the DICE and DPED gate arrays. Op codes, 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/ITC 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/ITC software and the attached external devices through DICE registers. The DPED gate array passes data between the IDC/ITC 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/ITC hardware incorporates several systems and logic elements to facilitate testing and fault isolation. All storage registers in the IDC/ITC gate arrays and selected board level registers are scannable from the 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/ITC hardware to be tested at a level that would be difficult or destructive, or both, with active devices attached.

The IDC/ITC clocking requirements are relatively complex. The MPU subsystem generates its own free-running 10-MHz clock from an onboard oscillator and control logic. The remainder of the IDC/ITC hardware uses a 10-MHz gated clock derived from the 20-MHz PBUS backplane clock and MPHASE and DMODE signals. The IDC/ITC 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/ITC register map

The register map for the integrated disk/tape channel is listed in Table 1–1.

Table 1–1, IDC/ITC 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 is continued on the next page.

Table 1–1, IDC/ITC 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 look-up 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.

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. Figure 1–3 illustrates the format of the map.

Figure 1–3, 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. Table 1–2 defines these function codes.

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:

- Address translation
- 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 micromachine 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 microcontrol 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 used to shift and mask data coming out the two read ports on the register file. A 32-bit arithmetic logic unit (ALU) is included in the address translation data path; the ALU is capable of addition and performing logical AND and exclusive NOR operations. 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 micromachine. The microcontrol store is internal to the gate array and is loaded by the MPU at boot time. The gate array is not used by the MPU 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 micromachine, the microcontrol store, and the register file.

1.4.1.1 Micromachine

The micromachine contains the necessary program flow control and ALU logic to carry out the address translation and buffer management algorithm in the microcontrol store. It is enabled by a run bit in the gate array control register. Signals from the IPI interface requesting data movement trigger the micromachine 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 micromachine 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 micromachine uses information from the register file to perform the address translation and obtain the physical address for the transfer. Then the micromachine initiates the actual transfer of data between the buffer and main memory.

1.4.1.2 Microcontrol store

The microcontrol store is loaded by the MPU at boot time with a microprogram provided by the SPU. The control store is 32 bits by 128 microinstructions and occupies one quadrant of the gate array. The PIGA microcode uses various parameters contain The entire control store occupies 1,024 bytes of MPU address space at address 0x00FF7200. The microcontrol store is word read/write addressable.

1.4.1.3 Register file

The register file is a 36-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. Table 1–3 defines the map of the register file.

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. Table 1–4 defines the LBUS addresses for these sets of registers.

Table 1–4, Register set LBUS addresses

LBUS address	Description
0x00FF70e0	Port 3 register set 1
0x00FF70C0	Port 3 register set 0
0x00FF70a0	Port 2 register set 1
0x00FF7080	Port 2 register set 0
0x00FF7060	Port 1 register set 1
0x00FF7040	Port 1 register set 0
0x00FF7020	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 transfer and a pending transfer for each port. At the completion of a transfer, the sequencer automatically switches sets. Table 1–5 lists the addresses and sizes of the registers in a transfer register set.

Table 1–5, PIGA register addresses

Description		Address
Constants		0x00FF7180–0x00FF71FF
Channel 3		0x00FF7160–0x00FF717F
Channel 2		0x00FF7140–0x00FF715F
Channel 1		0x00FF7120–0x00FF713F
Channel 0		0x00FF7100–0x00FF711F
BFSIZ	Data buffer block size	0x00FF711C
BUFADR	Data buffer starting address	0x00FF7118
XBC2	Length of second transaction	0x00FF7114
XBC	Transaction length	0x00FF7110
OLDLA	Logical address of last transaction	0x00FF710C
L2PTE	Level 2 PTE value	0x00FF7108
L1PTE	Level 1 PTE value	0x00FF7104
SDR	SDR value	0x00FF7100
Channel 3, set 1		0x00FF70e0–0x00FF70FF
Channel 3, set 0		0x00FF70c0–0x00FF70Df
Channel 2, set 1		0x00FF70a0–0x00FF70Bf
Channel 2, set 0		0x00FF7080–0x00FF709F
Channel 1, set 1		0x00FF7060–0x00FF707F
Channel 1, set 0		0x00FF7040–0x00FF705F
Channel 0, set 1		0x00FF7020–0x00FF703F
Channel 0, set 0		0x00FF7000–0x00FF701F
T0	Temporary storage	0x00FF701C
BLNDR	Value for PBI blender logical	0x00FF7018
HDROP	Transaction type	0x00FF7014
SDR7	Physical address of SDR 7	0x00FF7010
SDR0	Physical address of SDR 0	0x00FF700C
COUNT	Bytes remaining to be transferred	0x00FF7008
LADDR	Logical address of current transaction	0x00FF7004
CSR	Control and status register	0x00FF7000

The remaining 32 registers in the register file are used to hold constants and masks needed by the microprogram. These registers are loaded by the MPU at boot time. Table 1–6 lists the constant register assignments.

Table 1–6, Constant registers

Address	Name	Description	Constant
0xFF71C4	hex8	Hex 8	0x8
0xFF71C0	errflag	CSR error flag bit	0x100
0xFF71BC	hex7	Hex 7	7
0xFF71B8	pgsiz	Page size	4096
0xFF71B4	tpb	Two page buffer	4
0xFF71B0	smsk	SDR mask	0E0000000
0xFF71AC	sl1msk	SDR level 1 mask	0FFC00000
0xFF71A8	sdrvmsk	SDR validity mask	10
0xFF71A4	sdrmsk	SDR mask	0FFFFFE00
0xFF71A0	read4	PBUS header operation	0004F9FF
0xFF719C	pomsk	Page offset mask	00000FFF
0xFF7198	pfmsk	Page frame mask	0FFFFF000
0xFF7194	l1l2vmsk	L1L2 validity mask	51
0xFF7190	l1l2msk	L1L2 mask	0FFFFF000H
0xFF718C	csrxv	Control and status translation valid	2
0xFF7188	csrdone	Control and status completion	1
0xFF7184	bfsiz	Buffer size	512
0xFF7180	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 micromachine 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.

Parity is checked on three of the four read ports on the RAMs. It is checked on the S port of the control store but not on the D port since the port is not used. The S and D ports of the register file are checked. One signal from the PIGA indicates a parity error has been detected. All of the parity checker outputs are combined using a logical OR operation 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

This section lists the formats of the PIGA gate array internal registers.

Table 1–7 defines the format of the parity status register (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] RMX <2..0> signals should be taken to 0b100 in order to read the parity error register.

Table 1–8 defines the format of the control store register.

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

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 in this guide. Table 1–9 defines the format of the header operation register.

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

Table 1–10 defines the format of the control and status register.

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 ITC. 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/ITC 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/ITC 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 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. Overlapping instruction accesses in this manner increases the access rate.

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:

- Instruction fetch
- Local data bus read
- 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: instruction fetch and LBUS 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 an *LBUS RAM access* signal. This signal, along with an *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 the LBUS receives this acknowledgement, the LBUS state machine deasserts the *LBUS access* signal, allowing 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 SRAM, known as the DSTORE, is accessible through the data memory (DMEM) logic. The 88100 also uses special-purpose registers on the IDC/ITC 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 data memory port. Then the DMEM logic determines which part of the IDC/ITC 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 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 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 delay 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:

- DSTORE SRAM access
- 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 completed, whether the access was successful. The data reply generator then reports this access status to the 88100 with a *reply* signal.

The LBUS logic provides a path from the 88100 data memory port to the IDC/ITC's local data bus. The LBUS logic allows the 88100 to access the MPU's CSTORE SRAM and EPROM registers which control the IDC/ITC'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)

This section describes the buffer arbitration (BARB) logic, which coordinates accesses to the four I/O port data buffers.

1.6.1 Buffer access

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.

1.6.2 Buffer arbitration registers

The formats of the registers in the buffer arbitration (BARB) logic are listed in this section. Table 1–11 defines the format of the arbiter counter register.

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>

Table 1–12 defines the format of the arbiter port control register.

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 block	0 = unlimited

1.7 Diagnostic logic (DIAG)

This section describes the diagnostic (DIAG) logic, which performs a variety of functions associated with IDC/ITC system operations.

1.7.1 Overview

The DIAG logic contains the IDC/ITC 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/ITC 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/ITC COP chip resides in the DIAG block at the least significant end of the scan ring. The COP chip contains IDC/ITC identification and revision information that is read by the SPU during system configuration. An LBUS register that drives LEDs on the foreplane edge of the IDC/ITC is written by MPU software to indicate IDC/ITC status in realtime. 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/ITC software and the SPU. The SCR register allows the SPU to indirectly access ITC 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/ITC kernel software.

1.7.2 Diagnostic registers

The formats of the registers in the diagnostic (DIAG) logic are listed in this section.

Table 1–13 defines the format of the fault source register.

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.

Table 1–14 defines the format of the read header low register.

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>

Table 1–15 defines the format of the read header hi register.

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>

Table 1–16 defines the format of the slot ID register.

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>

Table 1–17 defines the format of the scan communication register.

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>

Table 1–18 defines the format of the diagnostic control register.

Table 1–18, Diagnostic control register format (DCR)

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

Table 1–19 defines the format of the test result register.

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. The interrupt controller prioritizes interrupts as they arrive from various sources within the IDC/ITC and sends them to the IDC/ITC. 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/ITC 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 that asserted them. Board interrupts (ports 3–0) are acknowledged by writing to the interrupt acknowledge register (IAREG) with the appropriate bit(s) set.

Table 1–20 defines the format of the interrupt pending register.

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

Table 1–21 defines the format of the interrupt send register.

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>

Table 1–22 defines the format of the interrupt status register.

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

Table 1–23 defines the format of the interrupt enable register.

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

Table 1–24 defines the format of the interrupt acknowledge register.

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

Table 1–25 defines the format of the interrupt control register.

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>

Table 1–26 defines the format of the interrupt vector register.

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/ITC 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. The DICE gate array records command completion, exception conditions, the occurrence of external events, and reports them to the MPU via status registers and interrupts.

The DICE gate array is independent of the IPI logical interface (command set). The gate array does not generate or interpret IPI control octets; operations do not depend on the content of information transfers. As a result, the IDC/ITC 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 for a listing of the DICE internal registers.

The MPU reads the DICE internal registers and writes them asynchronously. 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/ITC 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 op code field that identifies one of eight commands or IPI operations to be performed. The CREG also contains a 3-bit op code modifier field that is presented to DICE output signal pins while CREG execution is active. These signals control external hardware. Internal DICE operations are affected by the modifier bits. The op codes and the associated functions are described in the ANSI specification X3.129-1986 IPI-1.

Table 1–27 lists the IPI control octets for Bus A and Bus B.

Table 1–27, IPI control octets

Op code	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

`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 time-out period expires. The op code 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 time-out error value.

`Selection` (5) This command initiates a selection sequence that establishes a virtual circuit connection between the IDC/ITC 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. MPU software verifies the successful execution of a selection sequence (indicated by the bit-significant address of the selected slave, and no other, set in the bus B octet) 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.

1.9.2.2 Command execution

Once programmed, the DICE gate array executes 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 instance 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, they typically are not chained because the DICE gate array can only report the outcome (the bus B octet) of these operations individually.

`Output Bus Exchange` and `Input Bus Exchange` commands communicate logical level command and response information and to transfer data. Each transfer command 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 up counters. 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, 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 bytes <26..24>. Maximum counter values are specified by the 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, while 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. Table 1–28 defines the command status values.

Table 1–28, Command status codes

Code	Definition
0	Idle state
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 time-out

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. Table 1–29 defines the transfer status values.

Table 1–29, Transfer status codes

Code	Definition
0	Idle state
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 types of interrupts:

- Command completion
- Exception conditions
- 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 an 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 effect.

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 nonactive interrupt bits are ignored by the peripheral interface during interrupt acknowledgement. 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 DICE internal registers

Table 1–30 lists the DICE internal registers, associated addresses, and access modes. The following sections list the format and field definitions of these internal registers.

Table 1–30, DICE internal registers

Address	Name	Description	Access
31	NA	Reserved	NA
30	NA	Reserved	NA
29	NA	Reserved	NA
28	NA	Reserved	NA
27	HREG3	Header register 3	read/write
26	HREG2	Header register 2	read/write
25	HREG1	Header register 1	read/write
24	HREG0	Header register 0	read/write
23	XREG7	Transfer register 7	read/write
22	XREG6	Transfer register 6	read/write
21	XREG5	Transfer register 5	read/write
20	XREG4	Transfer register 4	read/write
19	XREG3	Transfer register 3	read/write
18	XREG2	Transfer register 2	read/write
17	XREG1	Transfer register 1	read/write
16	XREG0	Transfer register 0	read/write
15	FCREG	Field configuration register	read/write
14	HCREG	Header counter control register	read/write
13	ECREG	Error check register	read
12	TCREG	Transfer count register	read/write
11	EXREG	Execution register	read
10	ISREG	Interrupt status register	read/write
09	OPREG	Operational parameter register	read/write
08	NA	Reserved	NA
07	CREG7	Command register 7	read/write
06	CREG6	Command register 6	read/write
05	CREG5	Command register 5	read/write
04	CREG4	Command register 4	read/write
03	CREG3	Command register 3	read/write
02	CREG2	Command register 2	read/write
01	CREG1	Command register 1	read/write
00	CREG0	Command register 0	read/write

1.9.3.1 Data registers

This section describes the functions of the individual DICE data registers.

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.

1.9.3.2 Status registers

The DICE status registers are used to record and report operational conditions and completion status to the MPU. This section describes the function and the format of the individual DICE status registers.

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. Table 1–31 defines the format of the error check register.

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. Table 1–32 defines the format of the execution register.

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 (Attn)
18	External interrupt 1 (PIGA complete)
17	External interrupt 2 (PIGA error)
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. Table 1–33 defines the format of the interrupt status register.

Table 1–33, 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	Reserved
14	Reserved
13	Reserved
12	Reserved
11	Command exception interrupt
10	External interrupt 0 (Attn)
09	External interrupt 1 (PIGA complete)
08	External interrupt 2 (PIGA error)
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. This section describes the function and the format of the individual DICE control registers.

Field control register (FCREG)—The FCREG enables and configures functions associated with IPI information transfers. The FCREG enables exception conditions and error checking (CRC and HREG/XREG comparisons) and contains start pointers for XREG and HREG transfers. Table 1–34 defines the format of the field configuration register.

Table 1–34, Field configuration register format (FCREG)

Bit	Definition
31	Transfer 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. Table 1–35 defines the format of the header counter control register.

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). Table 1–36 defines the format of the transfer count register.

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. It also associates interrupt levels with exception conditions and external events and contains the time-out value used in IPI control sequences. Table 1–37 defines the format of the operational parameter register.

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 op code, 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]. Table 1–38 defines the format of a command register.

Table 1–38, Command register format (CREG)

Bit	Definition
31	Op code bit <2>
30	Op code bit <1>
29	Op code bit <0>
28	Op code modifier bit <2> (inhibit buffer block count)
27	Op code modifier bit <1> (reserved)
26	Op code 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:

- **Input/output transfers**—Transfers of data between the data buffer and the IPI port (between the buffer bus and bus A and bus B)
- **Local bus accesses**—Transfers between internal DPED registers and the local bus

The following sections describe these functions in detail.

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 ITC. 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. 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 is then 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, and if there was ECC data, then the operation is complete. If there was no ECC data, the data remaining in the serial or parallel registers is transferred to the data buffer.

An output operation consists 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 fetches the first longword of data, the data in the serial registers is shifted onto bus A and bus B. After each halfword of a longword has been registered in SERA, a new longword is loaded into the serial registers. These operations continue 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 should be set for an input or output operation:

- Transfer size
- Block size
- ECC interleave
- Block reset

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:

- Error correction code errors
- Length errors
- Overrun/underrun errors
- 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 synchronized input in an amount different from what is expected during the time that the DPED gate array is ready to receive synchronized input. The expected number 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, an overrun or underrun error is generated if a buffer grant is not available when the last longword of a block goes out on the buffer bus. 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, an overrun or underrun error is generated if a buffer grant is not available when the second-to-the-last longword in a block is read from the buffer bus.

Parity errors are the result of parity checks on data going to 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. Data is not passed to the PBUS.

1.10.1.4 Local bus accesses

A local bus access is a local read of data from the LBUS or a write of data to the LBUS. 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 to the LBUS on local reads.

When data has been output onto the LBUS or written into an internal register, a register access acknowledgment is generated. If an invalid access is attempted, a local access fault is generated. An invalid access involves attempts to write read-only registers or to access addresses above 0x1000. A local access fault is also generated if there is a parity error on the incoming data from the LBUS.

1.10.2 Data path and error detection internal registers

Table 1–39 lists the DPED registers, associated addresses, access modes, and sizes.

Table 1–39, DPED internal registers

Micro address	Name	Description	IAD	Access	Size
00FFx0A0	BADDR	Buffer address register	1000	read/write	16
00FFx09C	SYNDROME	ECC syndrome register	0111	read	32
00FFx098	XCNT	Transfer count register	0110	read	24
00FFx094	BKCNT	Block count register	0101	read	13
00FFx090	BCREG	Basic control register	0100	read/write	32
00FFx08C	PARC & PARD	Parallel C & D data registers	0011	read	36
00FFx088	PARA & PARB	Parallel A & B data registers	0010	read	36
00FFx084	SERC & SERD	Serial C & D data registers	0001	read	36
00FFx080	SERA & SERB	Serial A & B data registers	0000	read	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 nonzero 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 nonzero 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 nonzero 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 nonzero 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 nonzero 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 nonzero 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 nonzero 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 nonzero data when read is after the premature termination of a read or write operation to an IPI port.

1.11 IDC/ITC LEDs

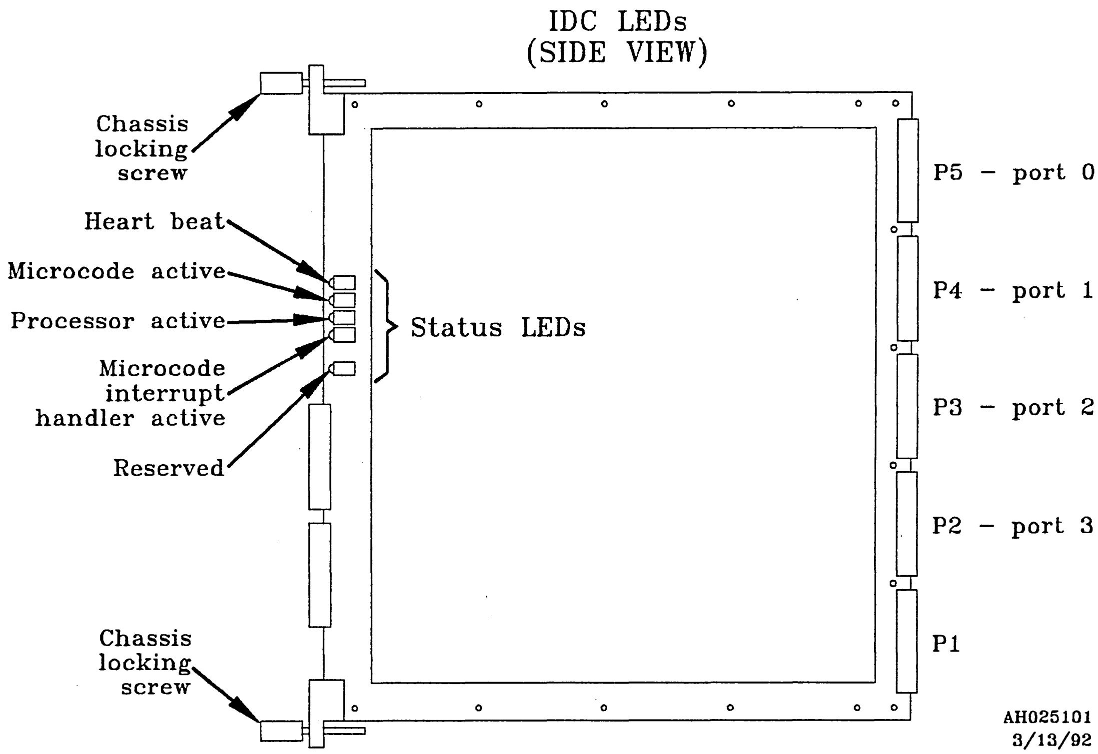
The IDC/ITC LEDs are defined as follows:

- **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. It is turned off when the microcode crashes. However, this 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 and 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 and off when an interrupt is not present.
- **Reserved**—The reserved LED is reserved for future use.

Figure 1–4 shows the IDC/ITC LEDs.

Figure 1–4, IDC/ITC LEDs



1.12 Disk drive front panel indicator and switch descriptions

Table 1–40 describes the disk drive front panel indicators and switches.

Table 1–40, Disk drive front panel indicators and switches

Item	Type	Description
Address	Indicators	Indicates the binary logical address (0–7) of disk drive.
	Switch	Used to set the logical address of a disk drive[1]. Press for 2 to 3 seconds to advance the logical address.
Selected	Indicator	**ON** = Disk drive selected by controller.
		OFF = Disk drive not selected by controller.
Start	Indicator	**ON** = Disk drive is ready.
		OFF = Disk drive is not ready.
		FLASHING = Disk drive is in START or STOP cycle.
	Switch	Used to start or stop a disk drive. Press and release to begin START cycle. Press and release again to begin STOP cycle.
Fault	Indicator	**ON** = Disk drive fault condition exists.
		OFF[2] = Disk 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** = Disk drive write operations disabled.
		OFF = Disk drive write operations enabled.
	Switch[3]	Used to enable or disable disk drive write operations. Press and release to disable disk drive write operations. Press and release again to enable disk 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 power switch on the rear of the disk drive is switched to the OFF position, or when there is a loss of site power.

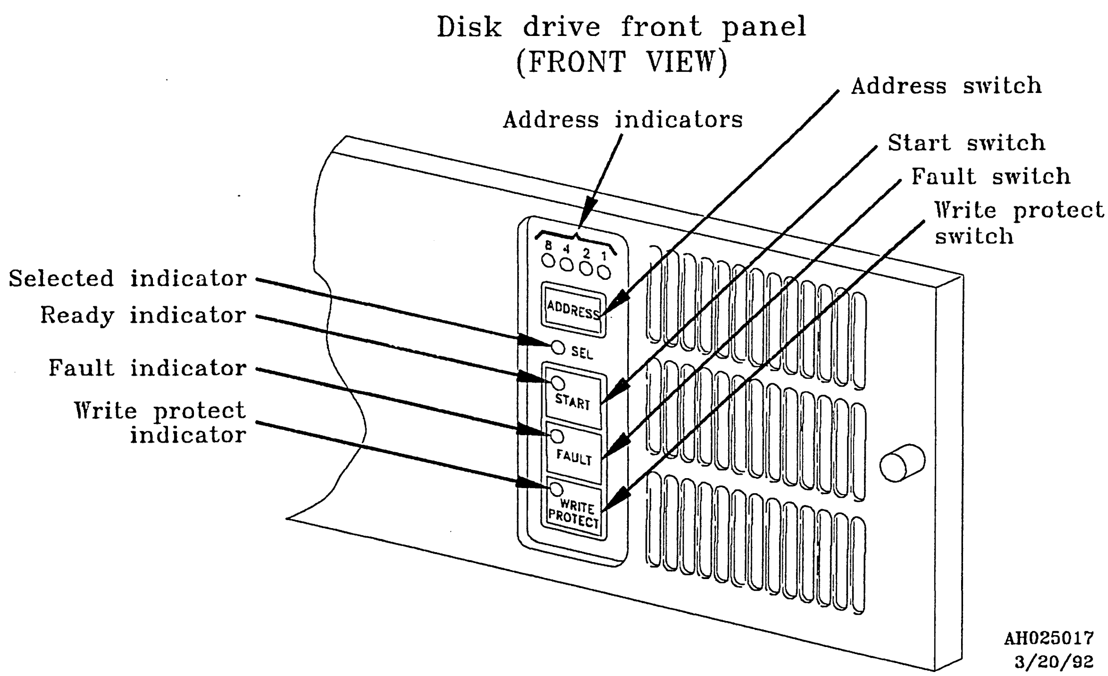
[2] The indicator is turned **OFF** by any of the following, if the fault condition no longer exists:

- By pressing the FAULT switch
- A disk drive start cycle begins
- A Fault Clear command from the controller occurs

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

Figure 1–5 illustrates the disk drive front panel indicators and switches.

Figure 1–5, Disk drive front panel



1.13 Disk drive control board jumper and switch descriptions

Table 1–41 lists the disk drive control board jumpers and switches.

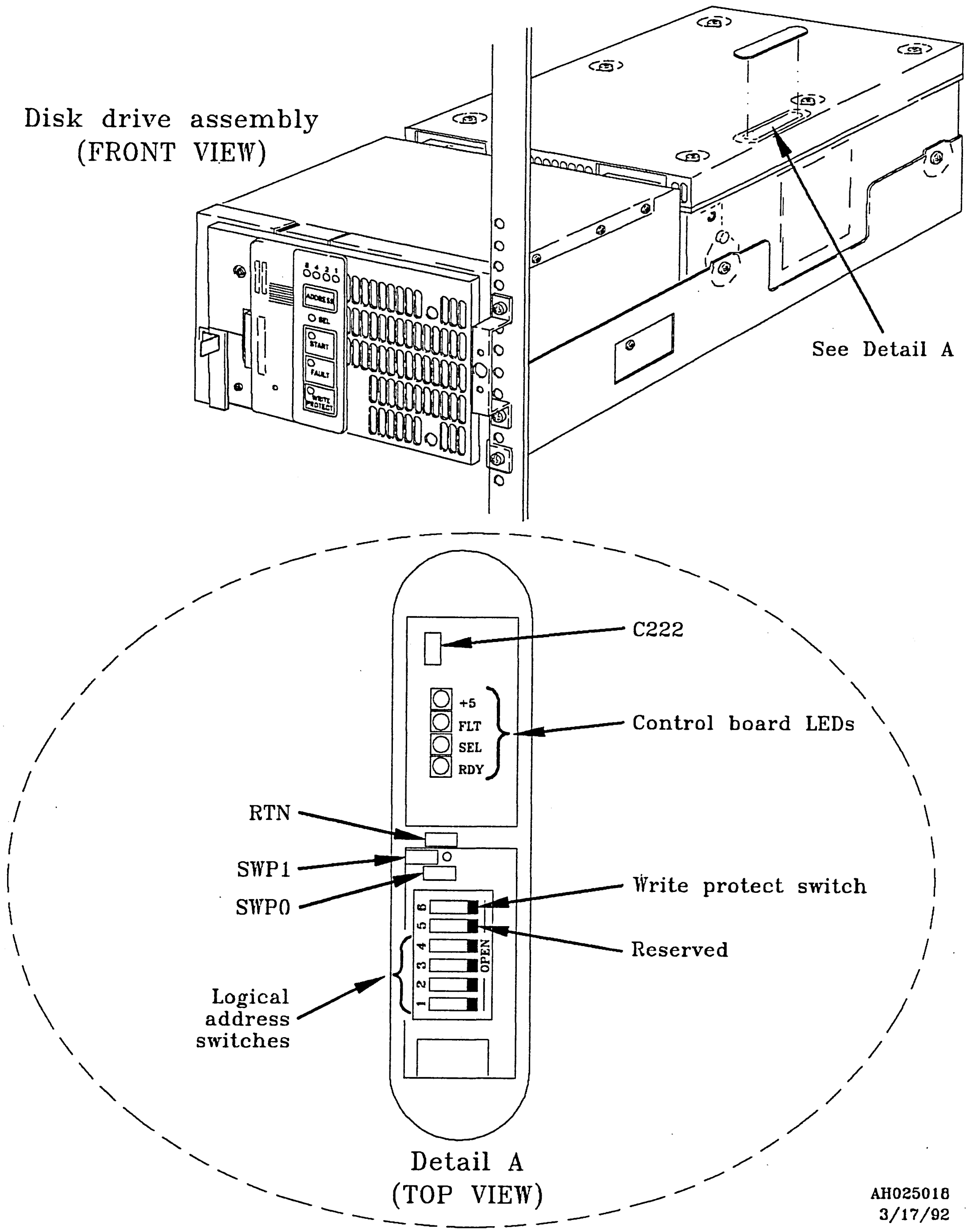
Table 1–41, Disk drive control board jumpers and switches

Item	Type	Setting	Description
RTN	Jumper	ON	Enables option to return heads to original position following sweep segment if disk 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

Figure 1–6 illustrates the disk drive control board jumpers and switches.

Figure 1–6, Disk drive control board



1.14 Disk drive input/output board switch descriptions

Table 1–42 lists the disk drive input/output board switch settings and descriptions.

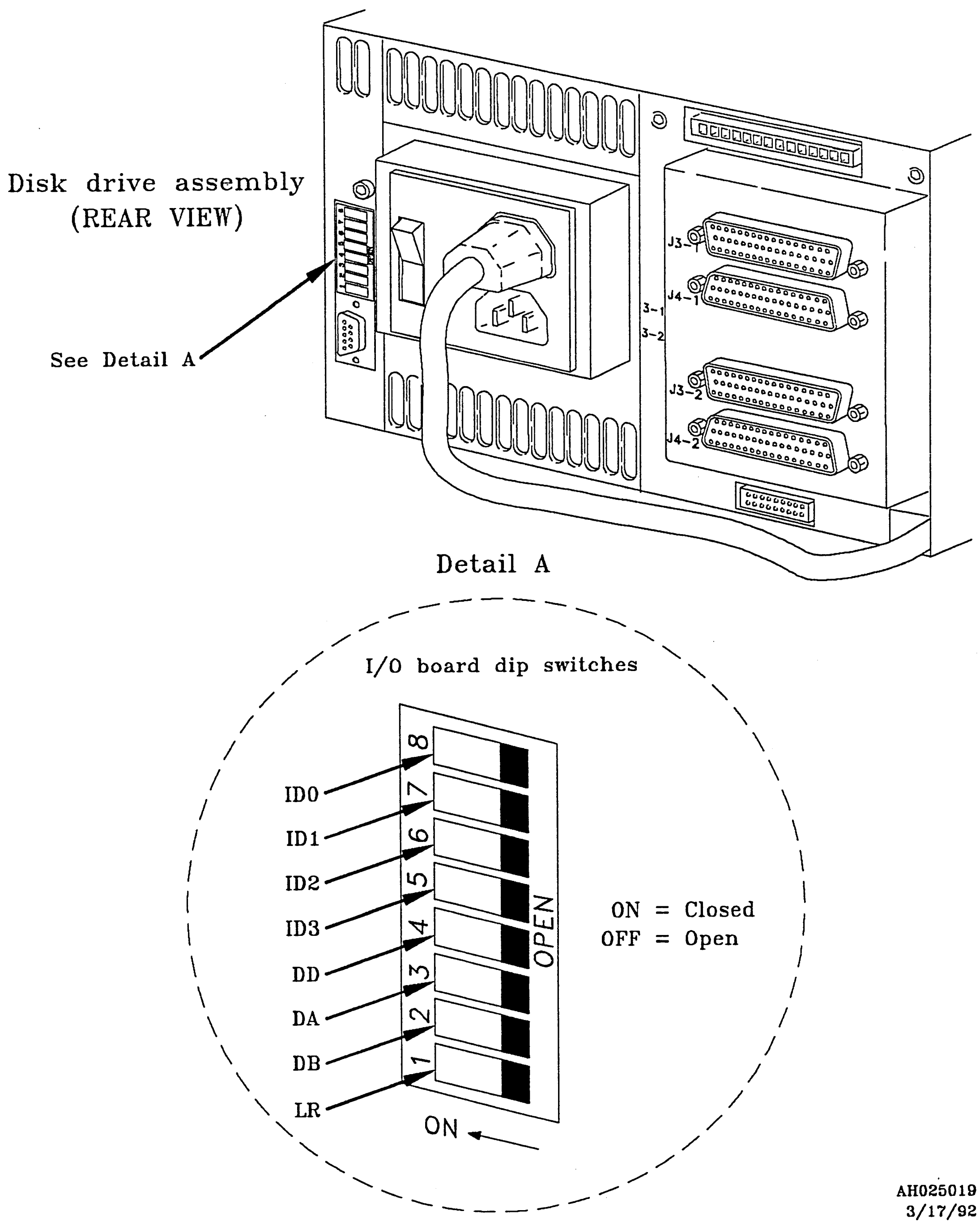
Table 1–42, Disk drive I/O board dip switches

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 disk drive.
	OFF[1]	**Local operation**—Disk spin-up starts when power is applied to the disk drive or when the disk drive front panel START switch is pressed and released.

[1] CONVEX switch setting

Figure 1–7 illustrates the disk drive input/output board switches.

Figure 1–7, Disk drive input/output board



1.15 Power controller description

The 48-amp power controller supplies 200 V–240 V 50–60 Hz single-phase power. The power controller delivers filtered, fused power to single phase peripherals, cabinet blowers, and cabinet sensors. It alsoprovides 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. Table 1–43 describes the main circuit breaker.

Table 1–43, Main circuit breaker

Contactor	Breaker	Rating	Trip (10 second)	Connectors
NA	CB1	60 A	NA	**J15** Main Input

- **Output circuit breakers**—These 2-pole circuit breakers are designed to disconnect power to the power outlets. Table 1–44 describes the output circuit breakers.

Table 1–44, Output circuit breakers

Contactor	Breaker	Rating	Trip (10 second)	Connectors
K2	CB2	30 A	75	**J14** Twist-lock
K1	CB3	30 A	75	**J9** Twist-lock
K2	CB4	20 A	50	**J12** & **J13** Duplex
K1	CB5	20 A	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. Table 1–45 describes the transformer circuit breaker.

Table 1–45, Transformer circuit breaker

Contactor	Breaker	Rating	Trip (10 second)	Connectors
NA	CB6	2 A	5	Transformer

- **Local/Remote switch**—A 3-position switch that provides control of two contactors. The three positions are defined as follows:
  - **OFF**—24 V 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.
  - **LOCAL**—Contactor K1 is energized with 24 V instantly, and contactor K2 is energized after delay T1 as described in Table 1–46. 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 V supplied, 2-pole contactor K1[2] closes
T0 + *T1*[1] s	2-pole contactor K2[2] closes
T1 + *T2*[1] s	**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**—Indicates whether input power is being delivered to the load side of the main circuit breaker. It is connected to the load side of the main circuit breaker and is ON while power is being delivered.
- **Contactor indicators**—Indicates whether the contactor is closed. Indicators are connected to the supply of each contactor; 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. Table 1–47 lists the contactor indicators.

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. Table 1–48 lists the shutdown input indicators.

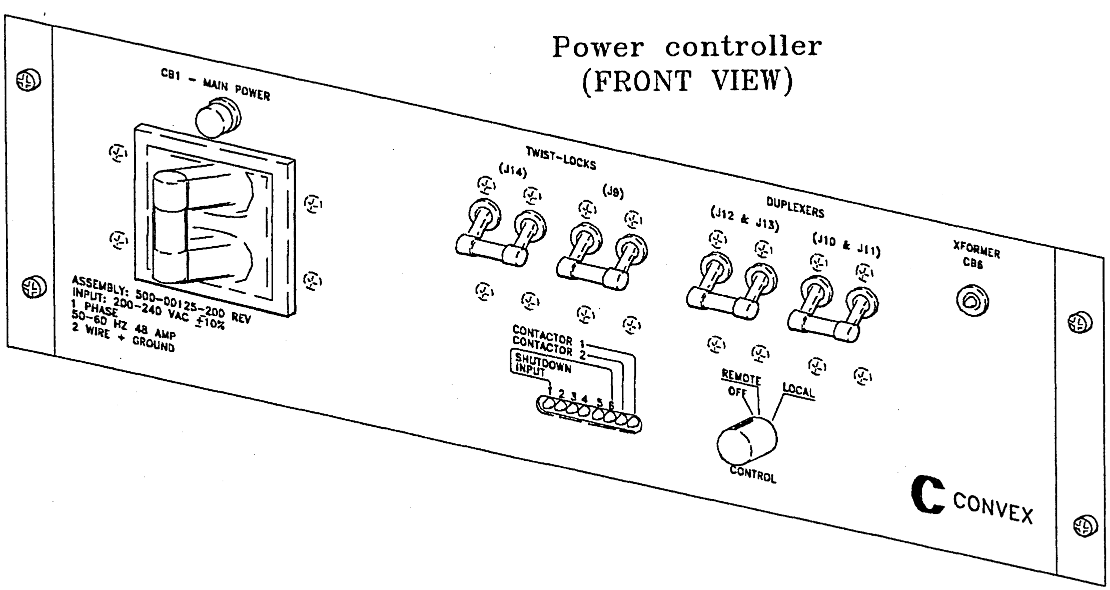
Table 1–48, Shutdown input indicators

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[1]

[1] All unused input connectors must have pin 1 and pin 3 jumpered to override the open cable detection.

Figure 1–8 illustrates the power controller front panel switches and indicators.

Figure 1–8, Power controller front panel switches and indicators



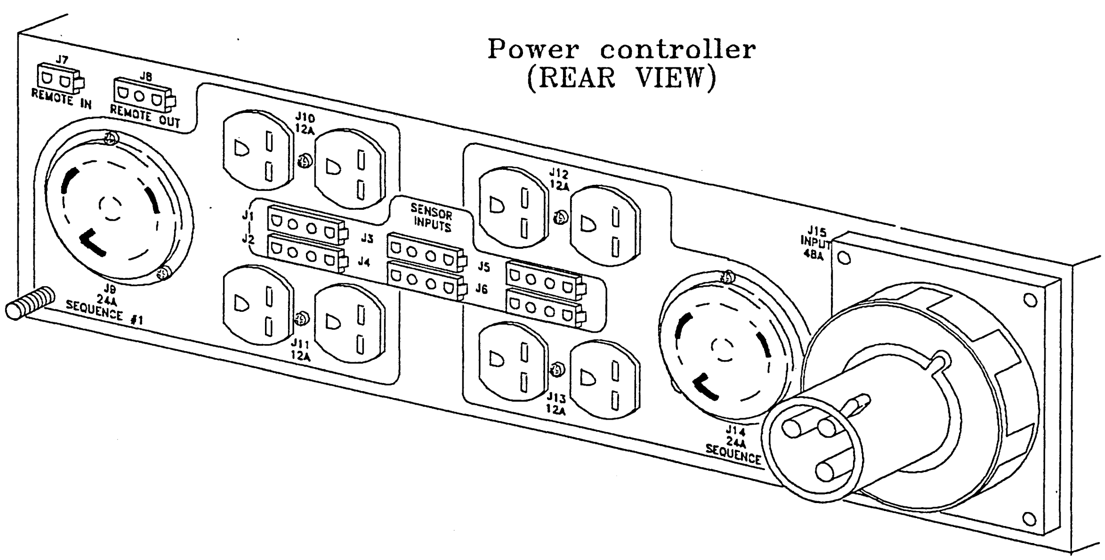
AH025034
3/18/92

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 power cord connection

Figure 1–9 illustrates the power controller rear panel connections.

Figure 1–9, Power controller rear panel connections



AH025035
3/17/92

1.16 IDC/ITC board specifications

Table 1–49 lists the specifications for the IDC/ITC:

Table 1–49, IDC/ITC specifications

Parameter	Value or 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/hour (8 °C/hour)
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/disk drive specifications

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

1.17.1 Dimensions and weights

Table 1–50 lists the physical dimensions of the peripheral equipment (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-101 Peripheral cabinet	355.0[1] (161.4)	25.1 (63.7)	39.3 (99.8)	62.3 (158.1)
EXP-102 Peripheral cabinet	355.0[1] (161.4)	25.1 (63.7)	39.3 (99.8)	62.3 (158.1)
EXP-105 Peripheral cabinet	382.0[1] (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)	9.5[4] (24.1)	28.0 (71.1)	5.25[5] (13.3)
DKD-502 Disk drive	60.0[2]/46.0[3] (27.2/20.9)	9.5[4] (24.1)	28.0 (71.1)	5.25[5] (13.3)
DKD-503 Disk drive	60.0[2]/46.0[3] (27.2/20.9)	19.0[4] (48.3)	23.3 (59.2)	5.25[5] (13.3)
DKD-504 Disk drive	60.0[2]/46.0[3] (27.2/20.9)	19.0[4] (48.3)	23.3 (59.2)	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

Tables 1-51 and 1-52 list the North American and international electrical specifications for the peripheral cabinet and disk drives.

Table 1–51, Peripheral equipment North American specifications

Equipment	Voltage ±10%	Frequency ±1 Hz	Phase no.	Current (amperes)
EXP-101 Peripheral cabinet	120	60	1	24.0[1]
EXP-102 Peripheral cabinet	120	60	1	24.0[1]
EXP-105 Peripheral cabinet[2]	200–240	50–60	1	48.0[3]
DKD-501 Disk drive	120	60	1	2.5[4]
DKD-502 Disk drive	120	60	1	2.5[4]
DKD-503 Disk drive	120	50-60	1	2.0-2.81[4]
DKD-504 Disk drive	120	50-60	1	2.0-2.81[4]

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

Table 1–52, Peripheral equipment international specifications

Equipment	Voltage ± 10%	Frequency ± 1 Hz	Phase no.	Current (amperes)
EXP-101 Peripheral cabinet	220–240	50–60	1	24.0[1]
EXP-102 Peripheral cabinet	220–240	50–60	1	24.0[1]
EXP-105 Peripheral cabinet[2]	220–240	50–60	1	48.0[3]
DKD-501 Disk Drive	120–220	50–60	1	1.5–2.5[4]
DKD-502 Disk Drive	120–220	50–60	1	1.5–2.5[4]
DKD-503 Disk Drive	208/240	50–60	1	1.5–1.35–1.65[4]
DKD-504 Disk Drive	208/240	50–60	1	1.5–1.35–1.65[4]

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

1.17.3 Power cord and cable specifications

The customer needs a HUBBELL Part Number 360C6W (CONVEX Part Number 304-000036-001) cable connector to cable the power controller to the site power source.

The international cord must be hard wired into the site power source.

Tables 1–53 and 1-54 list the specifications for output and control cables.

Table 1–53, Output cable specifications

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

Table 1–54, Control cable specifications

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

1.17.4 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 at the air intake 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 air intake of the cabinet.

Table 1–55 lists the temperature and humidity ranges for CONVEX equipment.

Table 1–55, 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

1.17.5 Dissipation and air conditioning requirements

Table 1–56 lists power dissipation and air conditioning requirements for various CONVEX IDC/ITC equipment.

Table 1–56, Dissipation and air conditioning requirements

Equipment	Watts	Btu/hour	Kcals/hour	Tons of refrigeration
EXP-101 Peripheral cabinet	1,00	3,413	860	0.28
EXP-102 Peripheral cabinet	1,00	3,413	860	0.28
EXP-105 Peripheral cabinet	3,00	10,239	2,580	0.85
IDC/ITC 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
DKD-503 Disk drive	147	489	126	0.04
DKD-504 Disk drive	147	560	126	0.04

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2
Unpacking and installation

2.1 Overview

This chapter describes procedures to install the IDC/ITC channel control unit in a VMEbus chassis, including cable installation. Procedures are included for:

- Inspection and damage claims
- Electrostatic discharge (ESD) prevention
- Peripheral cabinet unpacking and installation
- CCU installation

2.2 Inspection and damage claims

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 and contact the carrier immediately. Damage claims should be completed by the customer and given to the shipping representative. Claim forms are normally obtained from the shipping representative.

> **NOTE**
>
> Save all packing material until after operational checkout of the equipment. This allows you to return the equipment if there are problems.

2.3 Electrostatic discharge damage

Because of the electrostatically-sensitive devices used within the circuitry, the IDC/ITC and disk drives are sensitive to static electricity. They can be damaged by electrostatic discharges (ESD) caused during maintenance procedures such as installation. Use proper care when handling or performing maintenance on or around the equipment.

Static charges take place when various objects are separated or rubbed together, often creating a high voltage level. The voltage charge is primarily determined by the following factors:

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

Table 2–1 lists charge levels based on various activities and humidity levels.

Table 2–1, Static charge levels and relative humidity

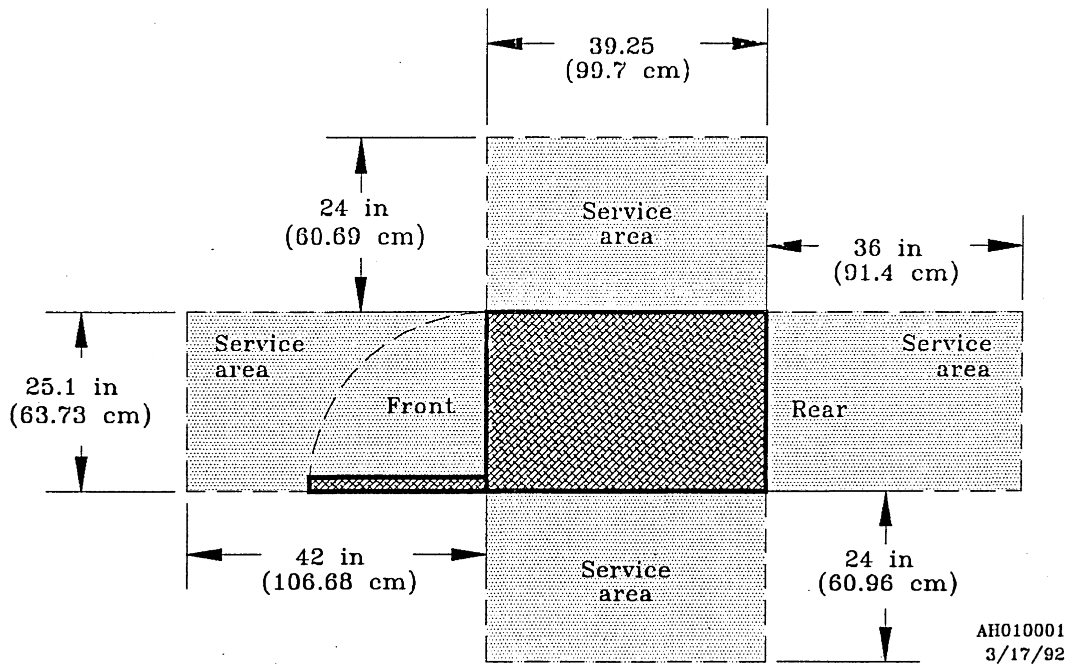
Personnel activity[1]	Humidity[2] & charge levels (volts[3])			
	26%	32%	40%	50%
Person walking across linoleum floor	6,150 V	5,750 V	4,625 V	3,700 V
Person walking across carpet	18,450 V	17,250 V	13,875 V	11,100 V
Person getting up from a plastic chair	24,600 V	23,000 V	18,500 V	14,800 V

[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.

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. Figure 2–1 is the template for a peripheral cabinet.

Figure 2–1, Peripheral cabinet template



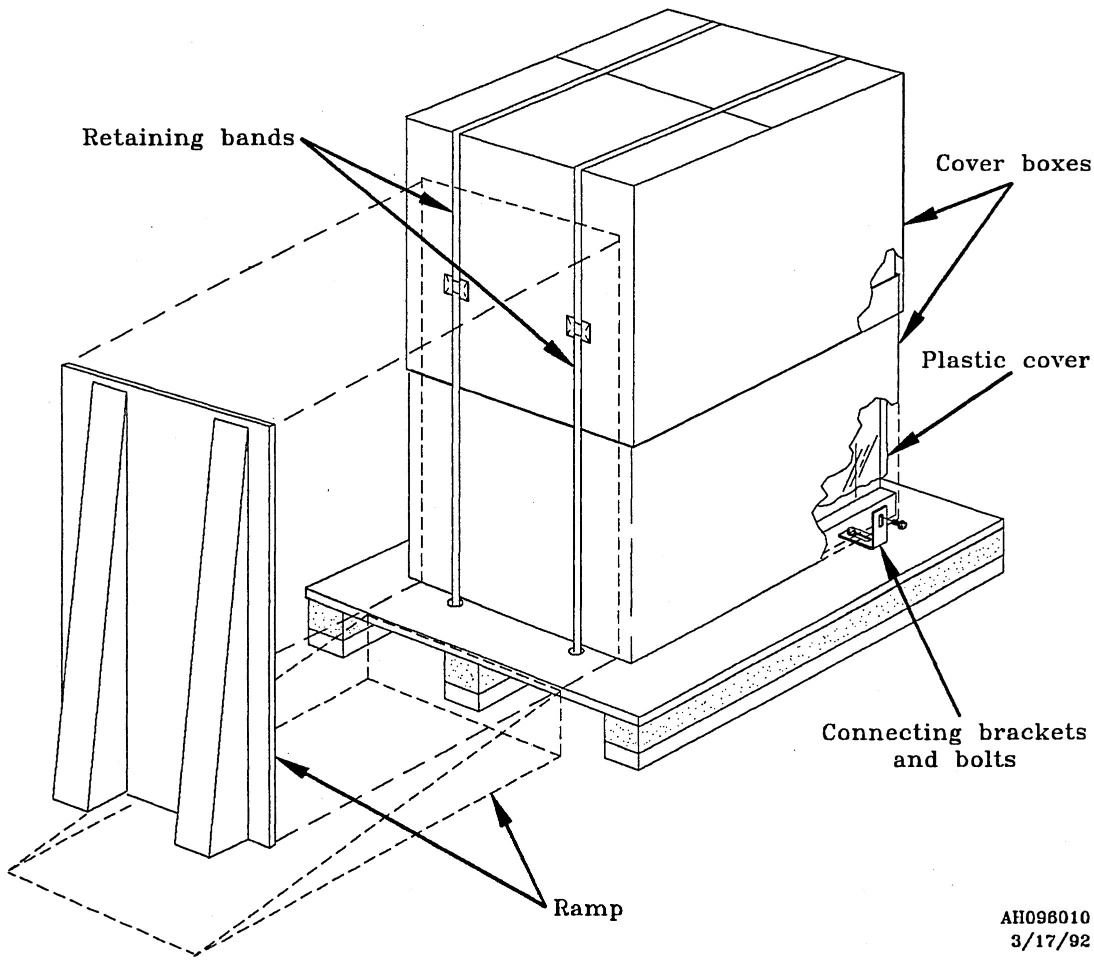
Refer to the applicable processor site preparation guide (listed in the "Preface") for additional site preparation information.

2.5 Peripheral cabinet unpacking and installation

This section describes how to unpack and install the peripheral cabinet. Figure 2–2 shows a cabinet with its packaging.

> **NOTE**
>
> One ramp is shipped with the peripheral cabinet.

Figure 2–2, Cabinet packaging



2.5.1 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.

2.5.2 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 2–3 illustrates a cabinet mounted on a pallet with a ramp and auxiliary ramp.

Figure 2–3, Cabinet pallet, ramp, and auxiliary ramp

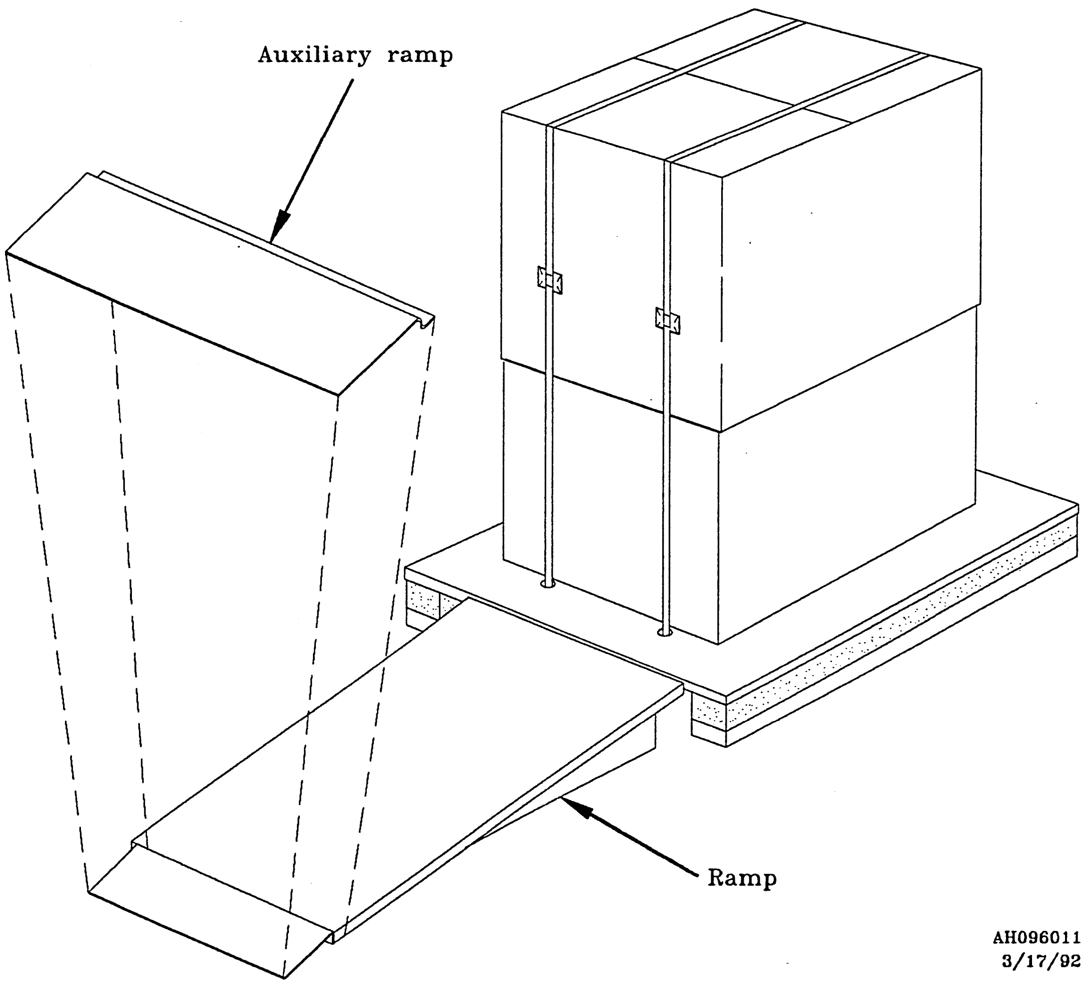
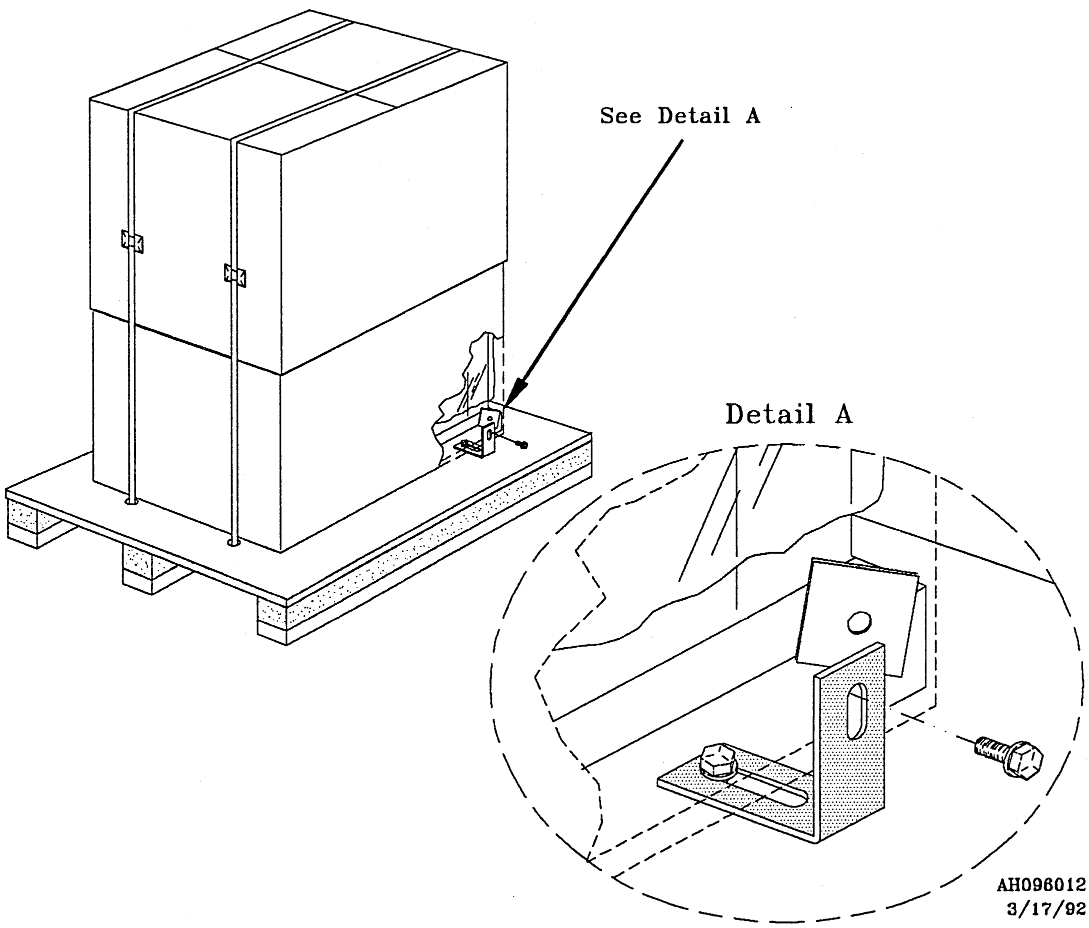


Figure 2–4 illustrates a peripheral cabinet mounted on a pallet with a pallet bracket and spacers.

Figure 2–4, 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 procedure 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 power cord of the cabinet.
9. Roll the cabinet into position.
10. Verify that the 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.

2.5.3 Unpacking 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 procedure to remove the accessories from the pallet:

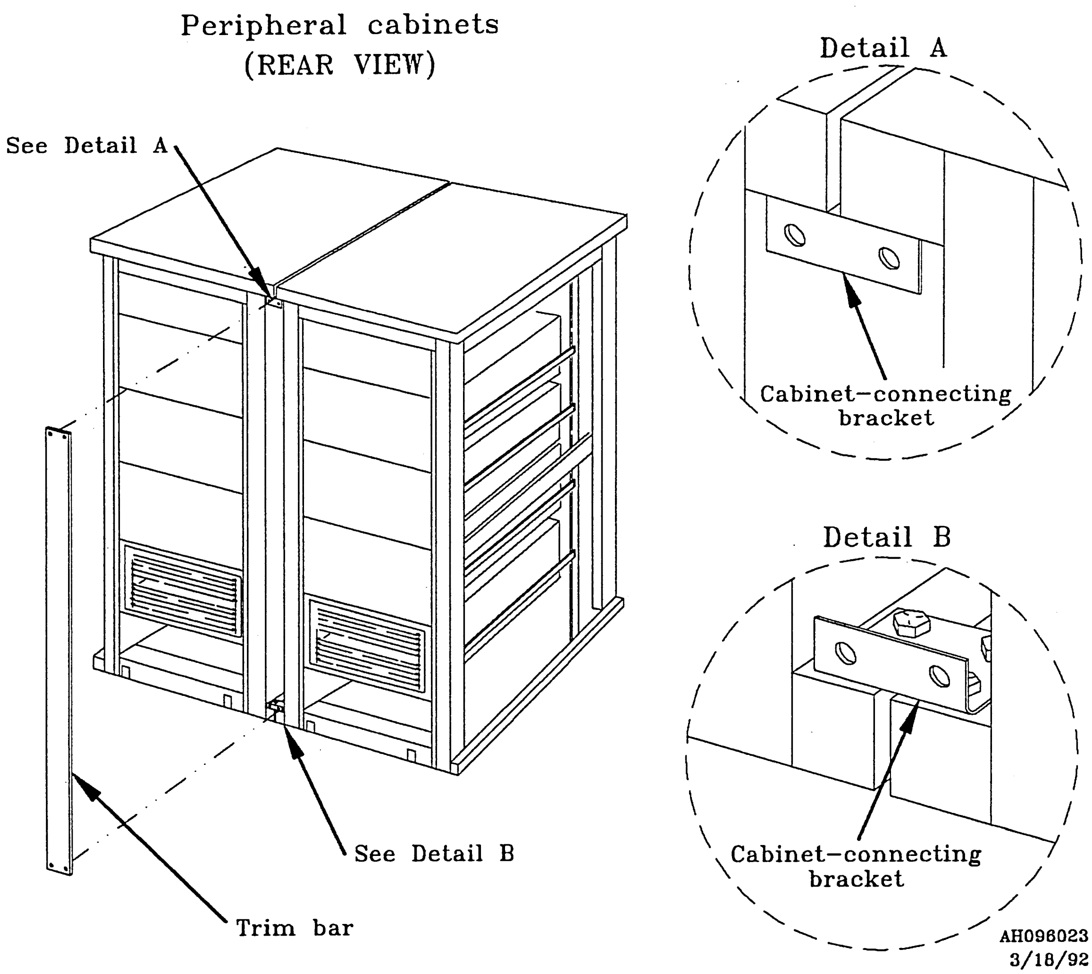
1. Cut the bands around the accessories on the pallet.
2. Remove the plastic film covering the accessories.
3. Remove each box from the pallet.
4. Assemble the table, and position it near the cabinets.
5. Remove the accessories from their boxes.
6. Place the accessories on the table and ensure they are 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.

2.5.4 Peripheral cabinet installation procedures

This section describes the procedure to install a peripheral cabinet, including mating, securing, and cabling the cabinets. Mounting hardware (screws, retaining plates) is used to mount the peripheral cabinet to the processor cabinet.

1. Position and level the processor cabinet(s).
2. After positioning the processor cabinet(s), secure additional cabinets (to the processor I/O cabinet of a C230 or C240 system) with cabinet connecting-brackets. Figure 2–5 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 2–5, Cabinet-connecting bracket



3. 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.
4. 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.

5. Position the peripheral cabinet next to the processor cabinet.
6. Lower the feet of the peripheral cabinet so the wheels are able to spin freely.
7. Adjust the level of the peripheral cabinet at the connecting edge to match the processor cabinet.
8. Level the remaining sides of the peripheral cabinet.
9. Install the remaining 2 bolts, and tighten all bolts on the 4 cabinet connecting brackets between the new peripheral cabinet and the existing cabinet.
10. Recheck the leveling of the cabinets.
11. 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.
12. 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.
13. 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.
14. 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 IDC/ITC board installation

This section gives detailed procedure for an initial installation of the IDC/ITC board. Additional steps, such as the cable routing or connections, may not be necessary when replacing an IDC/ITC. However, all installation steps should be read during either a full installation or a board replacement to guarantee proper technique and sequence of events.

2.6.1 IDC/ITC board installation procedures

The IDC/ITC is installed in an available CCU slot in the computer's card cage. This section describes the procedures for installing the IDC/ITC CCU.

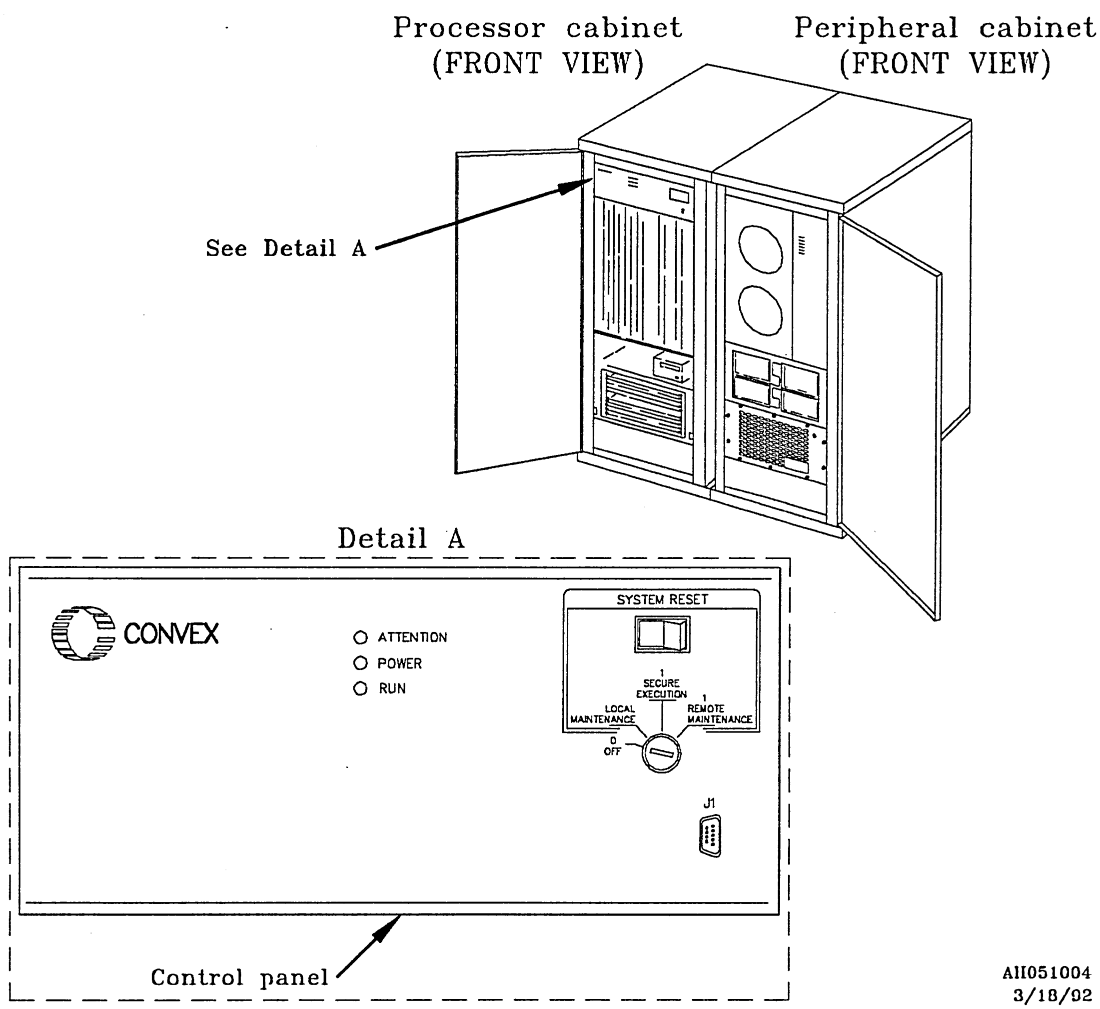
> **CAUTION**
>
> Failure to remove power before installing the IDC/ITC board assembly will damage electronic components on the IDC/ITC board assembly. Refer to *Managing ConvexOS: Operations Guide* for the shutdown procedures for a CONVEX computer.

1. Turn the processor's front control panel keyswitch to the **OFF** position as shown in Figure 2–6.

CAUTION

Remove keys from the front panel to avoid inadvertent power-on during the remainder of the procedure.

Figure 2–6, Typical front panel power control switch



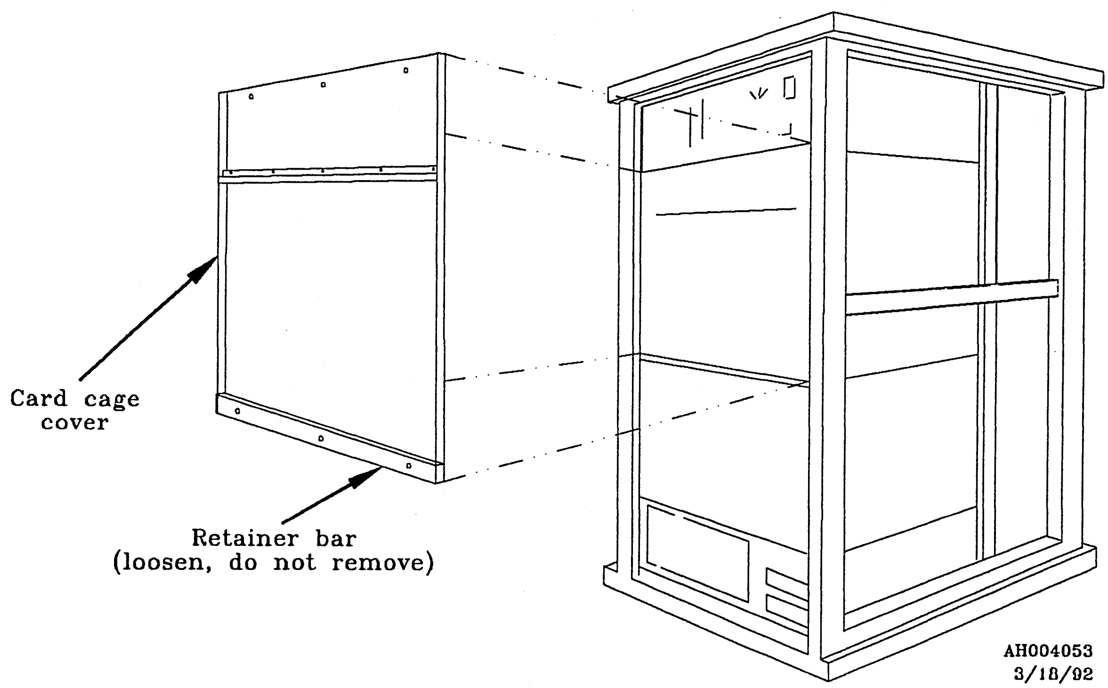
2. Set the power controller's main circuit breaker to the **OFF** position and set the LOCAL/REMOTE switch to the **OFF** position.

CAUTION

The IDC/ITC 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.

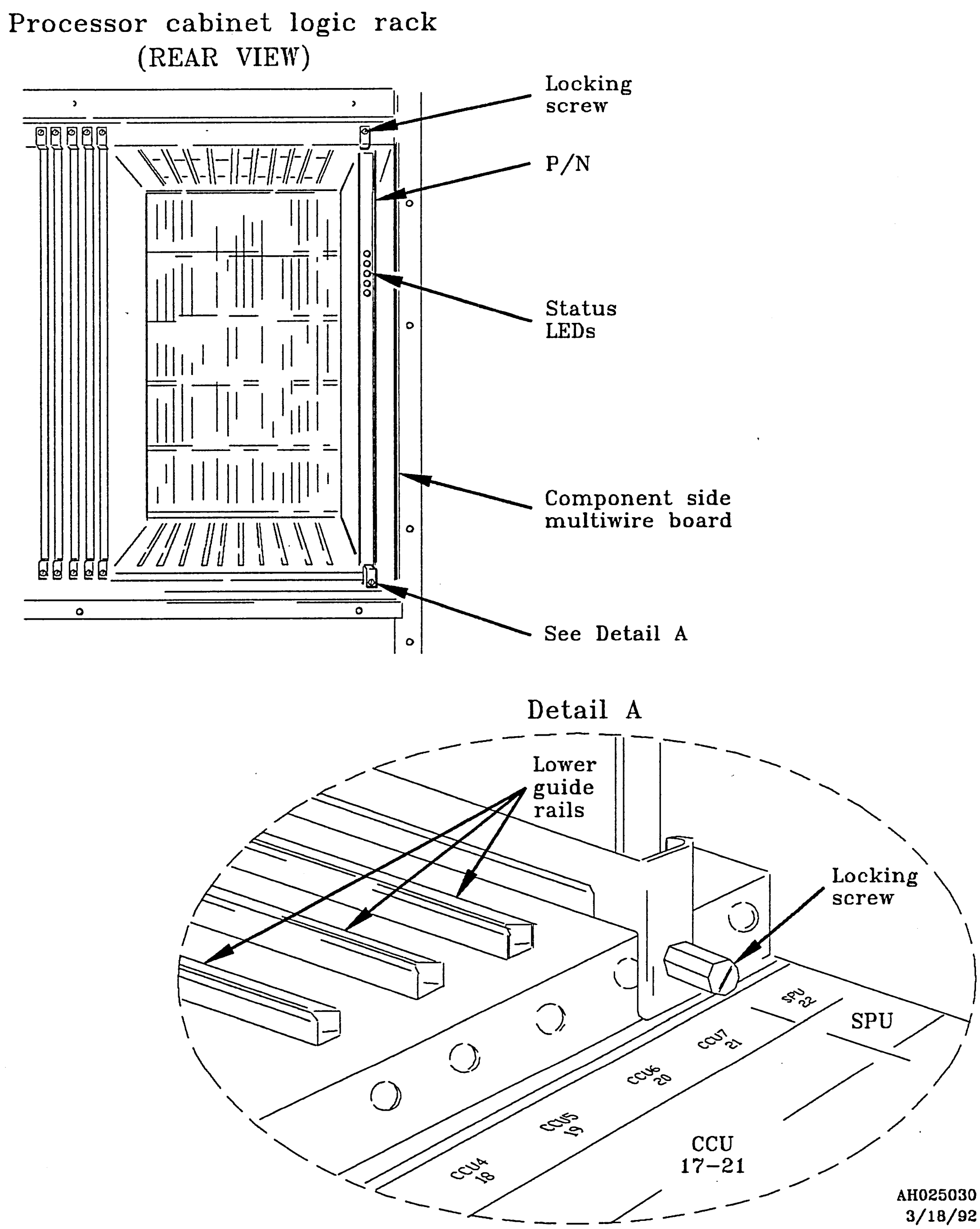
3. Remove the cover plate on the front of the processor card cage as shown in Figure 2–7.

Figure 2–7, Card cage cover plate



4. Install the IDC/ITC board assembly, with the component side facing right, in one of the available **CCU** slots in the processor cabinet's logic rack as shown in Figure 2–8.

Figure 2–8, 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/ITC board assembly as shown in the previous figure.

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

2.6.2 Backplane-to-bulkhead cabling

This section contains procedures for the installation of IDC/ITC cabling. These procedures includes cabling from the CONVEX backplane to the CONVEX bulkhead, from the CONVEX bulkhead to the disk drive(s), power sequencing control cabling, and power cabling.

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. 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 procedure 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. 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 2-9.

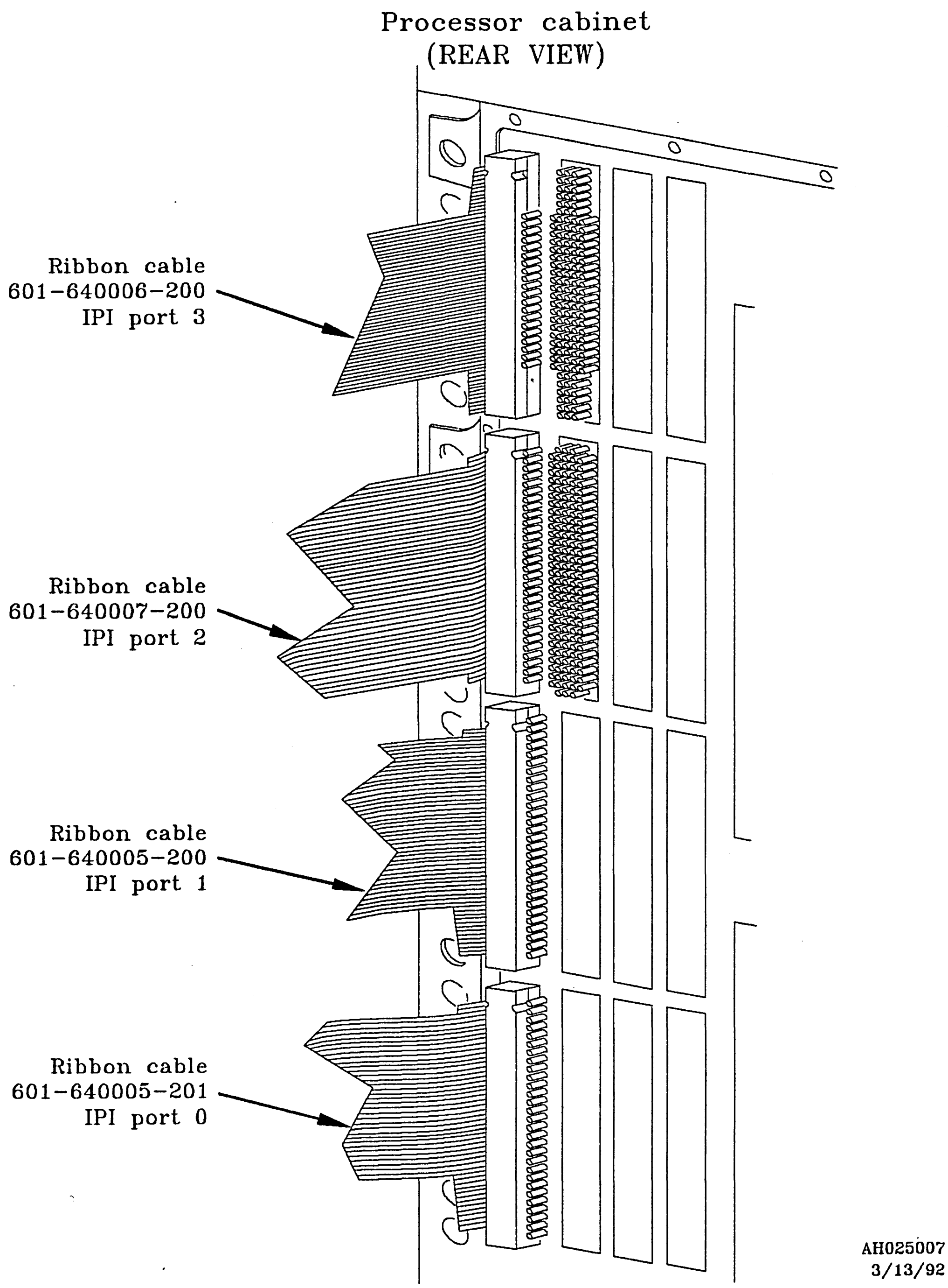
NOTE

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.

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.

Figure 2–9, Backplane cabling connections



4. Remove two pair of vertically adjacent bulkhead cover panels from the bulkhead located on the side of the processor cabinet.

> **NOTE**
>
> Refer to Figure 2–10 for more information during the following steps.

5. Install a connection plate (CONVEX Part Number 320-000299-500) on each open bulkhead slot.
6. From the processor side of the bulkhead, pass the IPI cable connectors on the IPI interface board (CONVEX Part Number 411-000202-200) through the openings in the upper two connection plates. Align the nut plates on the IPI cable connectors with the slots in the connection plates and install the locking screws.
7. From the processor side of the bulkhead, pass the IPI cable connectors on the IPI interface board (CONVEX Part Number 411-000201-200) through the openings in the lower two connection plates. Align the nut plates on the IPI cable connectors with the slots in the connection plates and install the locking screws.
8. Connect the ribbon cables to the ribbon cable connectors on the two IPI interface boards.

2.6.3 IDC-to-drive cabling

The following sections discuss how to connect cabling and various components of the IDC portion of the product. Skip these sections if you are making only an ITC connection.

2.6.3.1 Bulkhead-to-drive cabling

For the IDC, four ribbon cables and four IPI cables are used to connect the initial four disk drives to the IDC/ITC. 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.

Connect the four IPI cables from the four initial disk drives to the four IPI connectors on the bulkhead as shown in Figure 2–10.

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

Figure 2–10, Bulkhead cabling connections

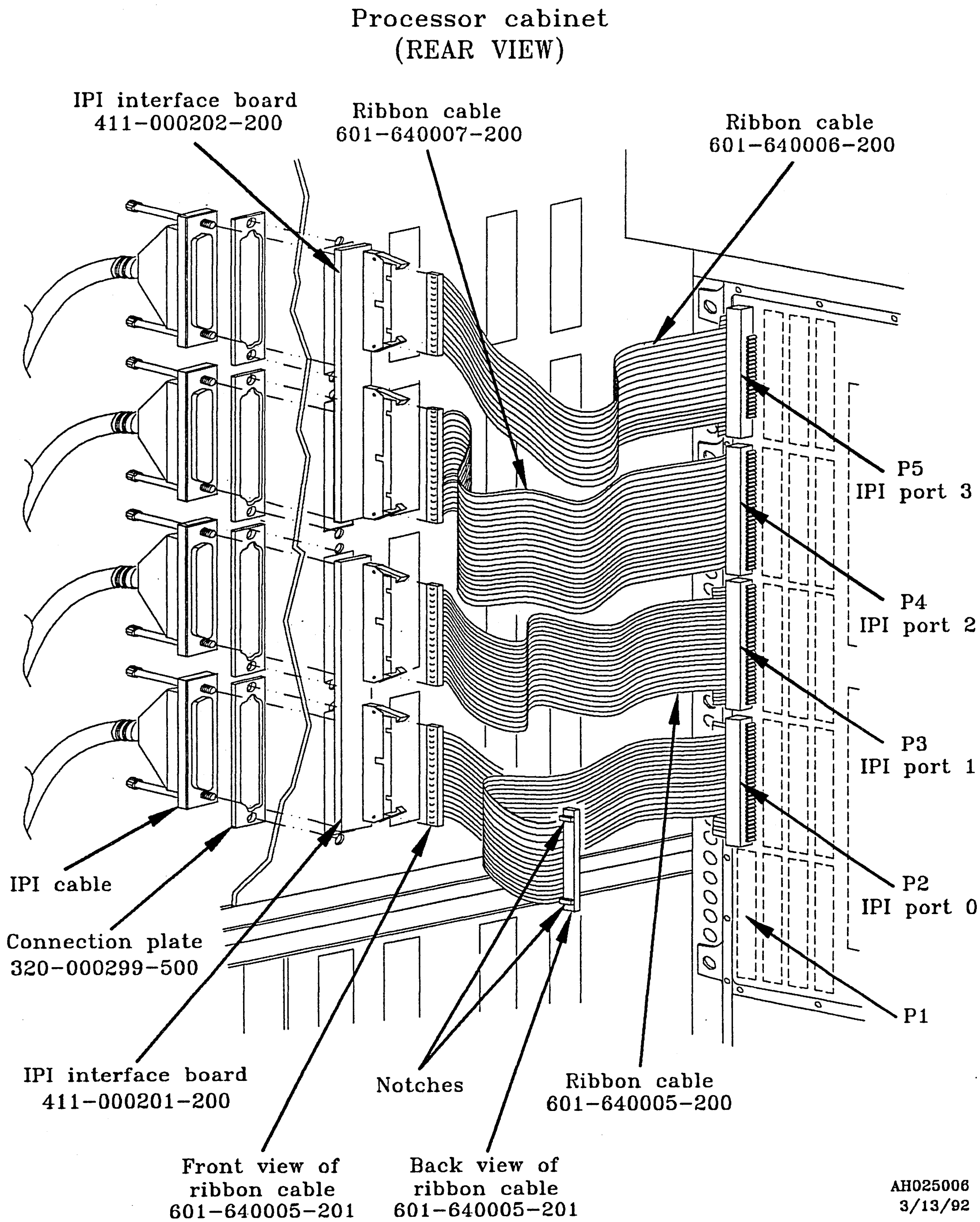
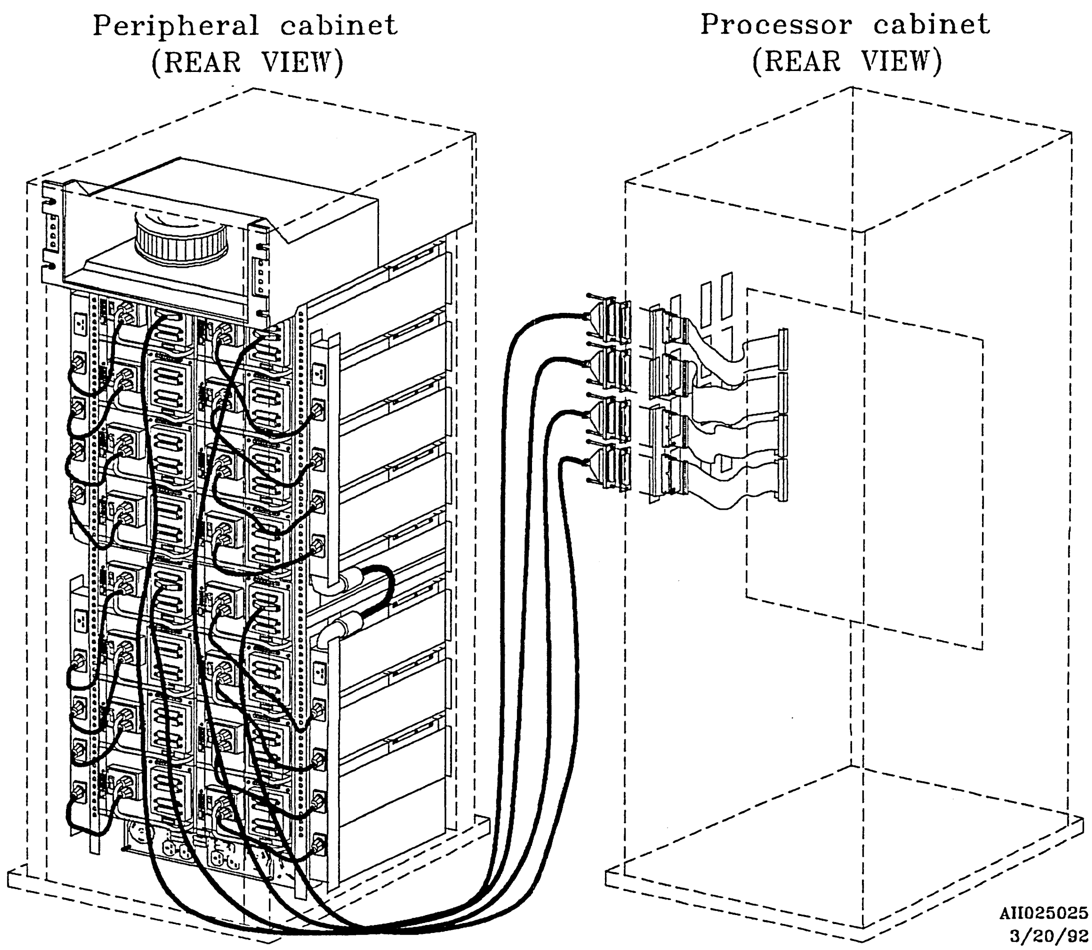


Figure 2–11, Bulkhead IPI cable connections



2.6.3.2 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

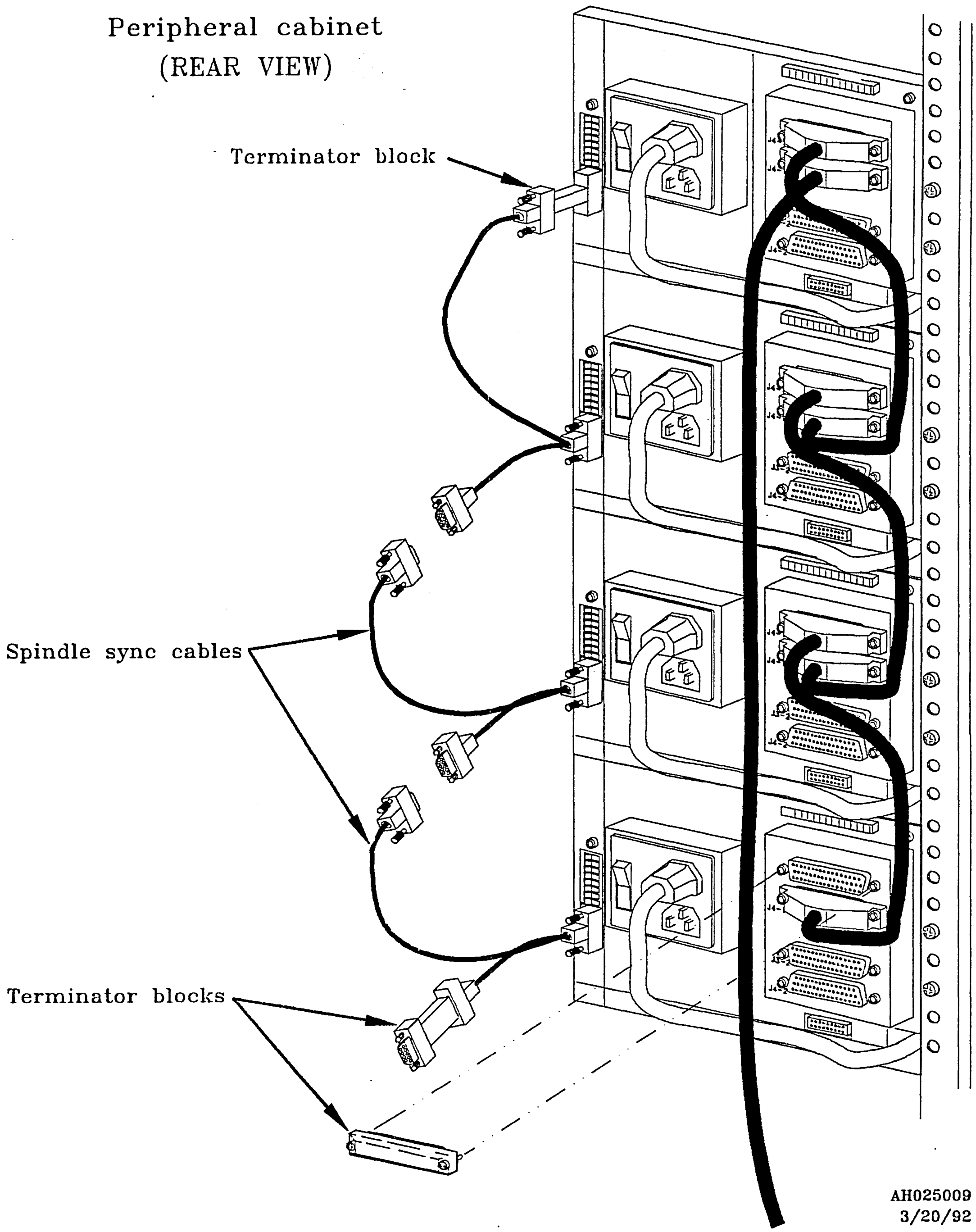
are no other disk drives in the daisy-chain, a terminator connector is placed on the **J3-1** connector of the last disk drive.

> **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 2–12 illustrates the daisy-chain cabling between two 6-Mbyte disk drives, the terminator connector, and the spindle synchronization cabling.

Figure 2–12, Daisy-chain and spindle synchronization cabling



2.6.3.3 Power sequencing control cabling

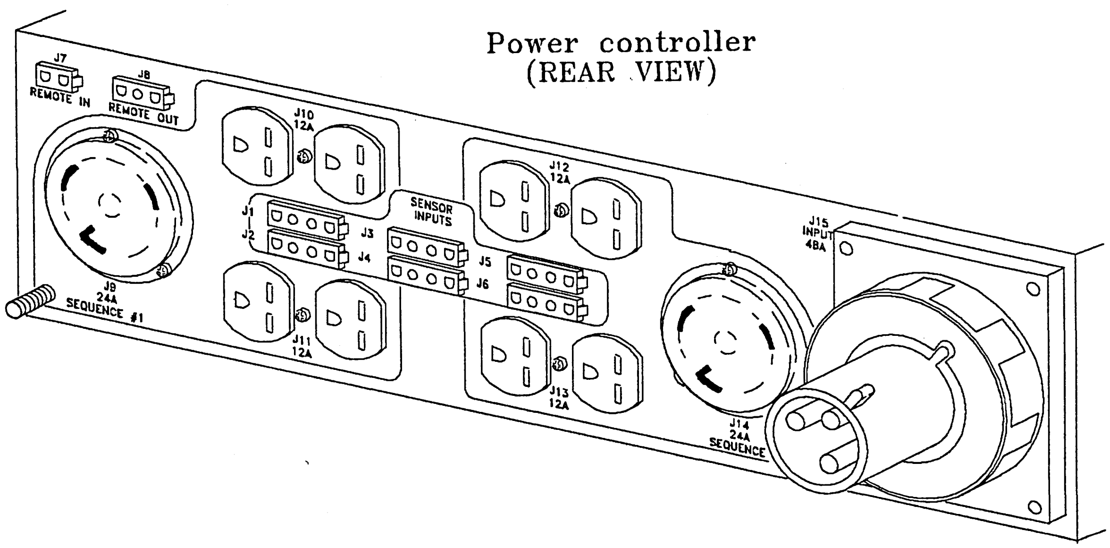
The power sequencing control cable allows control of peripheral cabinet power from the power controller. The power controller provides 220V to the peripheral devices. Use the following procedure to install the power sequencing control cable:

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 2–13.

Figure 2–13, Power controller remote-in connection



AH025035
3/17/92

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).

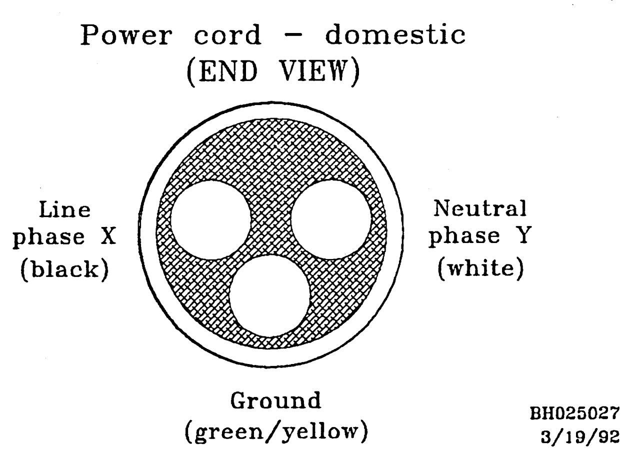
2.6.3.4 Power cabling, domestic

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

> **NOTE**
>
> The domestic power cord has a HUBBELL Part Number 360C6 W receptacle on the power controller end and a 360P6 W plug on the opposite end. A receptacle, HUBBELL Part Number 360C6 W (CONVEX Part Number 304-000036-001), must be installed on the site power cord.

Figure 2–14 illustrates the conductor configuration for a domestic power cord.

Figure 2–14, Power cord conductor configuration – Domestic



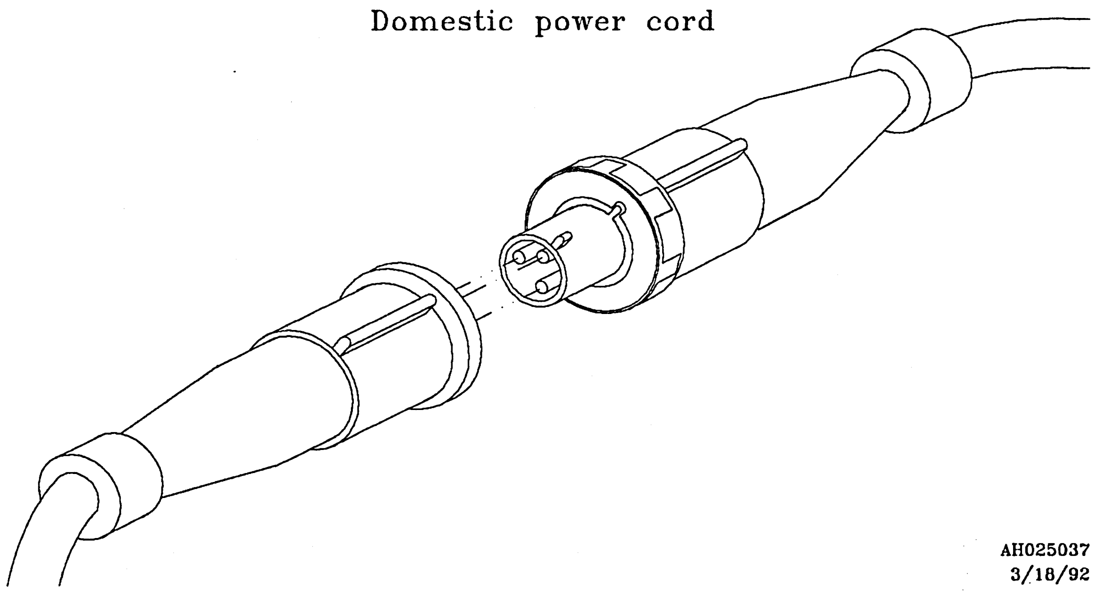
The power controller power cord should be connected to the site power cord according to the following procedure:

> **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. Install the receptacle on the site power cord.
3. Connect the site power cord to a site power source at a site power distribution panel.
4. Connect the power controller power cord to the site power cord as shown in Figure 2–15.

Figure 2–15, 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 keyswitch to the **ON** position.

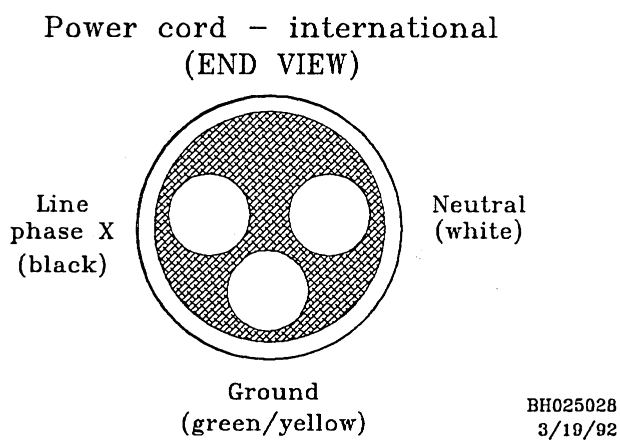
2.6.3.5 Power cabling, international

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

> **NOTE**
>
> The international power cord has a HUBBELL Part Number 360C6W receptacle on the power controller end but does not have a plug on the opposite end.

Figure 2–16 illustrates the conductor configuration for an international power cord.

Figure 2–16, Power cord conductor configuration - International



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

> **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 power cord to a site 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 keyswitch to the **ON** position.

2.7 Powering up the system

After all the cables have been connected, use the following procedure to power the system up:

1. Replace the cover on the front of the processor card cage.
2. Insert the front control panel key into its keyhole.
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 keyswitch to the **ON** position.

Chapter 3
Software and test

3.1 Overview

This chapter contains guidelines for integrating an integrated disk/tape channel (IDC/ITC) into the CONVEX operating system (ConvexOS). A brief description of disk striping and spindle synchronization for the IDC, and information about the IDC/ITC diagnostic programs is also provided.

IDC/ITC modules, intelligent 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. ConvexOS 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 and C3000 Series computers contain all software drivers for the hardware supported by CONVEX. This means that a system generation is not required when an IDC/ITC is installed in a CONVEX computer.

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

System-level hardware is identified to ConvexOS via a configuration file (/ioconfig) located on the service processor unit (SPU) disk. This file contains a description of all channel control units (CCUs), interfaces, controller boards, and peripheral devices for your system. The boot process reads this file to determine what devices are present.

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 an IDC/ITC 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
- MTC-IP3—ITC IPI-3 controller
- MTD-R90—ITC IPI tape drive

Figure 3–1 shows a typical /ioconfig file, including entries for an IDC with three 3-Mbyte/second 1-Gbyte drives (one master drive and two slave drives), three 6-Mbyte/second 1-Gbyte drives (one master drive and three slave drives), and one ITC with two R-90 drives.

> **NOTE**
>
> The booting order of the IDC disk units determines the names of the units in the file system /dev directory.

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 5
    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
itc 0
    ipi 0
     drvr MTC-IP3
          unit 0 type MTD-R90
          unit 1 type MTD-R90
```

In Figure 3-1, `idc 5` identifies the CCU slot number for the IDC, and `itc 0` identifies the CCU slot number for the ITC. The value can be any valid CCU slot number; it must correspond to the CCU slot that contains the IDC or ITC.

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

Whenever an IDC or ITC configuration is added or removed, the information in the hardware section of the /ioconfig file must be changed, otherwise system operation problems will occur.

Refer to *Managing ConvexOS: Configuration Guide* when making a change to this file.

3.4 Testing the IDC/ITC

The `idc4010` diagnostic program tests the IDC and related hardware, while the `itc4000` diagnostic program tests the ITC and related hardware. These programs verify that:

- The CCU is electrically sound in that its microprocessor can execute instructions from its ROM, access its data RAM, and access the PBUS.
- The subsystem is functional, reliable, and capable of supporting ConvexOS.

`itc4000` verifies that the normal path software can communicate with the R-90 tape drive and perform basic read, write, or verify tape operations.

The programs also provide an interactive debugger that can execute commands from a script file.

The `idc4010` and `itc4000` diagnostic programs are offline programs that must be executed on the SPU while the CPU is halted. The procedures for executing these tests are beyond the scope of this manual. However, this information is contained in the *CONVEX Integrated Disk Controller Subsystem (*`idc4010`*) Diagnostics Manual* and *CONVEX Integrated Tape Controller Subsystem (*`itc4000`*) Diagnostics Manual.* These manuals should be consulted before running the tests.

Chapter 4
Maintenance procedures and IPB

4.1 Overview

This chapter contains removal and replacement procedures and an illustrated parts breakdown (IPB) for all field replaceable units (FRUs) for the IDC/ITC.

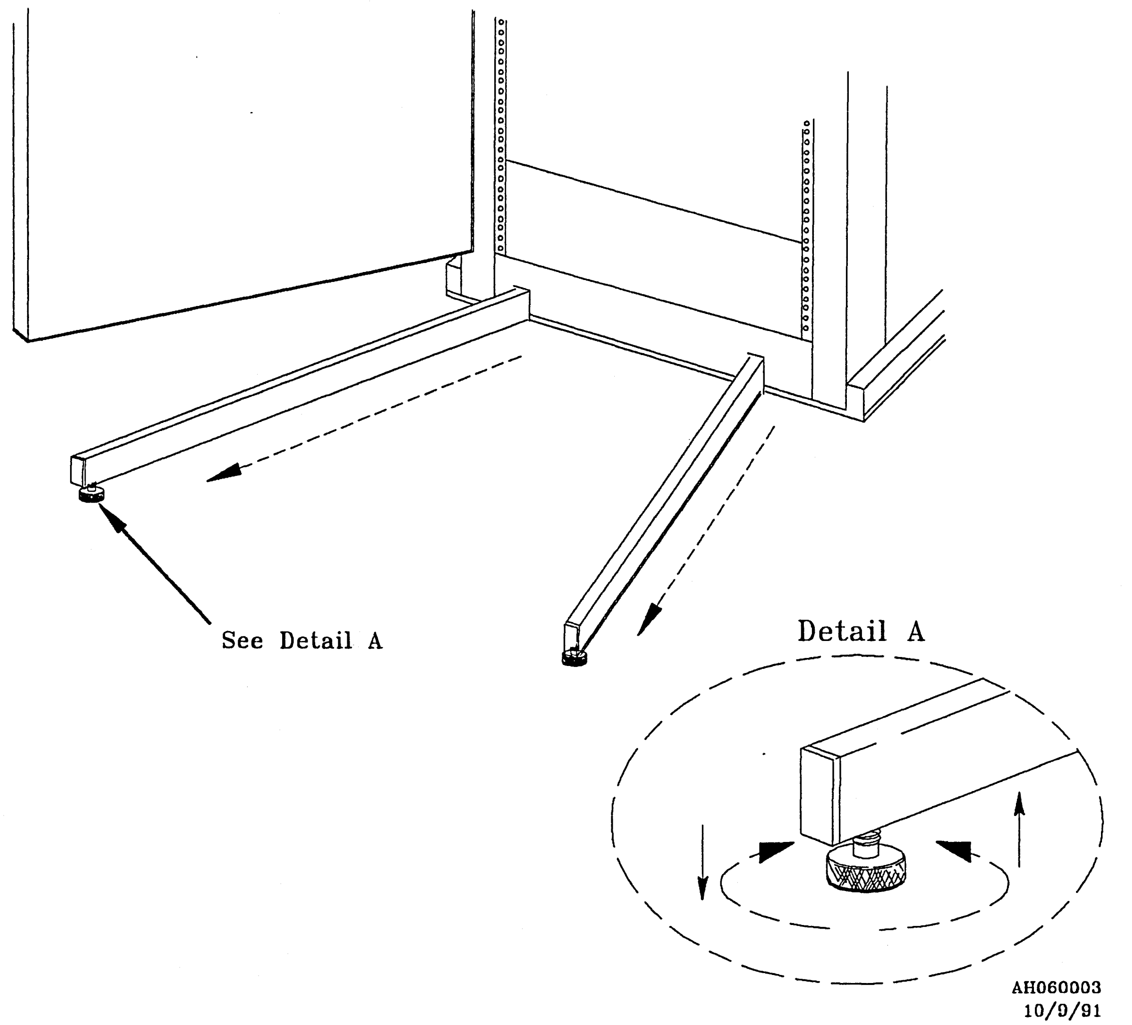
4.2 Maintenance safety procedures

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 the remainder of this chapter.

> **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 feet until they are in firm contact with the floor as shown in Figure 4–1.

Figure 4–1, Peripheral cabinet stabilizer bars



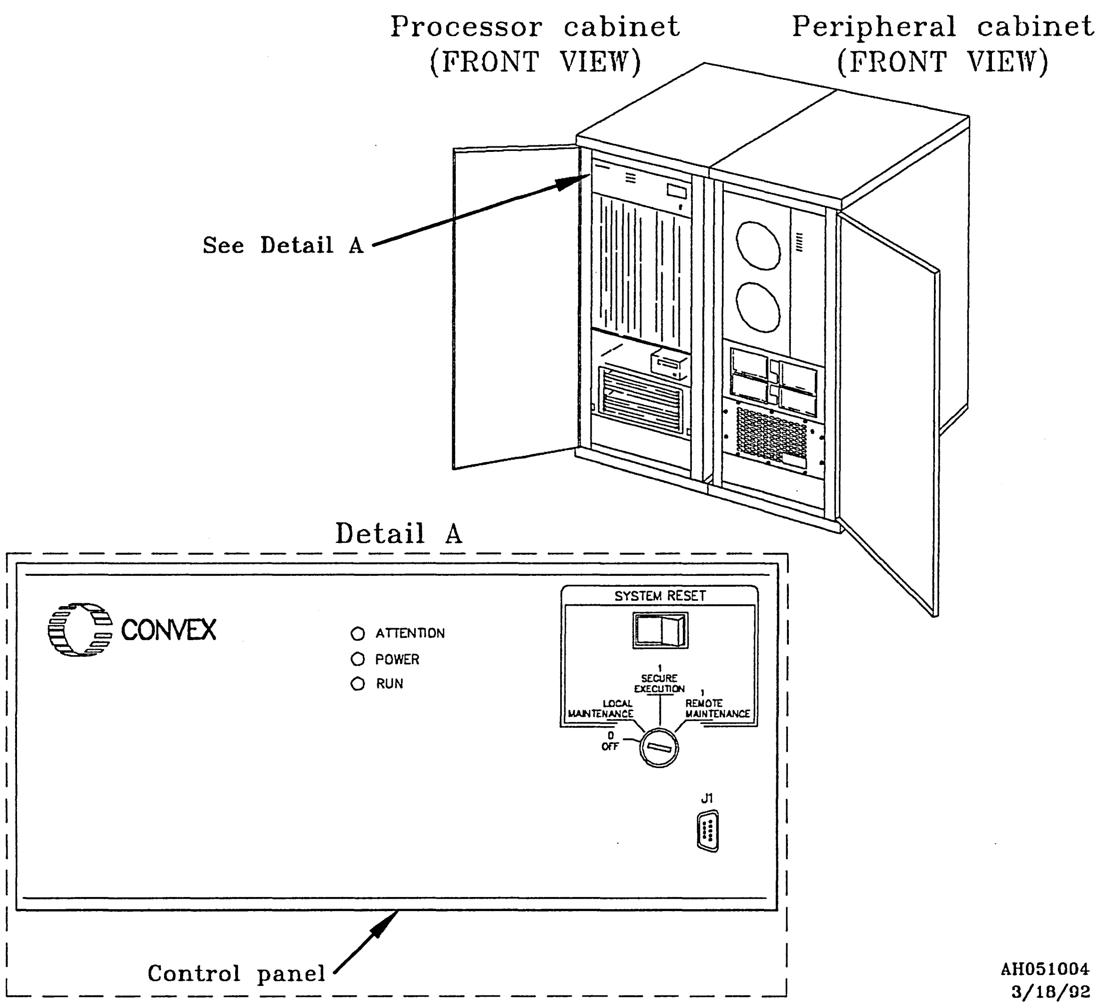
CAUTION

Failure to remove power before installing or removing equipment from the computer card cage will damage electronic components. Refer to *Managing ConvexOS: Operations Guide* for power down procedures on a CONVEX computer.

2. Turn the processor's front panel keyswitch to the **OFF** position as shown in Figure 4–2.

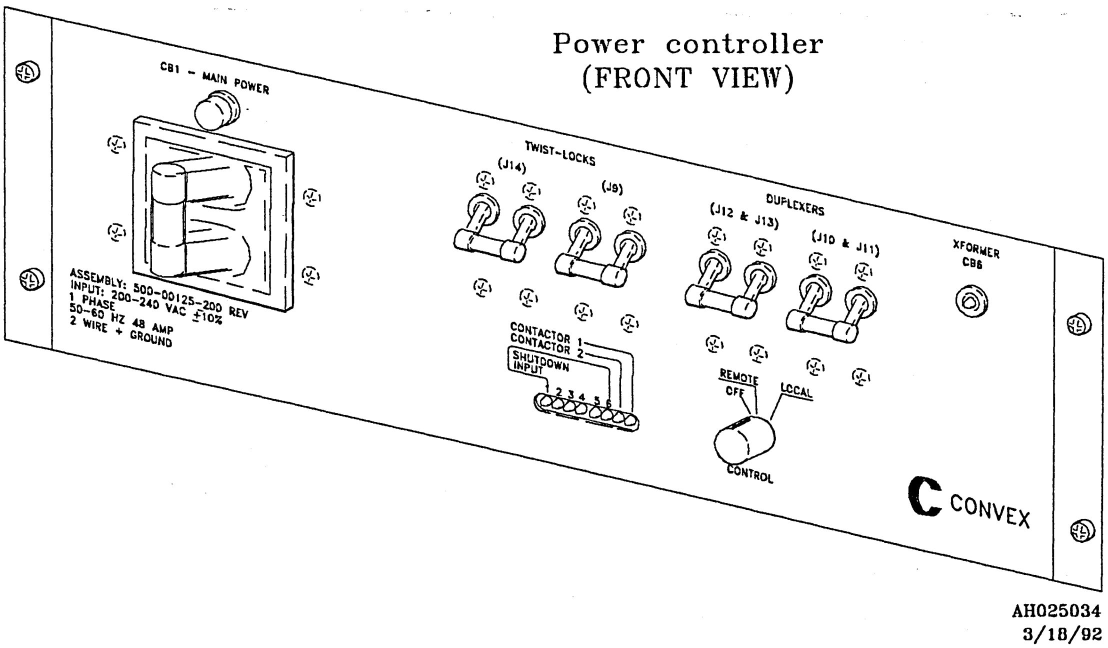
> **CAUTION**
>
> Remove keys from the front panel to avoid inadvertent power-on during the remainder of the procedure.

Figure 4–2, Typical front panel power control switch



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.

Figure 4–3, Power controller switches



4.3 IDC/ITC

This section defines the removal and replacement procedures for the IDC/ITC board assembly.

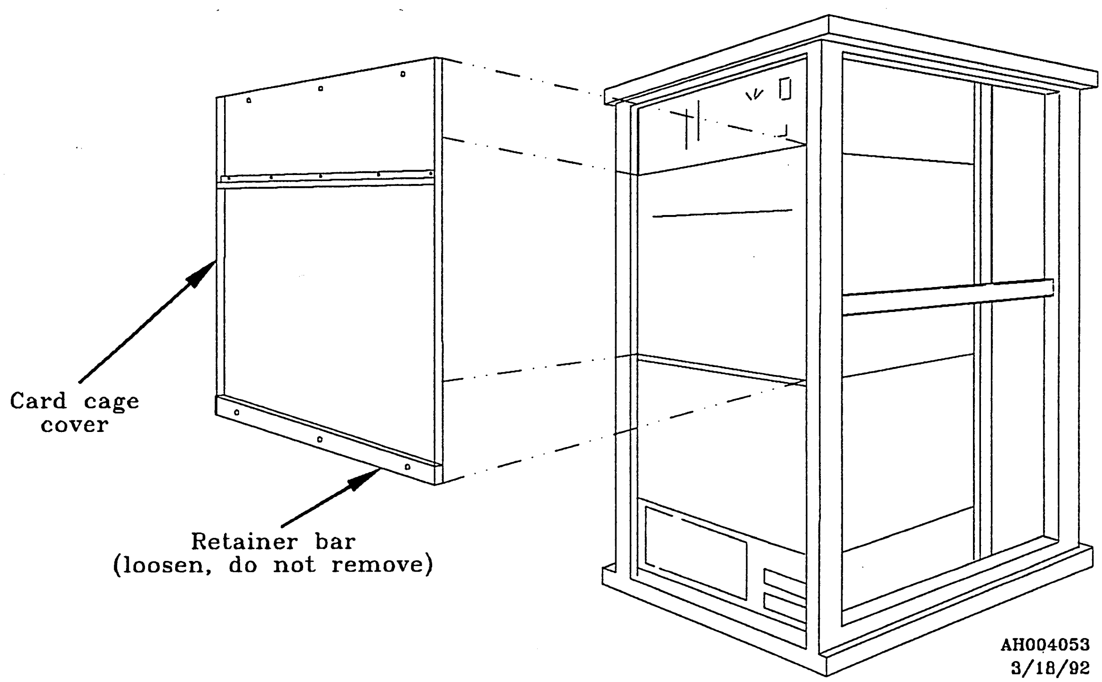
4.3.1 Removal

1. Complete the steps listed in Section 4.2, "Maintenance safety procedures."

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

2. Remove the cover plate on the front of the processor card cage as shown in Figure 4–4.

Figure 4–4, Card cage cover plate

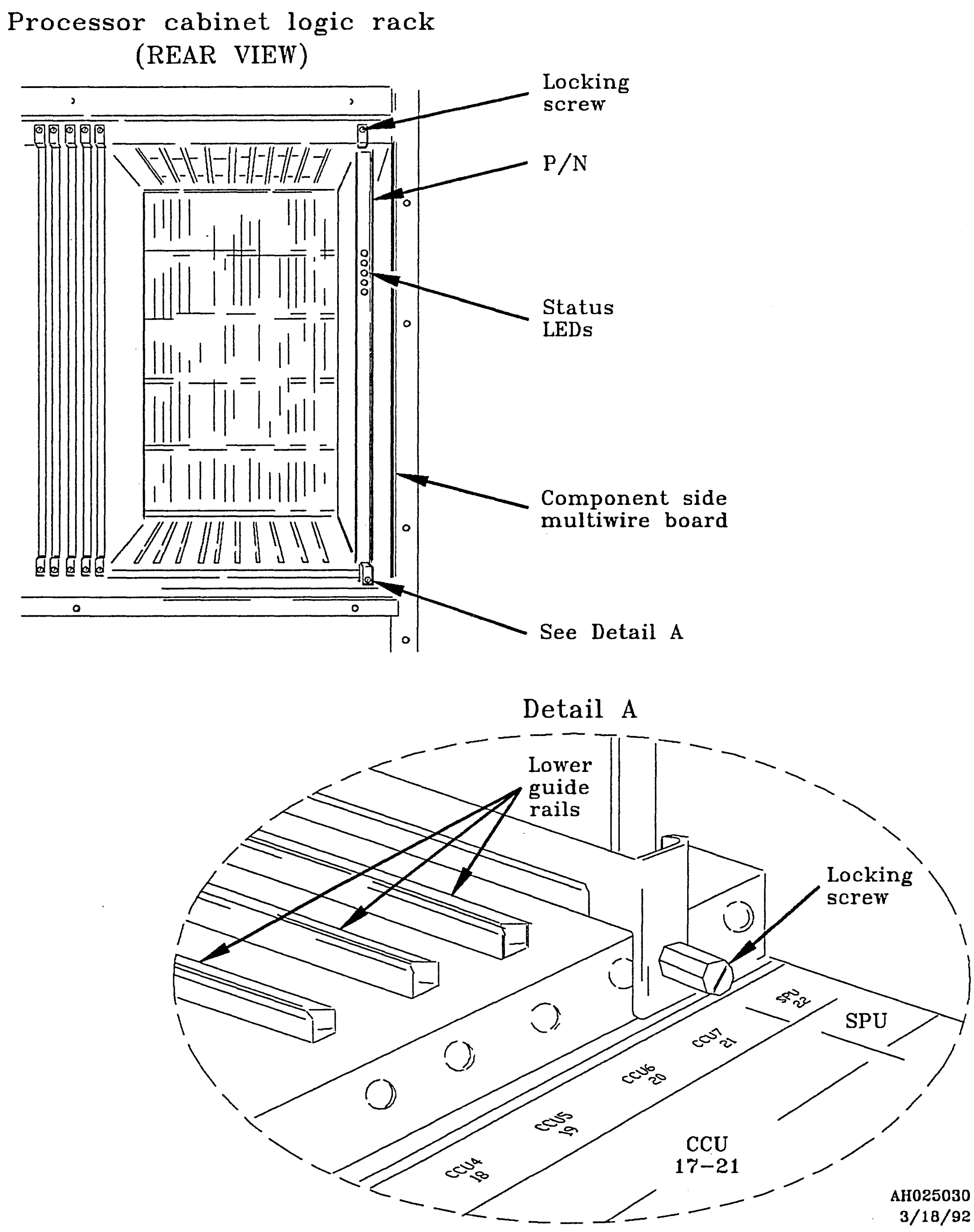


CAUTION

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/ITC as shown in Figure 4–5.
4. Remove the IDC/ITC board assembly from the CCU slot in the processor cabinet's logic rack.

Figure 4–5, CCU slots and mounting hardware



4.3.2 Replacement

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

> **NOTE**
>
> If an IDC/ITC is to be initially installed, refer to Chapter 2, "Unpacking and installation," section 2.6, "IDC/ITC board installation," for installation information.

The following procedure is for the replacement of an IDC/ITC board.

1. Complete the steps listed in Section 4.2, "Maintenance safety procedures."
2. Install the IDC/ITC in the CCU slot in the processor cabinet's logic rack as shown in Figure 4–5.

> **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/ITC board assembly as shown in Figure 4–5.
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 keyswitch to the **ON** position.
7. Return the peripheral cabinet stabilizer bars to their retracted positions.

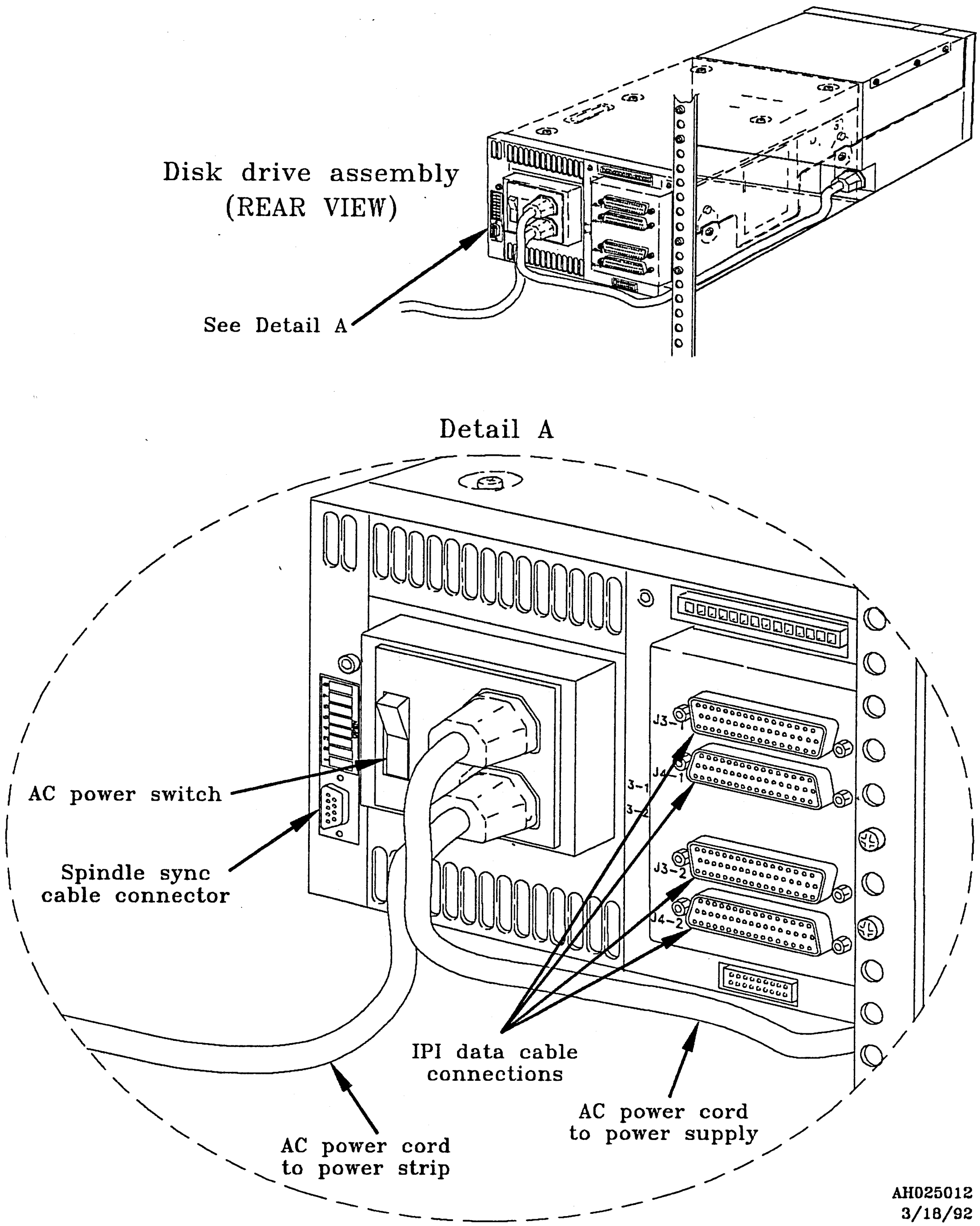
4.4 IDC disk drive module

This section defines the removal and replacement procedures for the disk drive module.

4.4.1 Removal

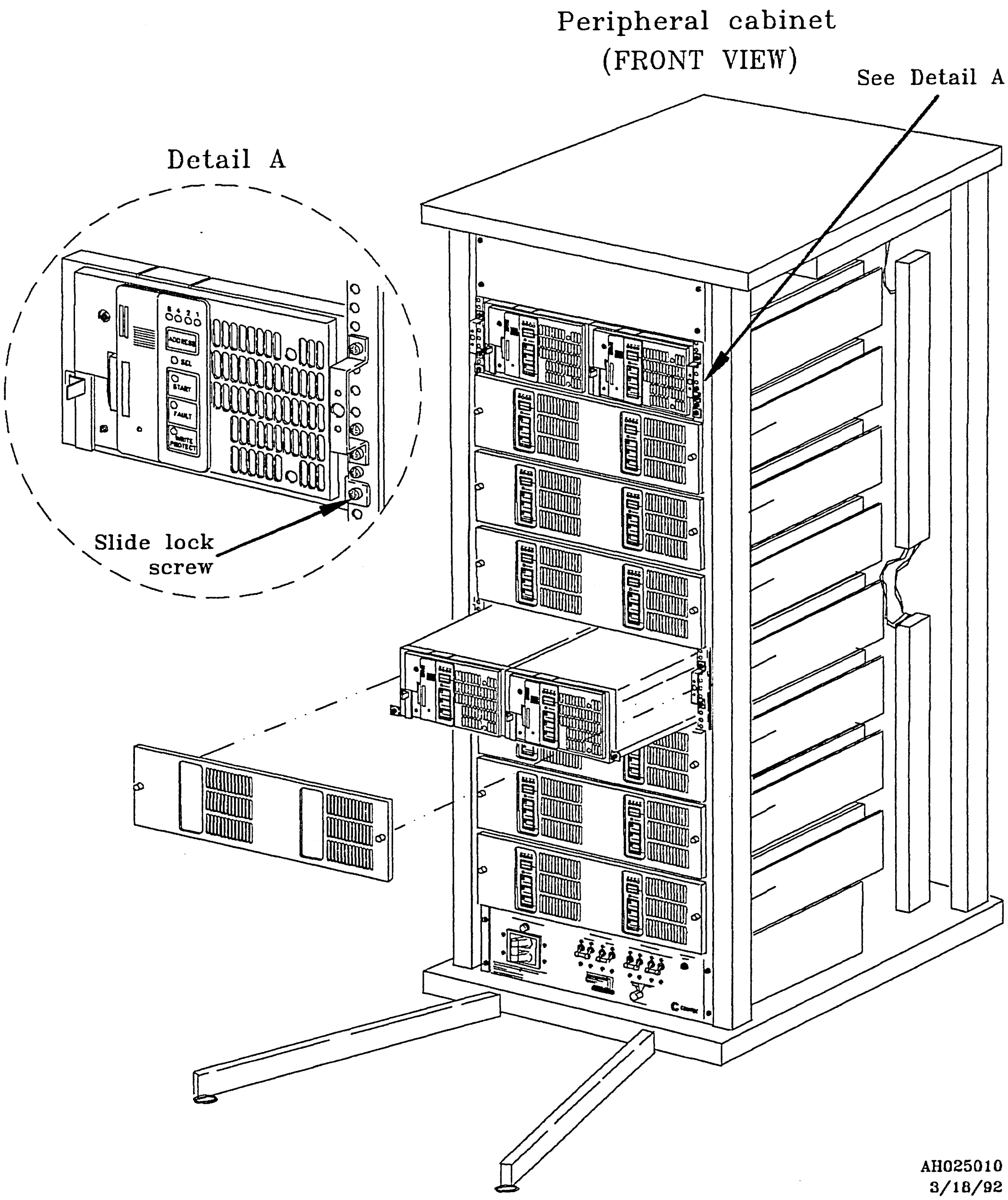
1. Complete the steps listed in Section 4.2, "Maintenance safety procedures."
2. Remove the peripheral cabinet rear panel.
3. Turn the ac power switch on the rear of the right drive to the **OFF** position. Then disconnect the ac power cord from the ac power cord connector on the rear of the right drive as shown in Figure 4–6.
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.
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 4–5.
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.

Figure 4–6, Drive connections



7. Release the 2 front cover captive-lock screws and remove the front cover as shown in Figure 4–7.

Figure 4–7, Front cover and slide lock



8. Release the 2 front chassis captive-lock screws and extend the assembly on its slides as shown in Figure 4-7.

CAUTION

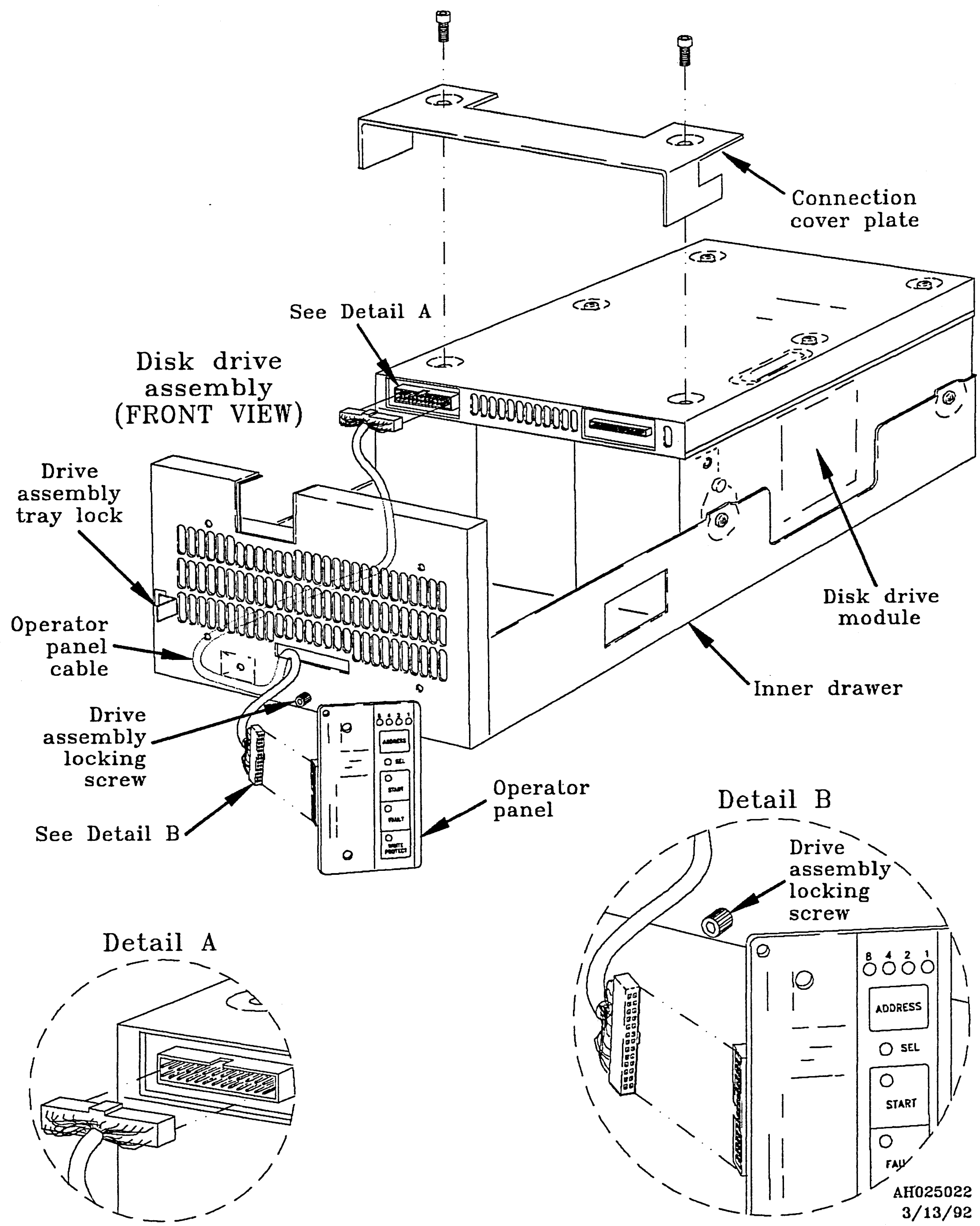
The operator panel can be damaged by ESD. A grounded wrist strap (or other grounding method) must be used when handling the operator panel.

NOTE

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

9. Remove the 2 screws in the operator panel and disconnect the operator panel cable from the operator panel as shown in Figure 4–8.

Figure 4–8, Operator panel cable



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

WARNING

Because a drive assembly weighs 60 pounds, 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.

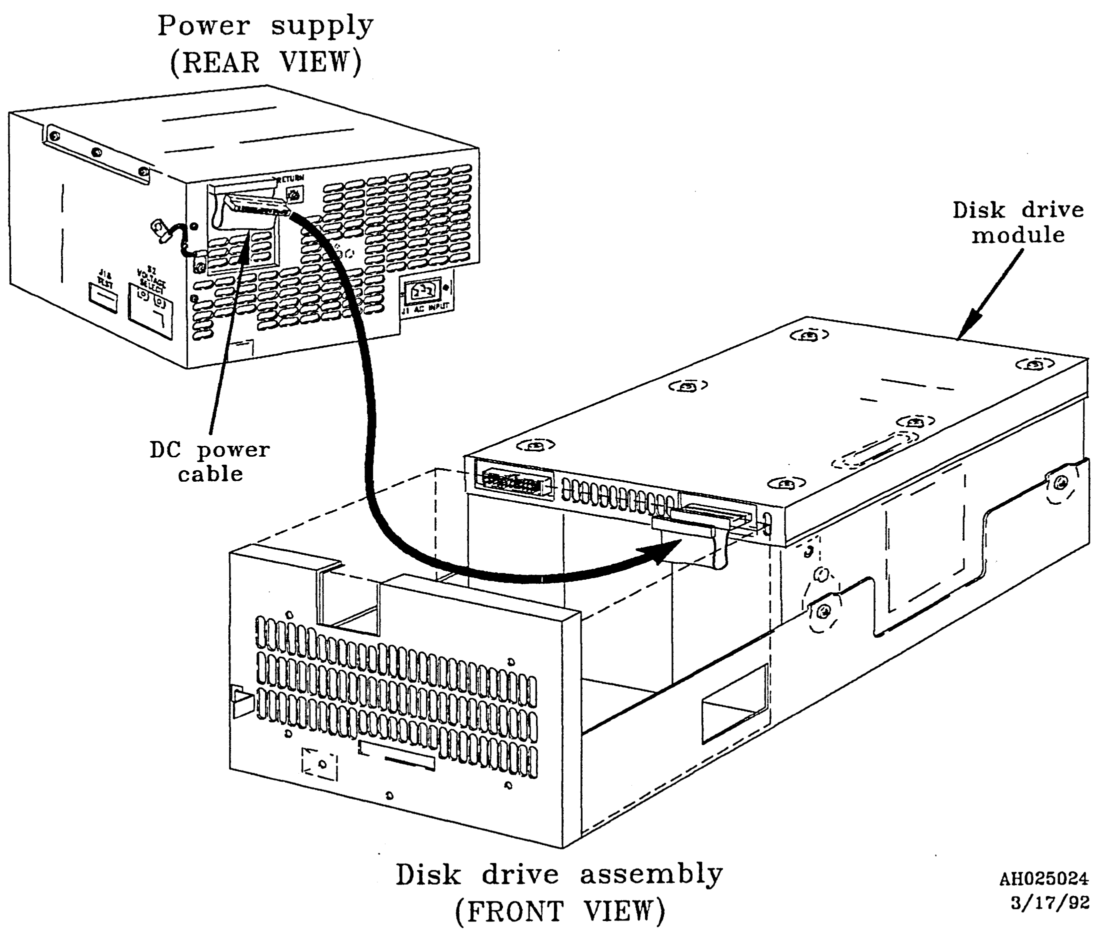
11. Press the drive assembly tray lock to the right, then lift the drive assembly tray free of the slide tray.
12. Disconnect the ac power cord from the assembly AC jack at the rear of the disk drive module.
13. Remove the 2 locking screws on the connection cover plate and remove the plate.
14. Loosen 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 4–9.

Figure 4–9, dc power cable



17. Disconnect the operator panel cable from the disk drive module as shown in Figure 4-8.
18. Remove the disk drive module from the drive assembly tray.

4.4.2 Replacement

1. Complete the steps listed in Section 4.2, "Maintenance safety procedures."

> **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 4-9.
4. Connect the operator panel cable to the disk drive module as shown in Figure 4-8.
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 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 pounds, 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.

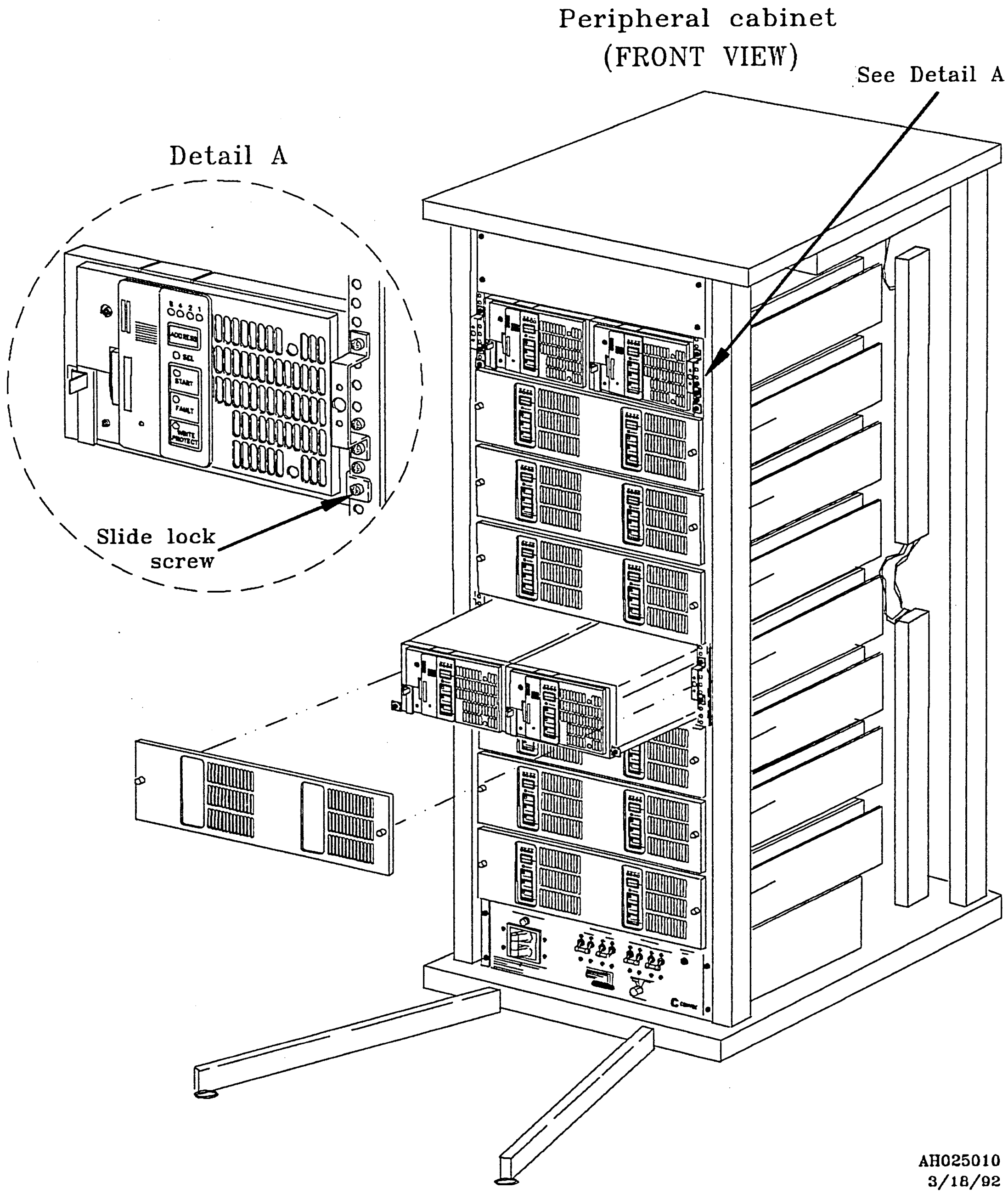
10. Press the drive assembly tray lock to the right, then lower the drive assembly tray into the slide tray.
11. Install the drive assembly locking screw located on the front of the tray.

CAUTION

The operator panel can be damaged by 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 4-8.
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 4–10.

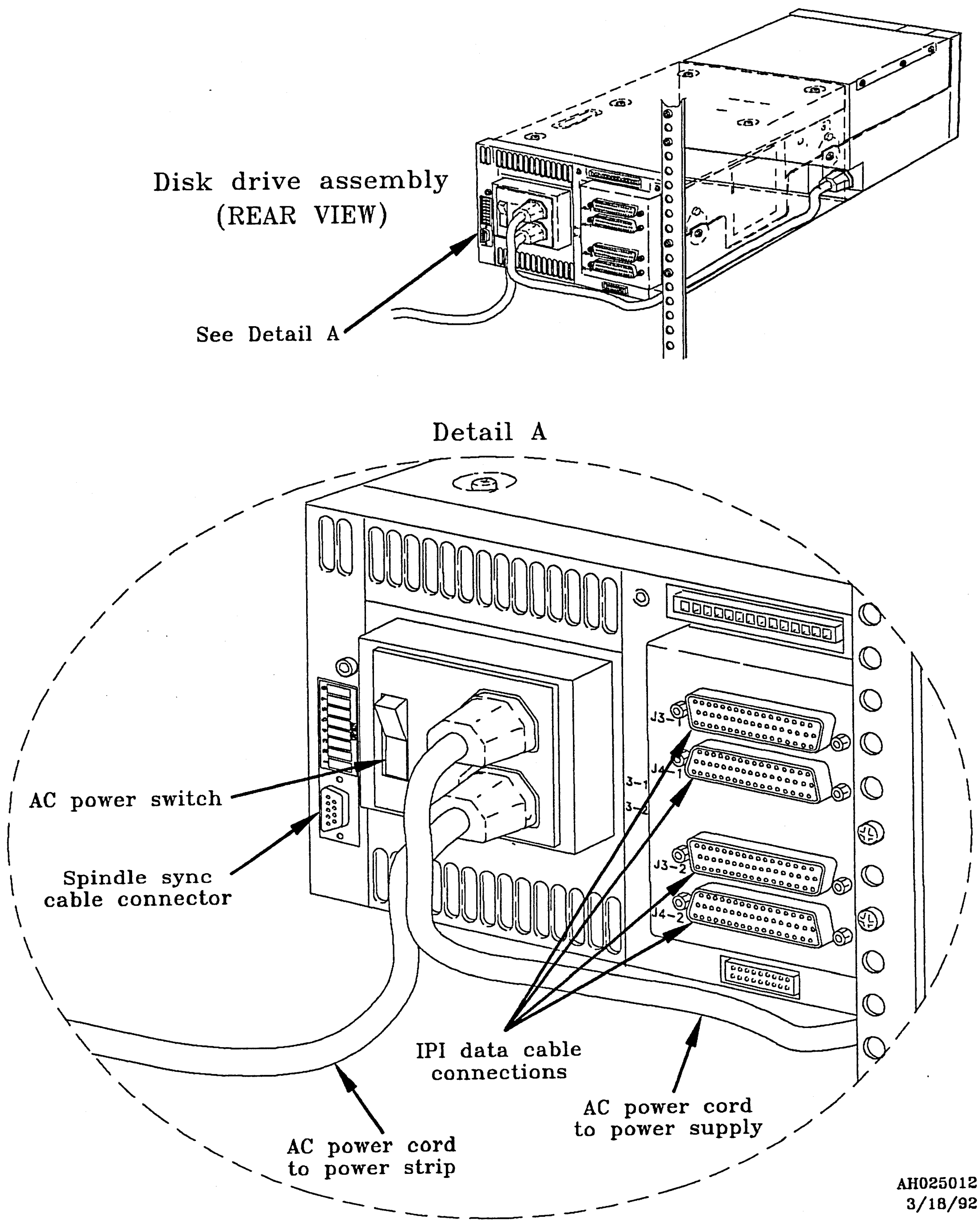
Figure 4–10, Front cover and slide lock



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 4–11.

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 4–11, 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 4–11.
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 keyswitch to the **ON** position.
23. Return the peripheral cabinet stabilizer bars to their retracted positions.

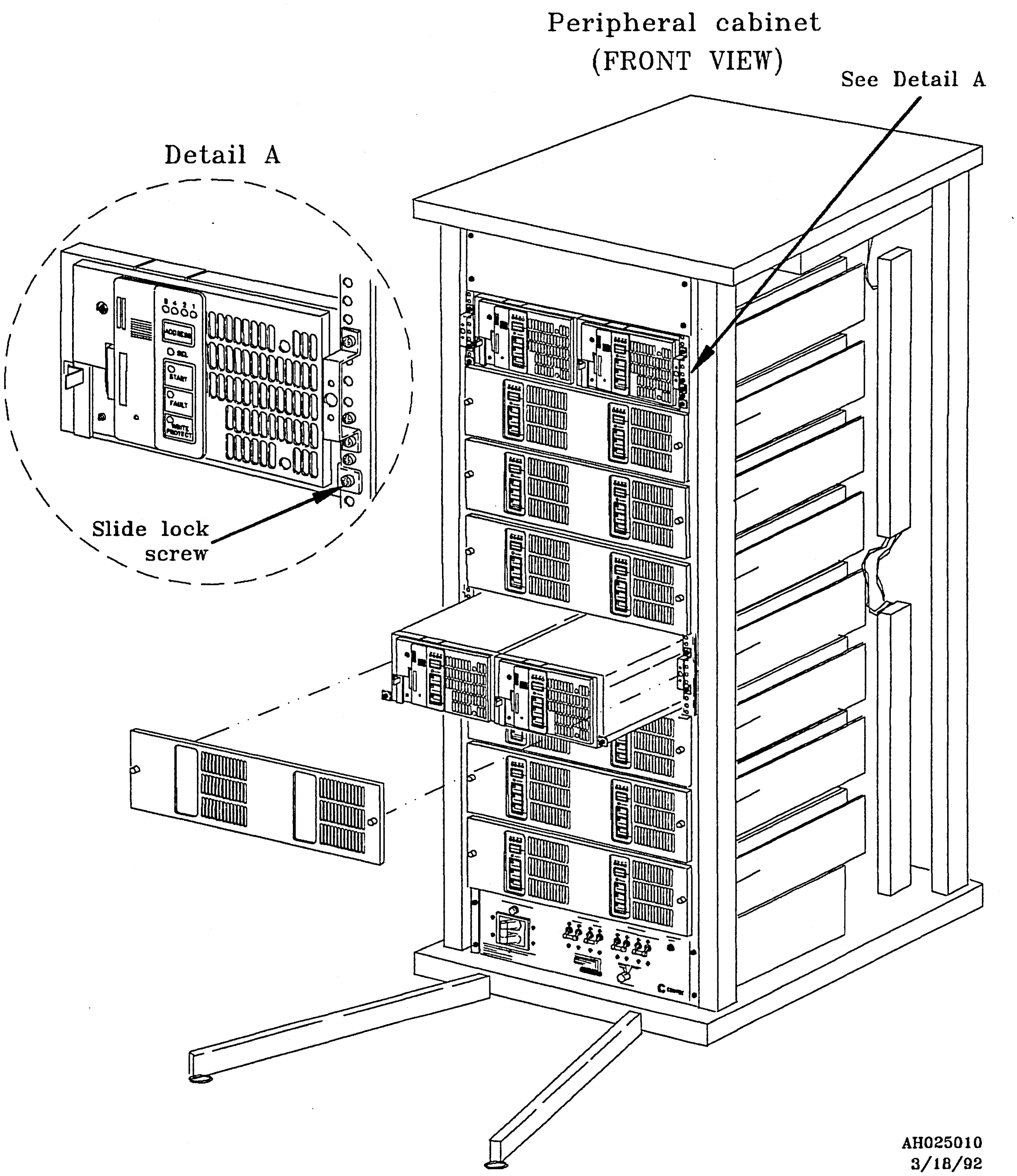
4.5 Power supply

This section defines the removal and replacement procedures for the power supply.

4.5.1 Removal

1. Complete the steps listed in Section 4.2, "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 4–11.
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.
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 4–11.
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 4–12.
8. Release the 2 front chassis captive-lock screws and extend the assembly on its slides as shown in Figure 4–12.

Figure 4–12, Front cover and slide lock



NOTE

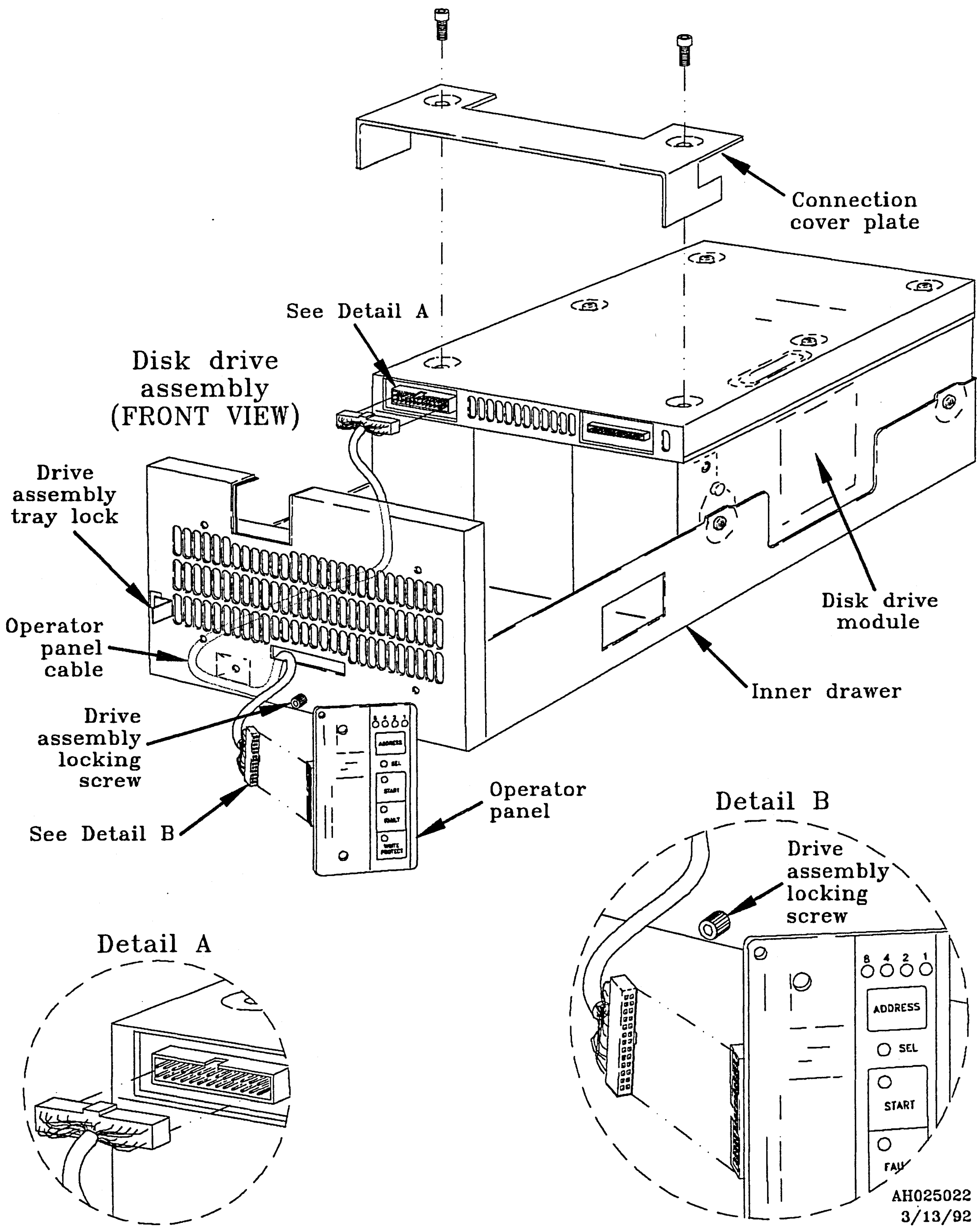
To remove the drive assembly locking screw located behind the bottom edge of the operator panel, the operator panel must be removed.

CAUTION

The operator panel can be damaged by 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 4–13.

Figure 4–13, Operator panel cable



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

WARNING

Because a drive assembly weighs 60 pounds, 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.

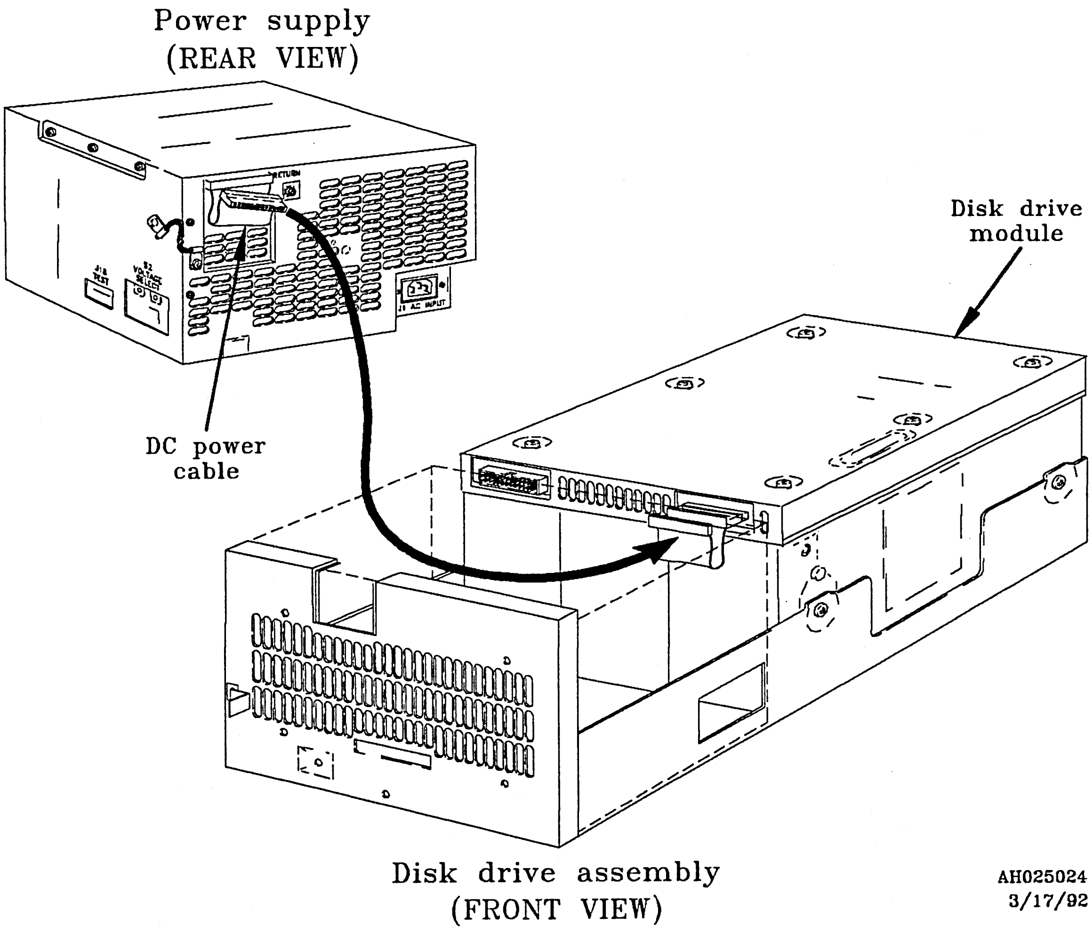
11. Press the drive assembly tray lock to the right, then lift the drive assembly tray free of the slide tray.
12. Disconnect the ac power cord from the assembly AC jack at the rear of the disk drive module.
13. Remove the 2 locking screws on the connection cover plate and remove the plate.
14. Loosen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

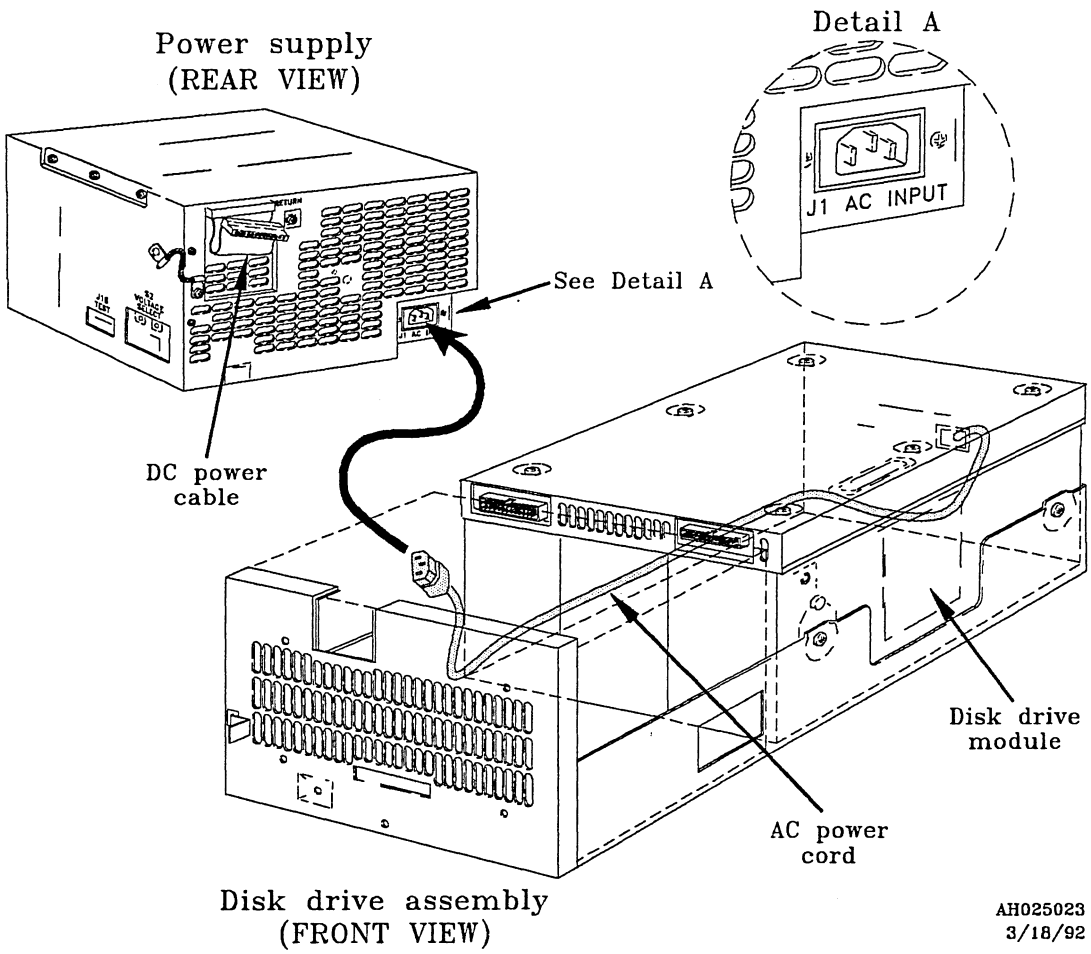
15. Lift the disk drive module until the disk drive module mounting screws are above the sides of the drive assembly tray. Then move the disk drive module to the rear of the disk assembly tray to allow access to the dc power and the operator panel cable connectors. Lower the disk drive module in the 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 4–14.

Figure 4–14, dc power cable



17. Disconnect the operator panel cable from the disk drive module as shown in Figure 4-13.
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 4–15.

Figure 4–15, 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.

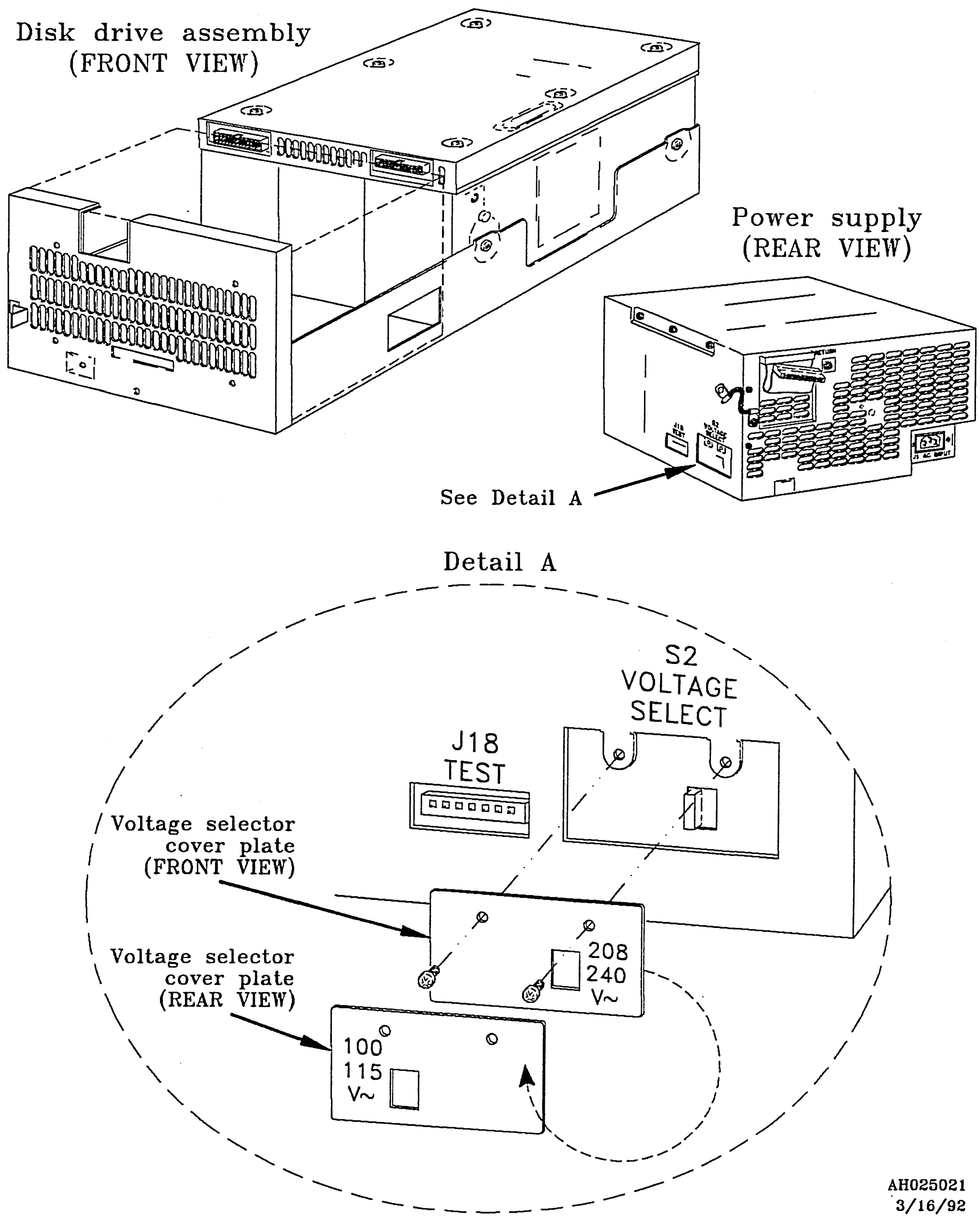
4.5.2 Replacement

1. Complete the steps listed in Section 4.2, "Maintenance safety procedures."

> **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 4–16.

Figure 4–16, Power supply voltage setting



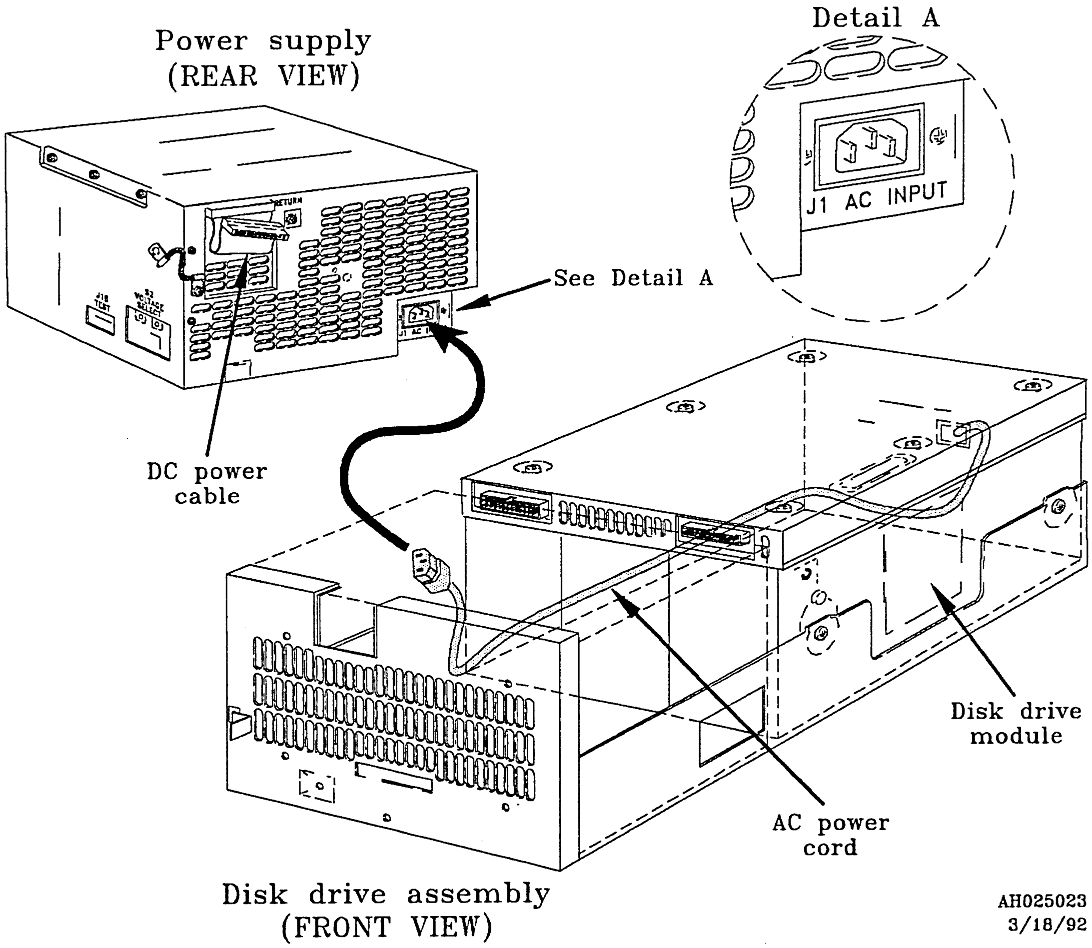
3. Set the ON/STANDBY switch on the front of the power supply to the **ON** position.

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 4–17.

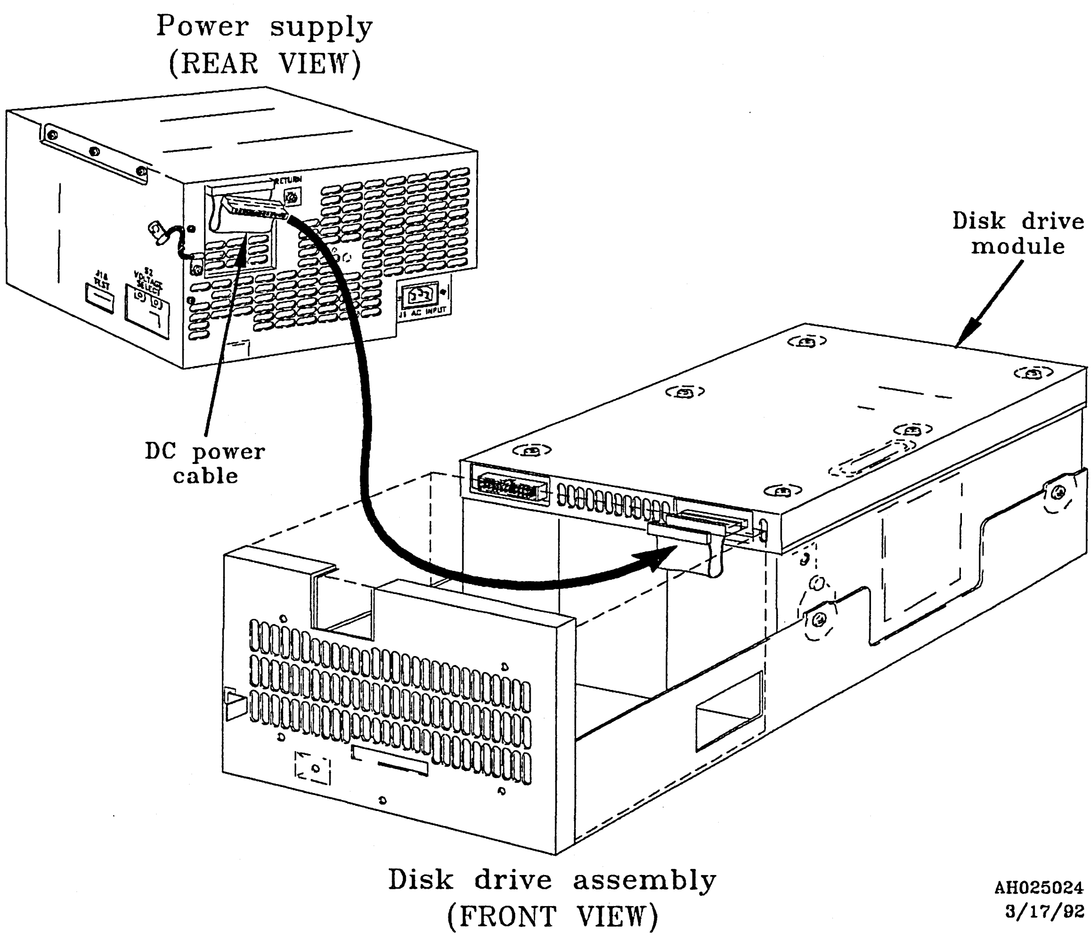
Figure 4–17, Power supply ac power cord



> **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.

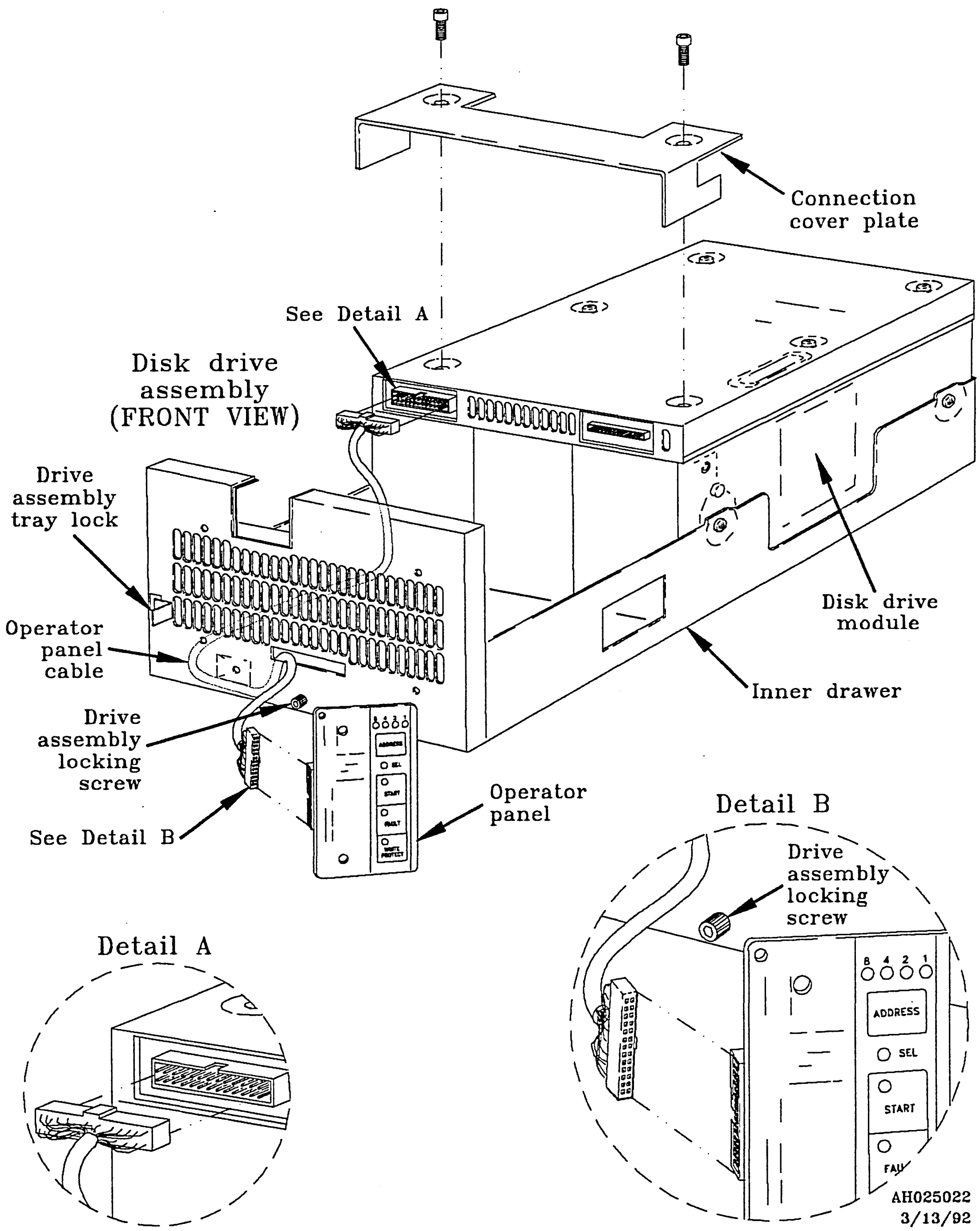
11. Lower the disk drive module into the drive assembly tray until the disk drive module mounting screws are supported by the sides of the drive assembly tray.
12. Connect the dc power cable to the disk drive module as shown in Figure 4–18.

Figure 4–18, dc power cable



13. Connect the operator panel cable to the disk drive module as shown in Figure 4–19.

Figure 4–19, 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 pounds, 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.

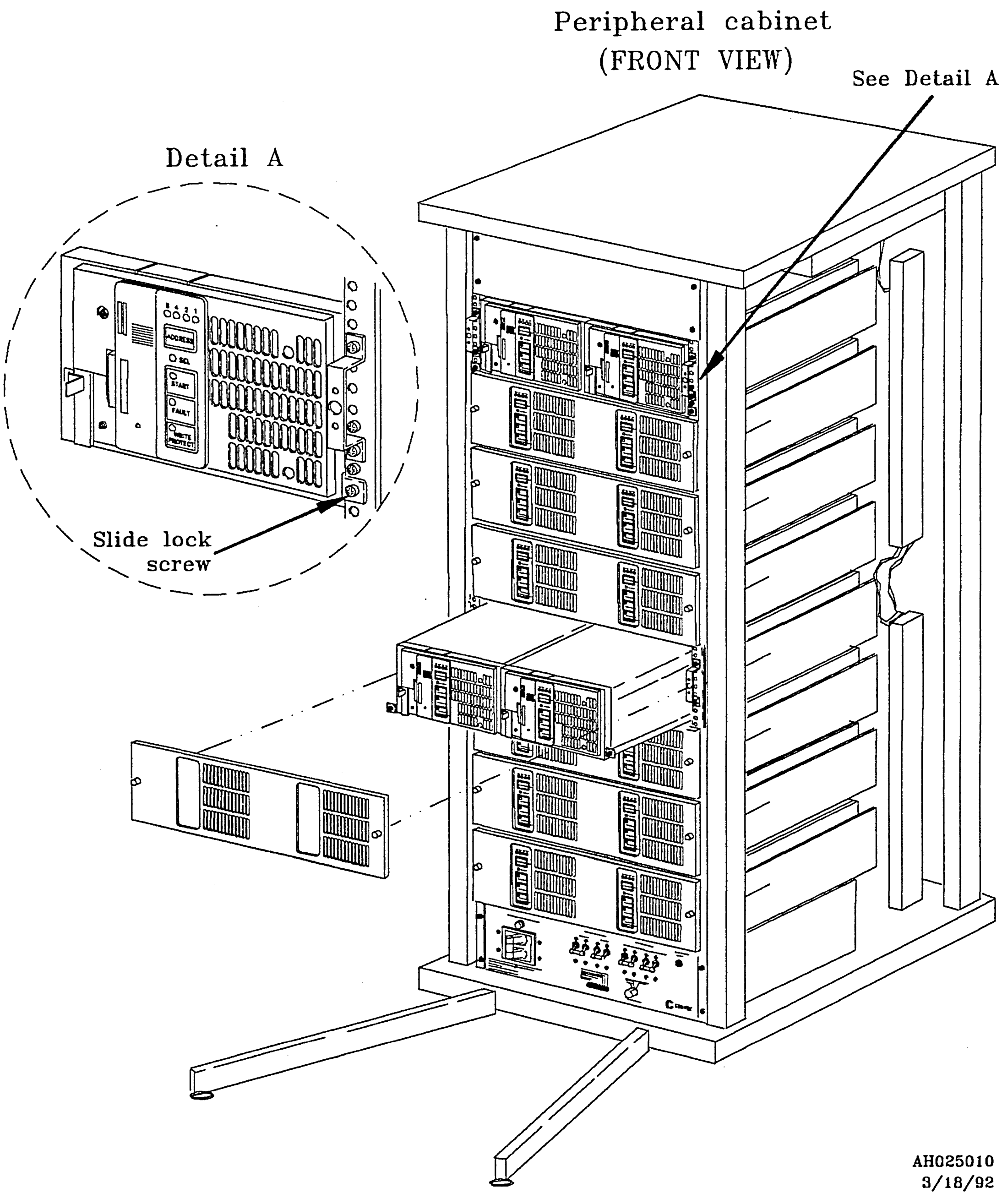
19. Press the drive assembly tray lock to the right, then lower the drive assembly tray into the slide tray.
20. Install the drive assembly locking screw located on the front of the tray.

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 4-19.
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 4–20.

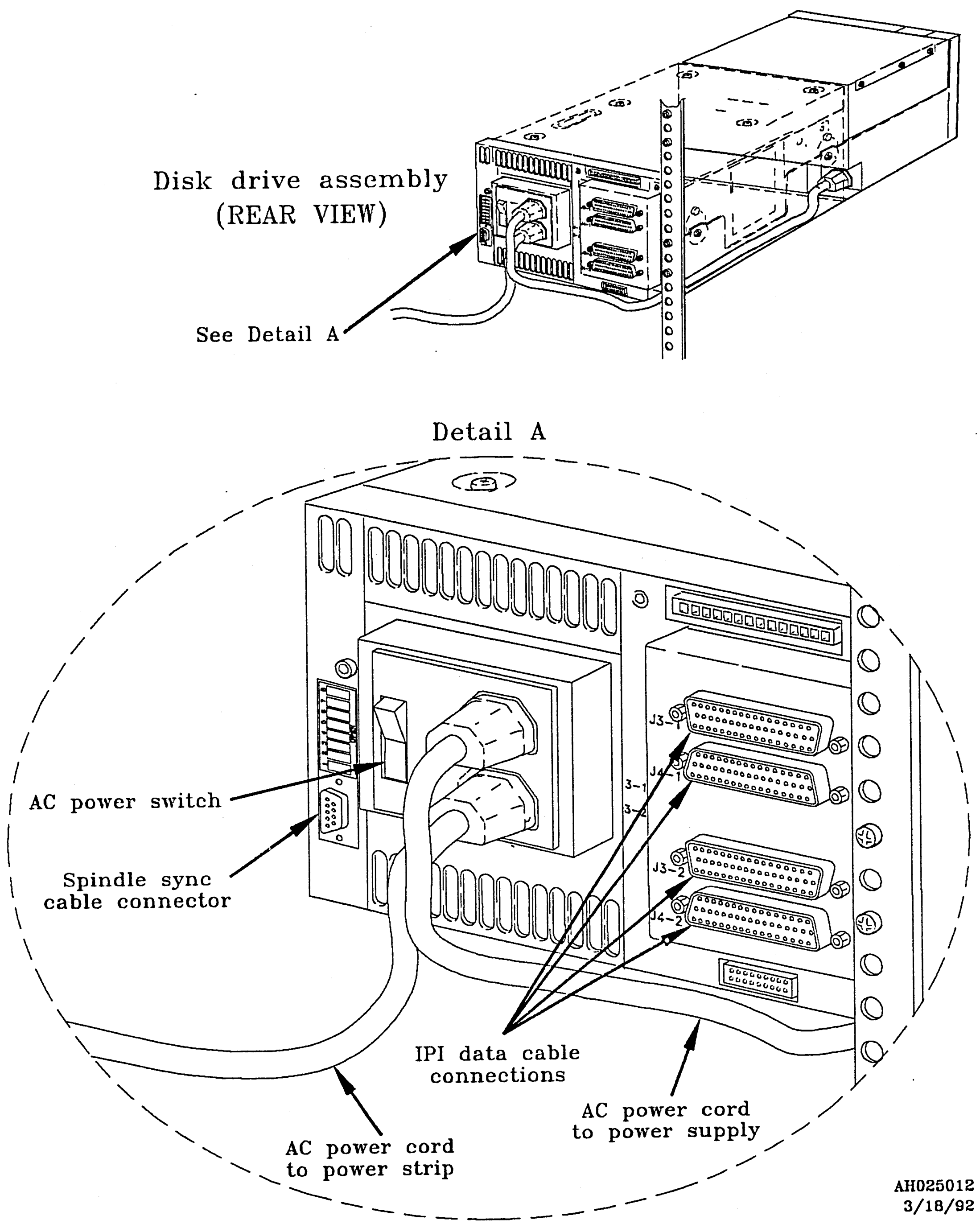
Figure 4–20, Front cover and slide lock



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 4–21.

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 4–21, 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 4–21.
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 keyswitch to the **ON** position.
32. Return the peripheral cabinet stabilizer bars to their retracted positions.

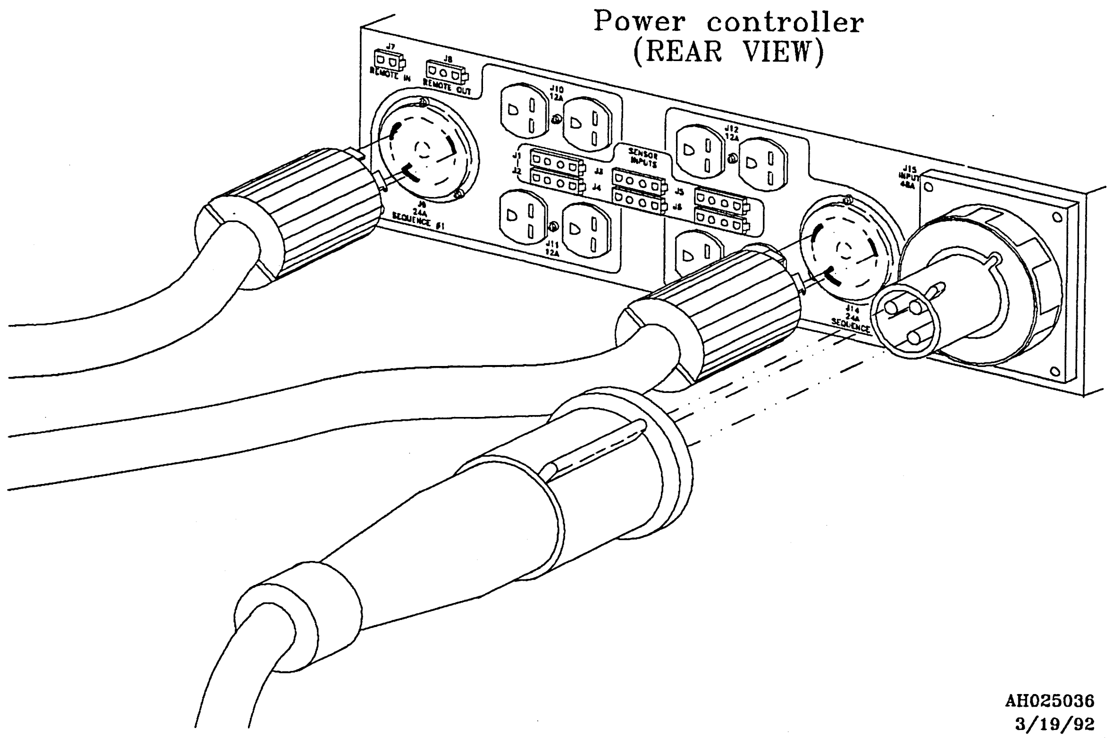
4.6 Power controller

This section defines the removal and replacement procedures for the power controller.

4.6.1 Removal

1. Complete the steps listed in Section 4.2, "Maintenance safety procedures."
2. Remove the peripheral cabinet rear panel.
3. Disconnect the power cable from the power controller AC-input jack as shown in Figure 4–22.
4. Disconnect the power strip twist-lock plugs from the twist-lock connectors.
5. Disconnect the fan sensor plugs from the **J1** and **J2** connectors.
6. Disconnect the thermostat plug from the **J3** connector.
7. Unplug and label all remaining cables and cords.
8. Remove the four screws on the front panel of the power controller.
9. Remove the power controller from the cabinet.

Figure 4–22, Power controller



4.6.2 Replacement

1. Complete the steps listed in Section 4.2, "Maintenance safety procedures."
2. Install the power controller in the cabinet.
3. Install the four screws on the front panel of the power controller.
4. Connect the power cable to the power controller AC-input jack as shown in Figure 4-30.
5. Connect the power strip twist-lock plugs to the twist-lock connectors.
6. Connect the fan sensor plugs to the **J1** and **J2** connectors.
7. Connect the thermostat plug to the **J3** connector.
8. Plug in all remaining cables and cords.
9. Install the peripheral cabinet rear panel.
10. Set the power controller's main circuit breaker to the **ON** position and select local or remote operation with the **LOCAL/REMOTE** switch.
11. Turn the processor's front control panel keyswitch to the **ON** position.
12. Return the peripheral cabinet stabilizer bars to their retracted positions.

4.7 Fan assembly

This section defines the removal and replacement procedures for the fan assembly.

4.7.1 Removal

1. 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.
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 4–23.
6. Remove the fan assembly from the rear of the cabinet.

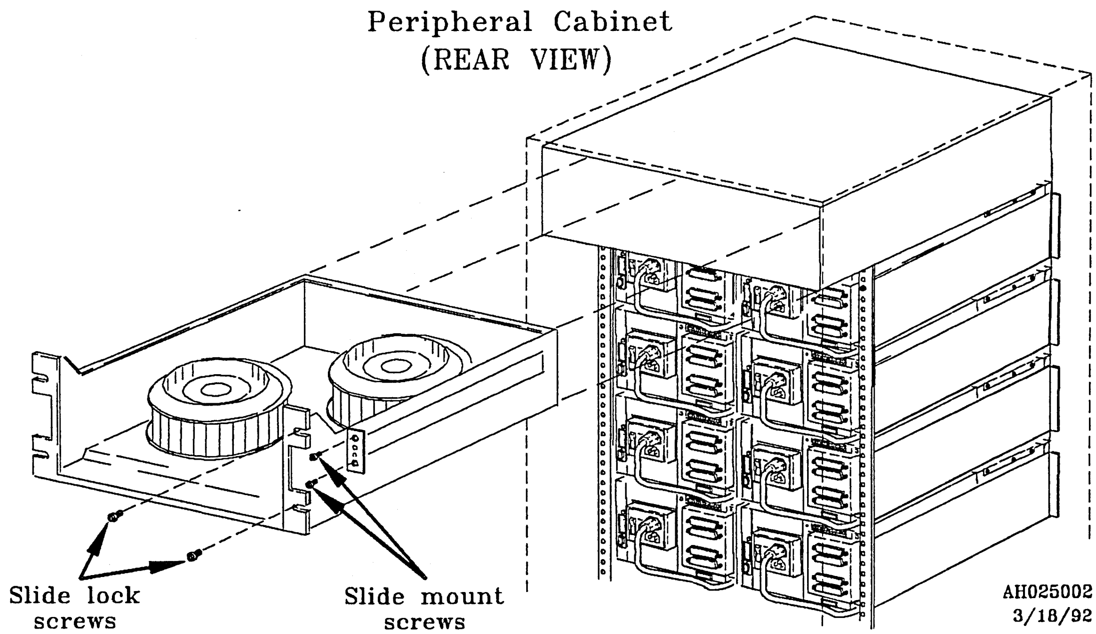
4.7.2 Replacement

1. 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.
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.

Figure 4–23, Peripheral cabinet fan assembly



4.8 Peripheral cabinet air filter

This section defines the removal and replacement procedure for the cabinet door air filters.

1. Release the 8 lock screws and remove the cabinet door air filter panels as shown in Figure 4–24.
2. Install the cabinet door air filter panels into the front door of the cabinet and secure with the 8 lock screws.

4.9 Disk drive air filter

This section defines the removal and replacement procedures for a disk drive air filter.

4.9.1 Removal

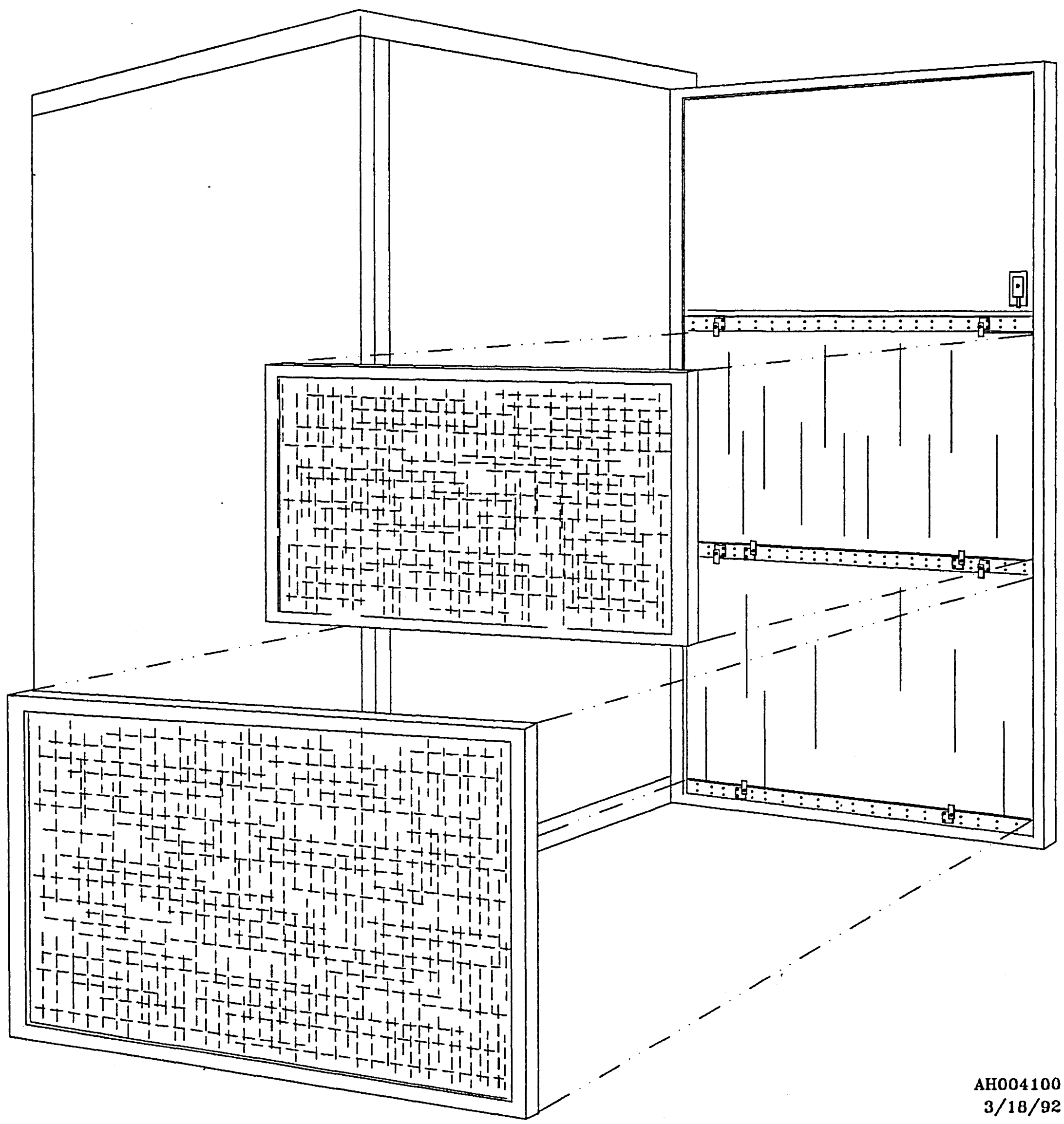
1. Extend the peripheral cabinet stabilizer bars and adjust the feet until they are in firm contact with the floor.
2. Release the 2 front cover lock screws and remove the cover.
3. Release the 2 front assembly lock screws and extend the assembly one inch on its slides.

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.

Figure 4–24, Air filter



4.9.2 Replacement

1. Extend the peripheral cabinet stabilizer bars and adjust the feet until they are in firm contact with the floor.

> **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.

4.10 Illustrated parts list

This section is the illustrated parts breakdowns (IPBs) for the IDC/ITC and peripheral equipment. Table 4–1 lists the CONVEX part numbers, descriptions, and quantities of the assemblies for an IDC/ITC 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, disk 1.15 Gbyte IPI, 1st disk	8
550-000296-201	Subsystem, disk 1.15 Gbyte IPI, 2nd disk	8
550-000245-200[1]	Disk cabinet, domestic	1
550-000244-200[1]	Disk cabinet, international	1

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

Table 4–2 lists the part numbers and descriptions for peripheral cabinet equipment as illustrated in Figures 4-25, 4-26, and 4-27.

Table 4–2, Peripheral cabinet parts list

Part number	Description	Figure number	Reference number
500-001025-200	Power controller assembly	4-25	1
253-000100-001	Fuse MDA-1/2A 250V	NA	NA
253-000100-004	Fuse MDA-2A 250V	NA	NA
500-000297-200	Fan Assembly , 230V, peripheral cabinet	4-26	1
230-000012-001	Fan, 365 CFM 230V, w/Hall effect	4-26	2
500-000296-200	Power strip assembly, 230V	4-26	3
614-000001-001	Cord, power, 3 conductor, 10 AWG, UL/CSA	4-26	4
500-000295-200	Thermostat assembly, 220V peripheral cabinet	4-27	1
605-030005-200	Cord assembly, 48A power controller, domestic	NA	NA
614-000004-002	Cord, power, 3 conductor, 6AWG, UL/CSA	NA	NA
304-000036-001	Connector, body, 60A, 2 pole 3 wire, IEC309	NA	NA
304-000038-001	Connector, plug, 60A, 2 pole 3 wire, 250V	NA	NA
605-030006-200	Cord assembly, 48A power controller, international[1]	NA	NA
614-000005-002	Cord, power, 3 conductor, 10SQMM, Harmon	NA	NA
304-000036-001	Connector, body, 60A, 2 pole 3 wire, IEC309	NA	NA

[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 4–25, Power controller

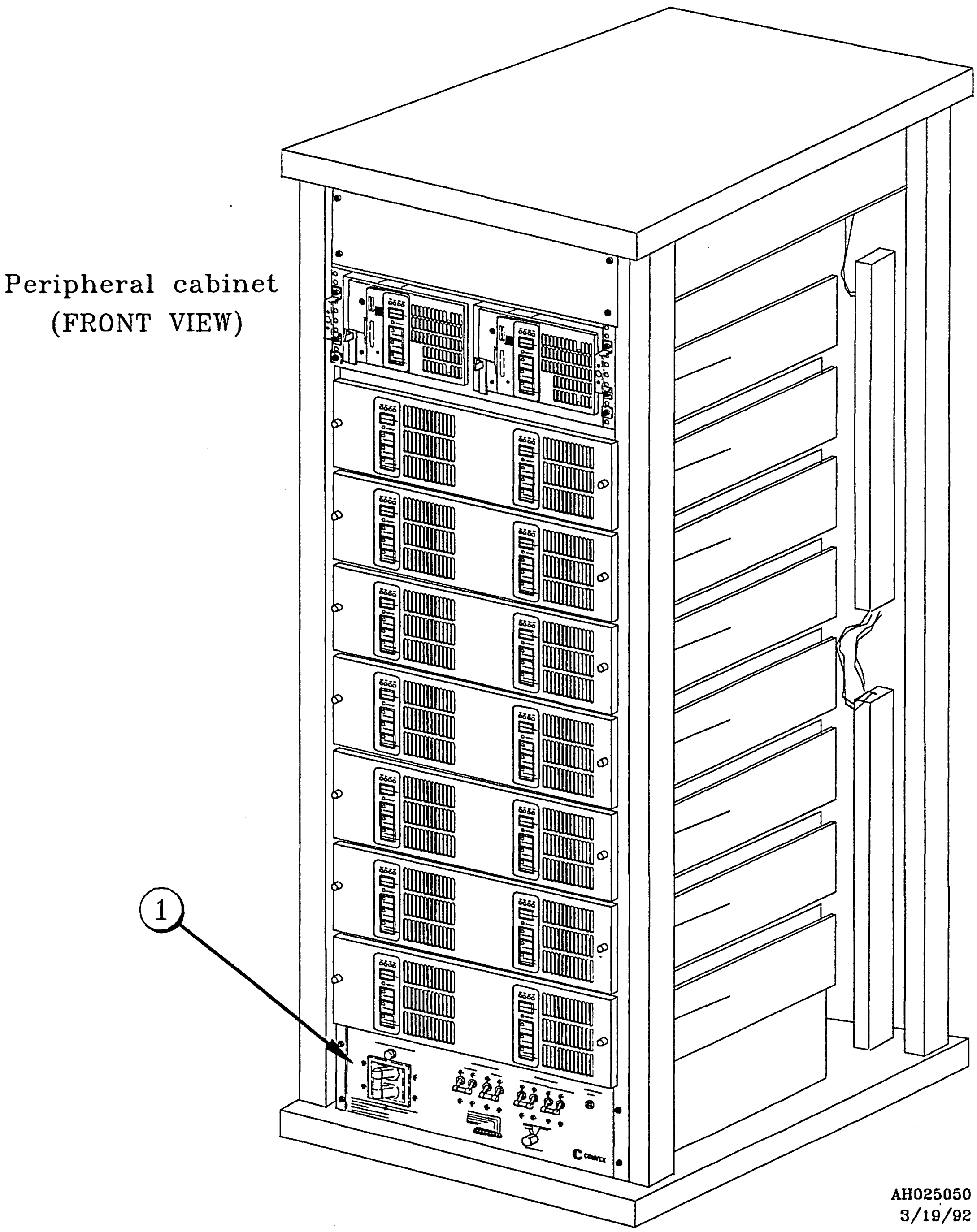


Figure 4–26, Peripheral cabinet fan assembly

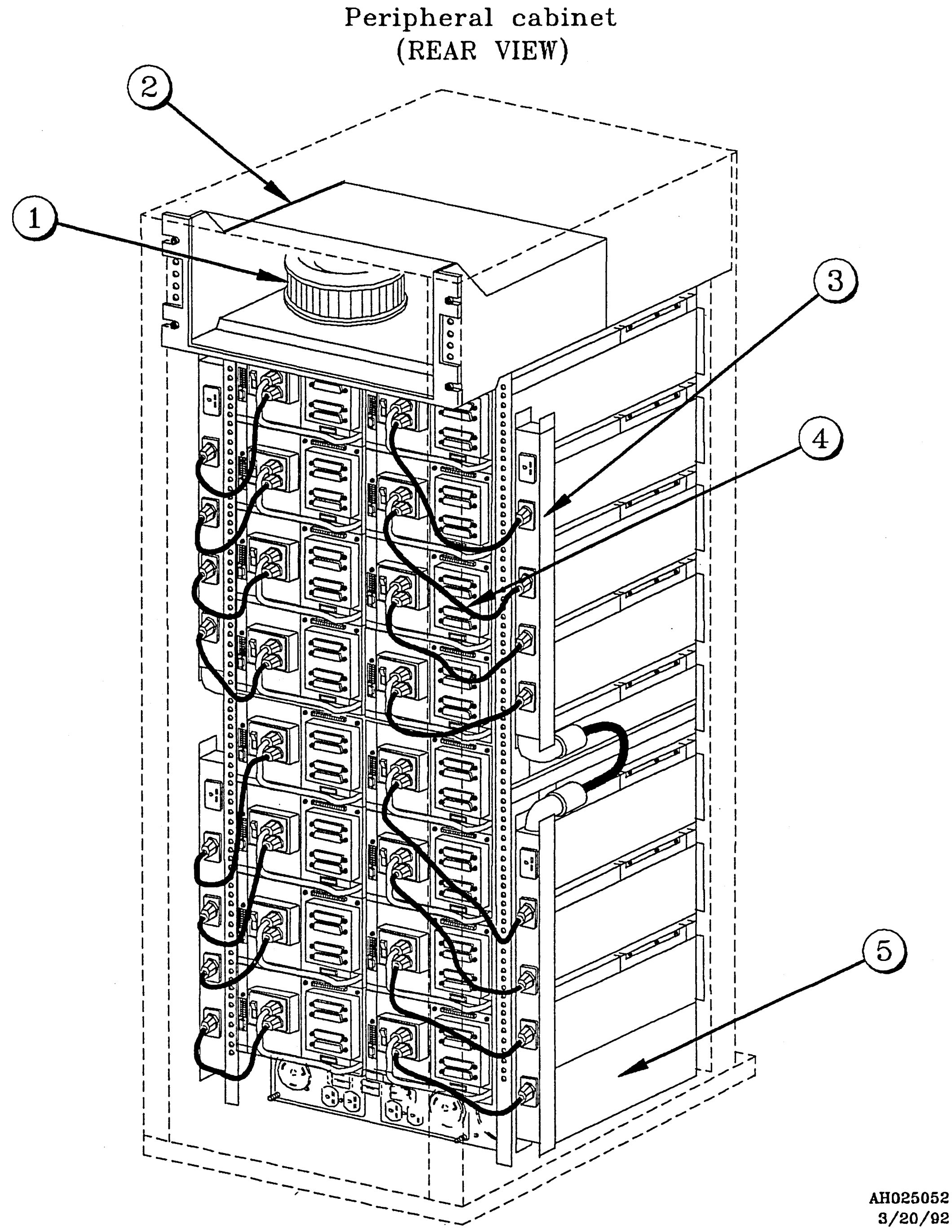


Figure 4–27, Peripheral cabinet thermostat assembly

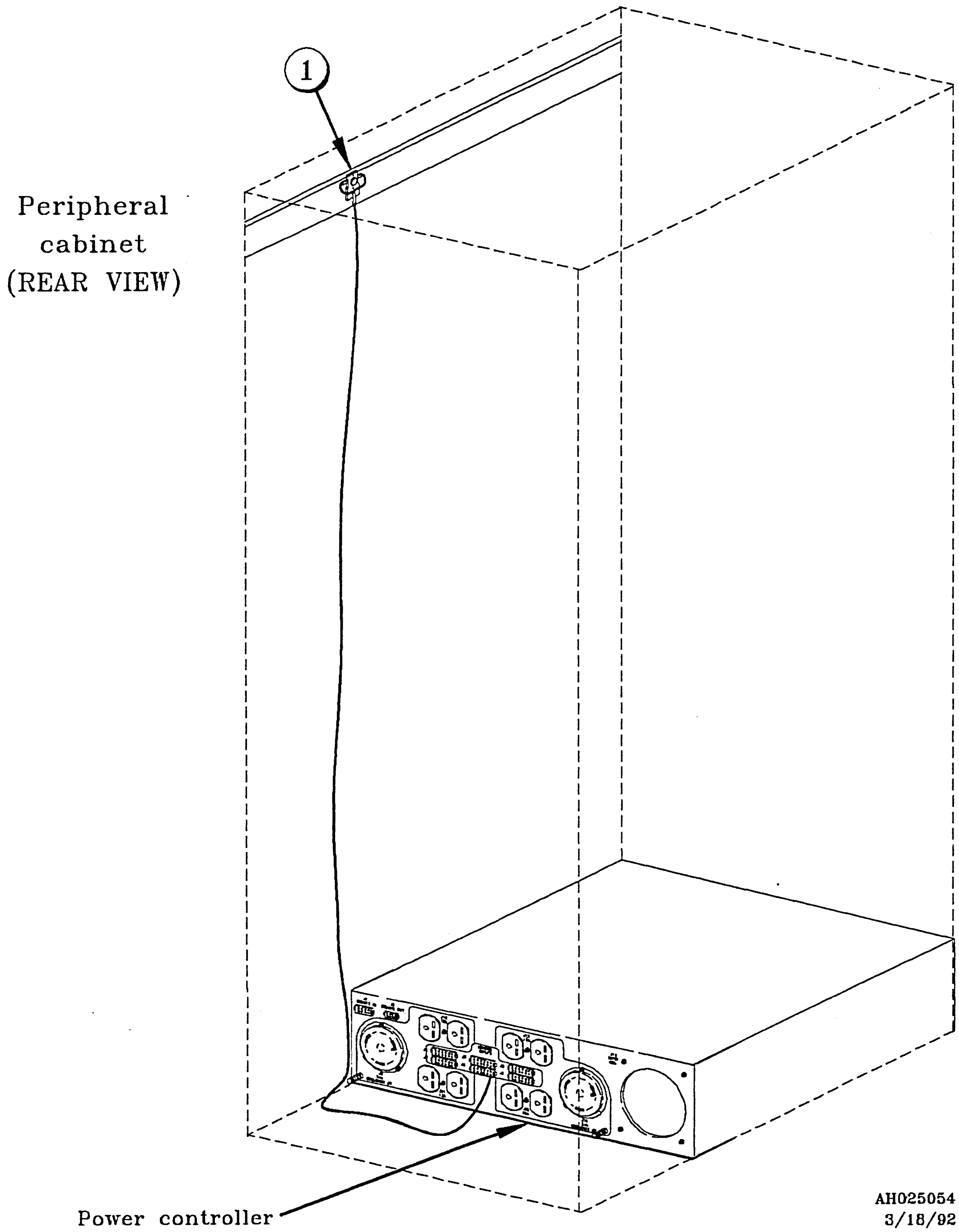


Table 4–3 lists the part numbers and descriptions for IPI-2 interface equipment as illustrated in Figures 4-28, 4-29, and 4-30.

Table 4–3, IPI-2 interface parts list

Part number	Description	Figure number	Reference number
410-001228-200	Board assembly, IDC/ITC	4-28	1
411-000202-200	Board assembly, IPI interface, top	4-29	1
411-000201-200	Board assembly, IPI interface, standard	4-29	2
601-640007-200	Cable assembly, IPI interface, 60/60	4-29	3
601-640006-200	Cable assembly, IPI interface, 60/40	4-29	4
601-640005-200	Cable assembly, IPI interface, standard	4-29	5
601-640005-201	Cable assembly, IPI interface, 3 key	4-29	6
320-000299-500	Connection plate, 50 POS, D shell	4-29	7
310-002301-001	Screw, 6-32 x 3/8, PHH, PNH, SEM	NA	NA
312-000124-002	Screw lock, 4-40 x 1/4, D connector, female	NA	NA
204-000015-006	Cable, sync spindle, 1.5 ft	4-30	1
204-000015-005	Cable, sync spindle, 3 ft	4-30	1
204-000015-001	Cable, sync spindle, 6 ft	4-30	1
204-000015-009	Cable, sync spindle, 10 ft	4-30	1
204-000015-010	Cable, sync spindle, 25 ft	4-30	1
604-500007-001	Cable, IPI data, 1.5 ft	4-30	2
604-500007-010	Cable, IPI data, 10 ft	4-30	3
604-500007-015	Cable, IPI data, 15 ft	4-30	3
604-500007-025	Cable, IPI data, 25 ft	4-30	3
604-500007-050	Cable, IPI data, 50 ft	4-30	3
204-000016-015	Terminator, sync spindle	4-30	4
204-000016-012	Terminator, IPI data	4-30	5

Figure 4–28, IDC/ITC board assembly

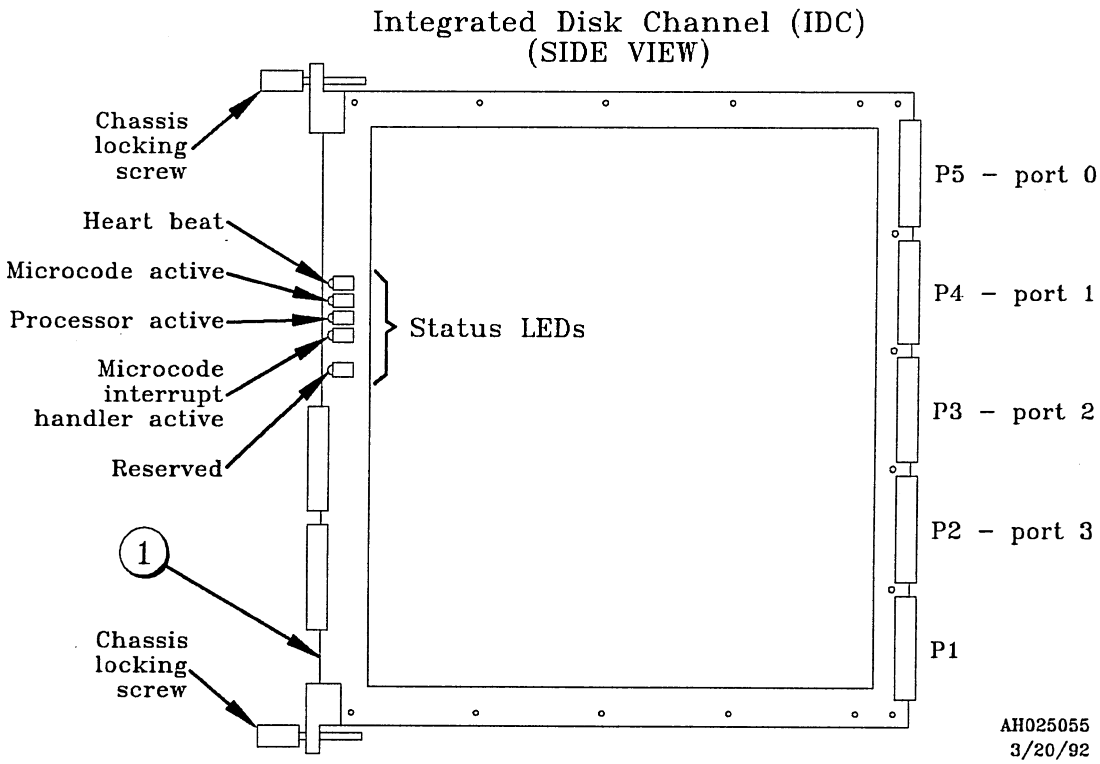


Figure 4–29, IPI interface cabling

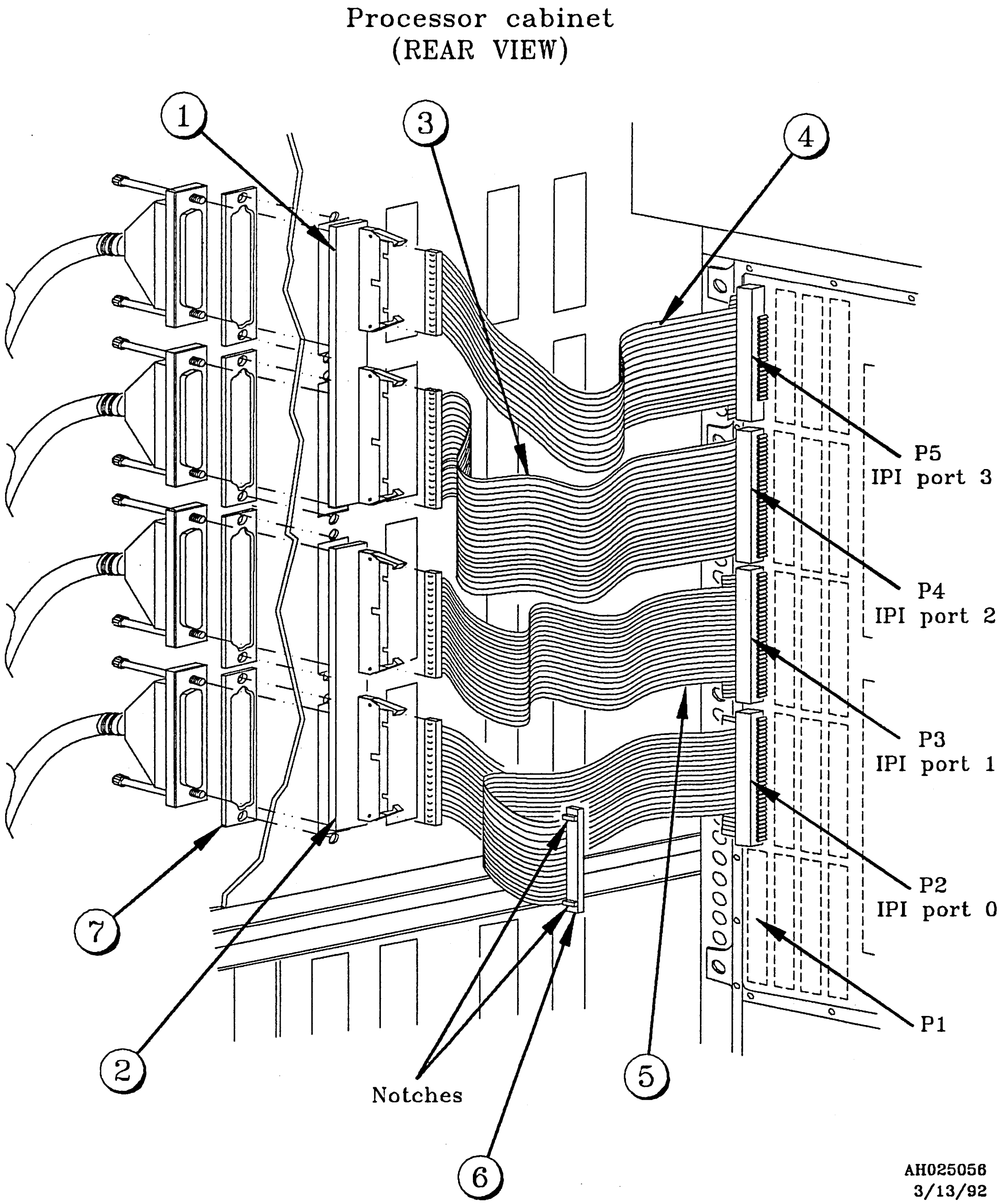


Figure 4–30, Peripheral cabinet drive cabling

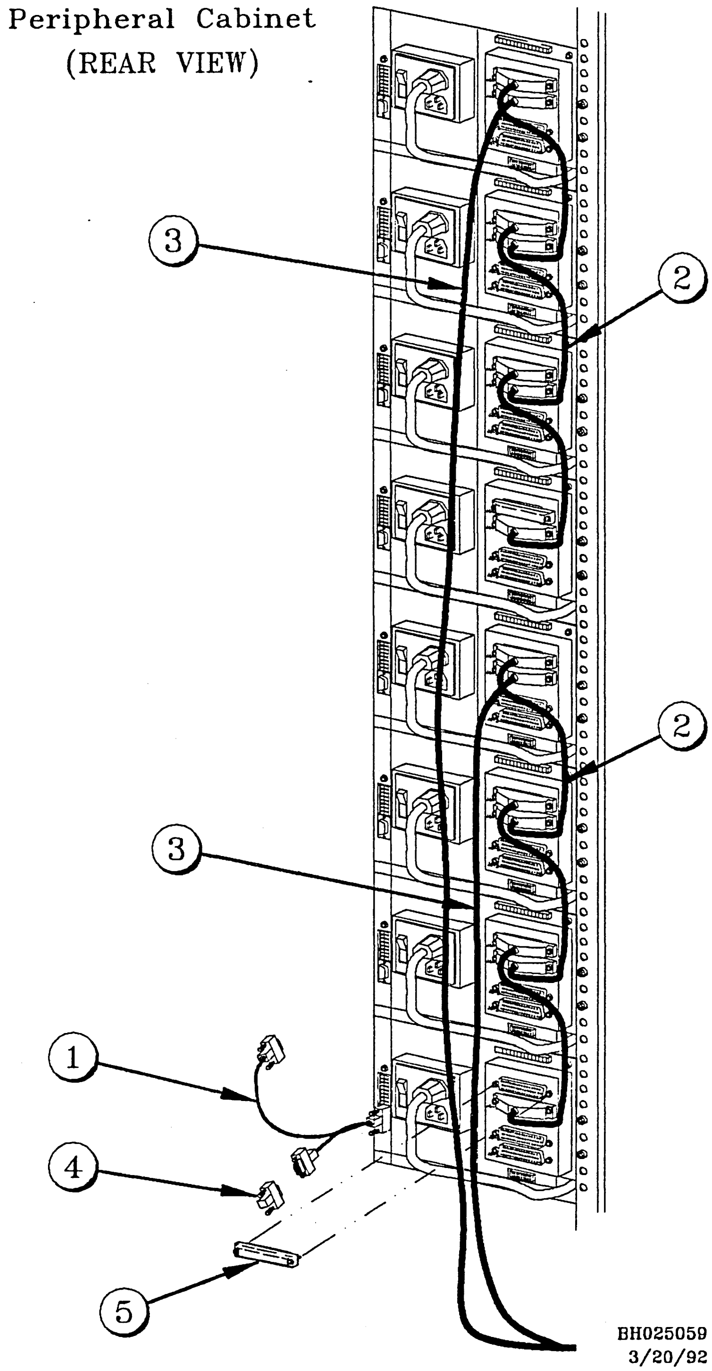


Table 4–4 lists the part numbers and descriptions for a disk drive assembly as illustrated in Figures 4-31 and 4-32.

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/s	4-31	1
204-000018-200	Disk drive, IPI-2 1.23 Gbyte 3MB/s	4-31	1
204-000015-200	Disk drive, IPI-2 1.15 GB 6MB/s	4-32	1
204-000017-200	Disk drive, 1.23 GB 3MB/s	4-32	1
204-000016-009	Drawer, inner, 1.15 Gbyte disk drive	4-32	2
204-000016-010	Power supply, IPI-2HP disk drive	4-32	3
204-000016-008	Cable, dc power, 2.5 in.	4-32	4
204-000016-007	Cable, power, 3 conductor, shielded	4-32	5
320-001464-500	Mfab, control panel	4-32	6
900-000340-001	Screw, 10-32 x 3/8, shock mount	4-32	7
312-000318-001	Filter, air, 8 in. disk	4-32	8
204-000016-011	Panel, operator	4-32	9
204-000016-004	Air baffle, 1.15 Gbyte disk drive	4-32	10
900-000553-001	Cable, operator panel	4-32	11
900-000345-001	Fan, 24 VDC, disk drive	NA	NA
204-000016-005	Clamp, power cable	NA	NA
204-000016-002	Screw, 6-32 x 1/4, hex	NA	NA
204-000016-003	Screw, 8-32 x 1/2, hex	NA	NA
204-000016-006	Washer, lock #6	NA	NA
204-000016-001	Cable, dc ground	NA	NA
204-000016-014	Ground straps kit, shock mount	NA	NA
250-000011-001	Switch, power connector	NA	NA
900-000341-001	Panel, status/control, disk drive (optional)	NA	NA
900-000344-001	Input/output board, disk drive	NA	NA
900-000346-001	Control board, disk drive	NA	NA
900-000421-001	Manual, diagrams, all Sabre 5	NA	NA

Figure 4–31, Disk drive subsystem assembly

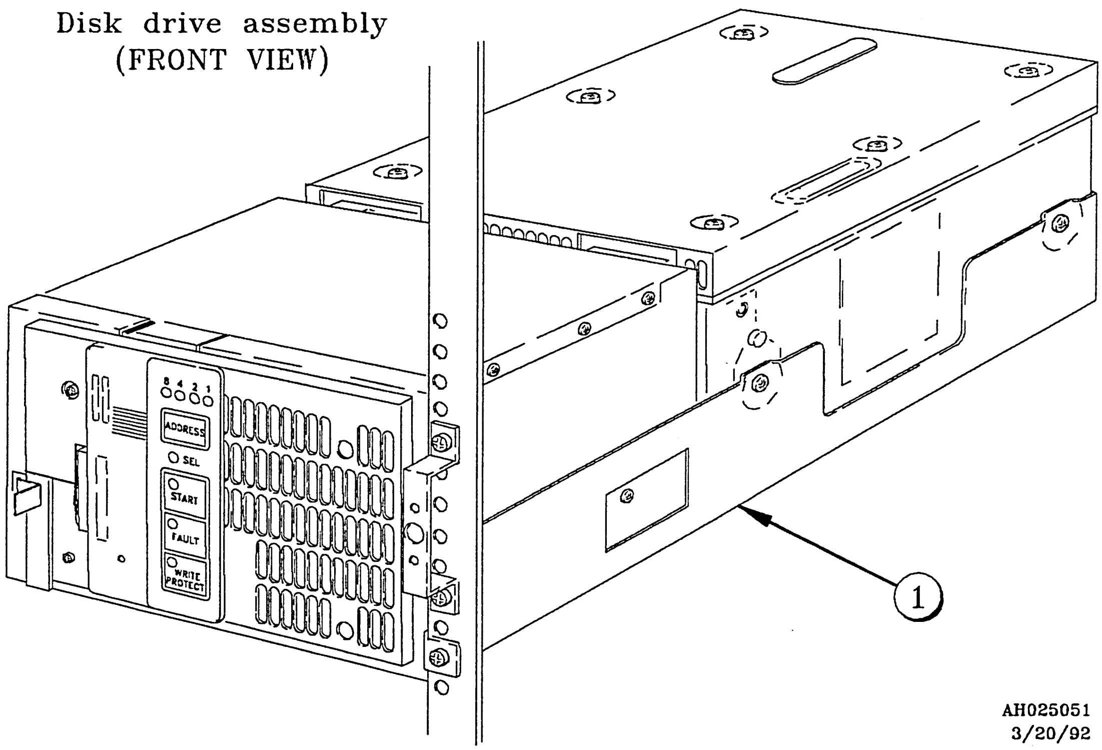


Figure 4–32, Disk drive assembly

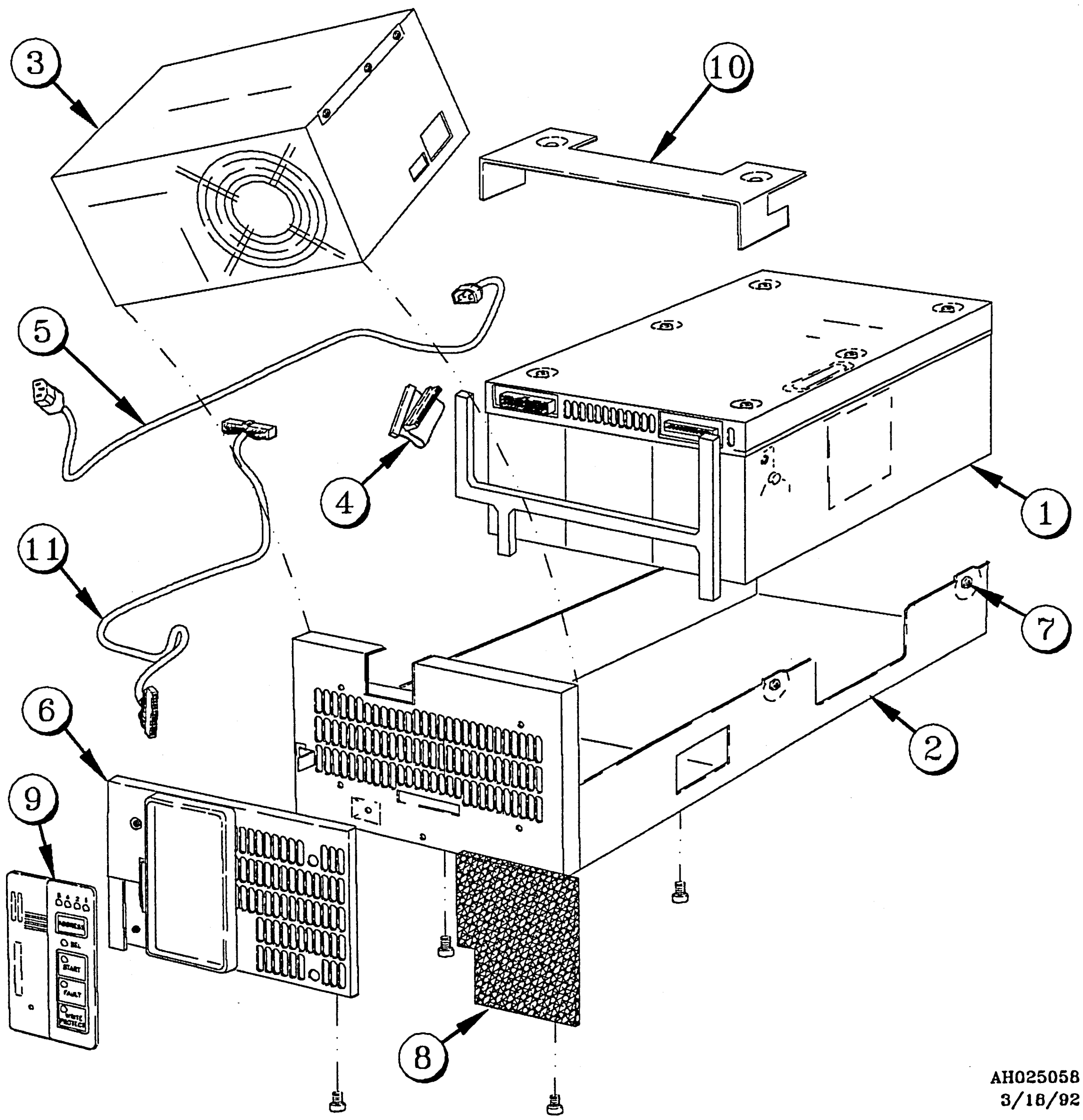


Table 4–5 lists the part numbers and descriptions for a disk drive mount assembly:

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

Index

A

B

C

D

E

F

H

I

J

L

M

N

O

P

R

S

T

V

W