# CONVEX VMEbus Tape Controller
# Service Guide

Document No. 081-001330-200

First Edition
January 1989

CONVEX Computer Corporation
Richardson, Texas USA

*CONVEX VMEbus Tape Controller*
*Service Guide*
DHW-054
First Edition



Printed in the United States of America

# Revision Sheet

*CONVEX VMEbus Tape Controller*
*Service Guide*

| Edition | Document No. | Date | Description |
|---|---|---|---|
| First | 081-001330-200 | January 1989 | First release. |

## FCC NOTICE

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

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

Warning: This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in strict accordance with the instructions manual, may cause interference to radio communications.

# Table of Contents

# Appendixes

# List of Tables

# List of Figures

# Preface

## Purpose and Intended Audience

This service guide provides a general description of the VMEbus Tape Controller (VBTC). The following areas are described:

- VBTC operation
- VBTC configuration
- VBTC installation
- VBTC integration into the CONVEX Operating System (OS)

This service guide is intended for CONVEX field engineers, manufacturing personnel, and customers installing and maintaining this equipment. This service guide is a subset of the *CONVEX VME Service Documentation.*

## Scope

The information presented in this service guide applies exclusively to the VBTC and the specified tape drives. The VBTC can be used with all CONVEX computers that have a VMEbus chassis installed.

## Outline

The content of each chapter is outlined below:

**Chapter 1. General Description**—Contains a general description of the VBTC hardware, specifications, and part numbers

**Chapter 2. VBTC Operation**—Contains VBTC operational descriptions and functional block diagram

**Chapter 3. Configuration and Installation**—Describes unpacking, damage inspection, VBTC jumper configuration, and VBTC installation procedures

**Appendix A. Glossary**—Contains an alphabetical listing of CONVEX-preferred technical nomenclature, including standard abbreviations

**Appendix B. Pending Level Registers**—Contains a layout of the pending level registers and descriptions of register contents

**Appendix C. Execution Level Registers**—Contains a layout of the execution level registers and descriptions of register contents

**Appendix D. Error Mux Status Bytes**—Contains a layout of the error mux status bytes for the STC and Fujitsu formatters

**Appendix E. VBTC Configurator Document**—Contains a copy of the VMEbus Tape Controller Configurator document

**Appendix F. Problem Reporting**—Contains information on using the *contact* facility for reporting problems

# Notational Conventions

The notational conventions used in this text are listed below:

- Bit numbering is left to right, `N-1` 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*.
- Individual bit positions within a register are denoted by specific positions separated by commas. For example, REG<15,4,0> denotes bits 15, 4, and 0 of REG.
- Byte numbering is from left to right
- A *bit* is a single binary value or entity
- A *nibble* is 4 bits
- A *byte* is 8 bits
- A *word* is 32 bits
- A bit is *set* when it contains a binary value of `1`
- A bit is *clear* when it contains a binary value of `0`
- All memory and I/O addresses are written in hexadecimal notation unless explicitly stated otherwise
- All register contents are written in hexadecimal notation unless explicitly stated otherwise
- A *register* is a programmer-visible hardware storage element internal to the controller
- The symbol *K* is an abbreviation for *kilo* or 1,024
- *Reserved* or *undefined* convey what to expect, if anything, from unused fields in registers, reserved memory, or reserved I/O space. Algorithm implementation based on the use of undefined or reserved fields is not recommended.

# Warnings

The following is an example of a warning and its typical content and location as used in CONVEX documents:

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

# Cautions

The following is an example of a caution and its typical content and location as used in CONVEX documents:

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

# Notes

The following is an example of a note and its typical content and location as used in CONVEX documents:

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

# Associated Documents

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

- *CONVEX VME Service Documentation,* Product No. DHW-050.
- *Fujitsu 243XL/R Magnetic Tape Subsystem Installation Manual.* The CONVEX part number for this manual is 900-000280-000.
- *Fujitsu 243XL/R Magnetic Tape Subsystem Maintenance Manual.* The CONVEX part number for this manual is 900-000275-000.
- *STC 2920 Maintenance Manual.* The CONVEX part numbers for this manual set are 081-000722-200 and 081-000622-200.
- *STC 1960 Maintenance Manual.* The CONVEX part numbers for this manual set are 900-000253-001, 900-000253-002 and 900-000253-003.
- *VBTC Configurator.* The CONVEX part number for this document is 410-001152-600.

# Ordering Documentation

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

The product number for this manual is DHW-054.
The document number for this manual is 081-001330-200

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

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

# Technical Assistance

Hardware and software support can be obtained through the CONVEX Technical Assistance Center (TAC). In Texas, the TAC can be reached by calling (214)952-0379. From other locations in the continental United States, call 1(800)952-0379. Customers outside the United States should contact their local CONVEX office.

# Electronic Mail

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

```
cnvxhwdoc@convex.COM
```

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

# What to Include in an Email Message

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

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

# Reader's Forum

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

THIS PAGE INTENTIONALLY LEFT BLANK

# Acknowledgments

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

- Technical contributors: Rob Carruthers, Dave Gillam, and Gene Quickel
- Document review team: Rob Carruthers, Ron Engelking, Art Fischman, Dave Gillam, Tom McClendon, Gene Quickel, and Chip Stroup
- Hardware Documentation staff: Larry Bonura and Jimmie Holman

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

C.D. Baugh, Lead Writer
CONVEX Hardware Documentation

THIS PAGE INTENTIONALLY LEFT BLANK

# Chapter 1
# General Description

## 1.1 Overview

This chapter describes the VMEbus Tape Controller (VBTC) under the following major headings:

- VBTC Overview
- VMEbus Interface
- Tape Formatter Interface
- VBTC Diagnostics
- VBTC Hardware Layout
- VBTC Specifications
- VBTC Part Numbers

> **NOTE**
>
> Refer to the *CONVEX VME Service Documentation* for additional information on the operation of the CONVEX VMEbus.

## 1.2 VBTC Overview

The VBTC is a peripheral controller that interfaces STC 2921, 2922, and 1968 tape drives and Fujitsu 2436 formatter tape drives to the CONVEX VMEbus. The VBTC functions as a high-performance Direct Memory Access (DMA) channel using minimal on-board intelligence. The VBTC relies on the intelligence in the VMEbus Input/Output Processor (VIOP) for control information, error recovery and retry operations.

The main function of the VBTC is interfacing asynchronous VMEbus 16-bit words to the asynchronous 9-bit (8 data bits and a parity bit) I/O buses of the tape formatters listed above. The VBTC supports standard blocked tape formats and reads and writes gapless SEG-D format tapes.

The VBTC has two modes of operation: *normal blocked mode* and *chain mode*. Normal blocked mode is used for tape transfers where interblock gaps are expected during tape reads and generated during tape writes. Chain mode is used when very long records (greater than 64 Kbytes) are read or written.

The VIOP downloads a byte count to the VBTC in both normal blocked and chain modes; reading or writing continues until the byte count is reached or the data is exhausted. The maximum block size that can be read and written in blocked mode is $2^{24}$ bytes. The VBTC

permits chaining of byte counts, allowing unknown length records to be read, and very long records to be written.

## 1.3 VMEbus Interface

The VBTC is a 9U VMEbus form factor board that can be installed in all CONVEX VMEbus chassis. It uses a 24-bit address path and transmits data in 8-bit bytes or 16-bit words. The VBTC's VMEbus address, bus request level, and interrupt level are jumper-selectable.

The VBTC may be a VMEbus *master* or *slave*. When receiving control and tape drive setup information from the VIOP, the VBTC is a VMEbus slave. When writing data to memory during tape reads or reading memory data during tape writes, the VBTC is a VMEbus master.

## 1.4 Tape Formatter Interface

The VBTC connects to the tape drive formatter via two cables that are routed through the cabinet bulkhead. Tape drive commands, e.g., *read, write,* and *rewind* are passed through the VBTC to the formatters. Formatter error status is available through multiplexers on the VBTC. Errors may be detected by either the VBTC or the tape drive formatters.

Data transfers between the VBTC and the formatters are asynchronous; therefore, the data transfer rate is determined by the particular tape drive/formatter in use. When writing to tape, the VBTC *unpacks* 16-bit words from the VMEbus into 8-bit bytes and generates parity for the bytes. When reading from tape, the VBTC checks parity and *packs* successive bytes into 16-bit words for transfer over the VMEbus.

## 1.5 VBTC Diagnostics

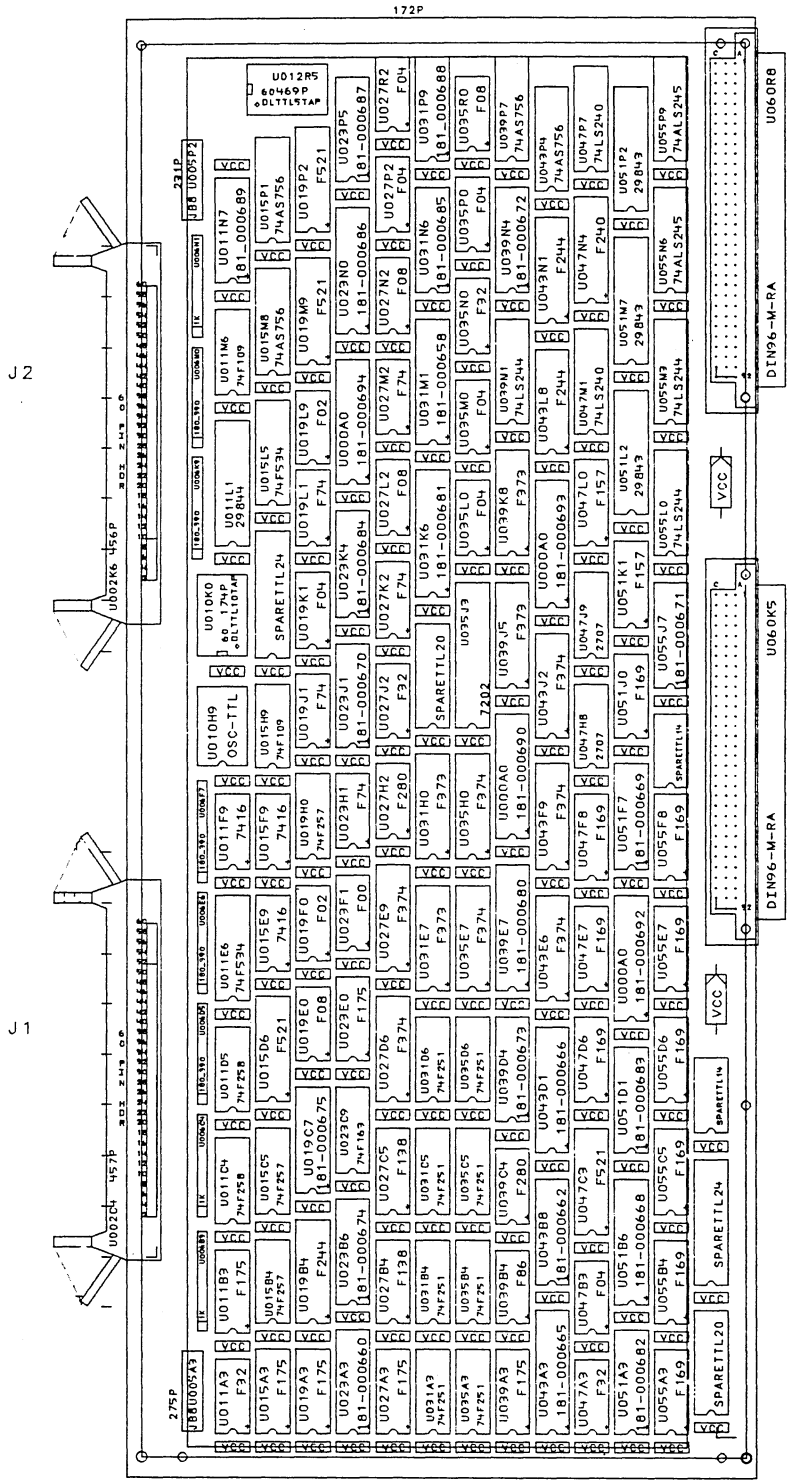
The VBTC has a diagnostic loopback mode and forced parity error mode. The diagnostic loopback mode allows data to be written to and read from the VBTC FIFO buffer; a maximum of 960 bytes may be written to the FIFO buffer in loopback mode. Connection to a tape drive is not required when performing VBTC loopback mode diagnostics.

The forced parity error mode allows the host to write data with even parity to the FIFO buffer and the attached tape drive. The tape drive reports tape data bus parity errors in this mode.

## 1.6 VBTC Hardware Layout

The main components of the VBTC are PAL control logic circuits, registers, counters, buffers, multiplexers, RAMs, and a 1,024 x 9-byte FIFO buffer. All control signals are generated by PAL state machines; there is minimum intelligence on the VBTC. The VBTC hardware layout is illustrated in Figure 1–1:

## Figure 1–1, VBTC Hardware Layout



## 1.7 VBTC Specifications

Tables 1–1 through 1–3 contain the operational, electrical, and physical specifications of the VBTC.

**Table 1–1, Operational Specifications**

| Parameter | Value(s) |
|---|---|
| Maximum Temperature Change Rate | 3.5 ° F/hour (2 ° C/hour) |
| Recommended Operating Temperature Range | 70 ° F to 80 ° F (21.0 ° C to 26.6 ° C) |
| Maximum Operating Temperature Range | 60 ° F to 90 ° F (15.5 ° C to 32 ° C) |
| Recommended Humidity Range | 40% to 60%, with no condensation |
| Maximum Humidity Range | 10% to 90%, with no condensation |

**Table 1–2, VBTC Electrical Specifications**

| Parameter | Value |
|---|---|
| Power Dissipation | 20.0 W (typical) |
| DC Voltage | 4.75 VDC to 5.25 VDC |

**Table 1–3, Physical Specifications**

| Parameter | Value |
|---|---|
| Width | 14.43 in (36.65 cm) |
| Height | 6.29 in (15.98 cm) |
| Thickness | 0.65 in *approx* (1.65 cm) |
| Weight | 3.0 lbs *approx* (1.3 kg) |

# 1.8 VBTC Part Numbers

Table 1–4 contains the CONVEX part numbers for the VBTC and connecting cables:

**Table 1–4, VBTC Part Numbers**

| CONVEX Part Number | Product Description |
|---|---|
| 410-001152-200 | VMEbus Tape Controller (VBTC) |
| | Cables (VBTC to Formatter/Tape Drive) |
| 604-600001-200 | 10 ft. |
| 604-600001-201 | 15 ft. |
| 604-600001-202 | 20 ft. |
| 604-600001-203 | 25 ft. |
| 604-600001-204 | 30 ft. |
| 604-600001-205 | 35 ft. |
| 604-600001-206 | 40 ft. |

**NOTE**

The maximum cable length for STC 2921 and 2922 drives is 40 feet. The maximum cable length for STC 1968 drive is 30 feet. The maximum cable length for the Fujitsu 2436 drive is 20 feet.

THIS PAGE INTENTIONALLY LEFT BLANK

# Chapter 2
# VBTC Operation

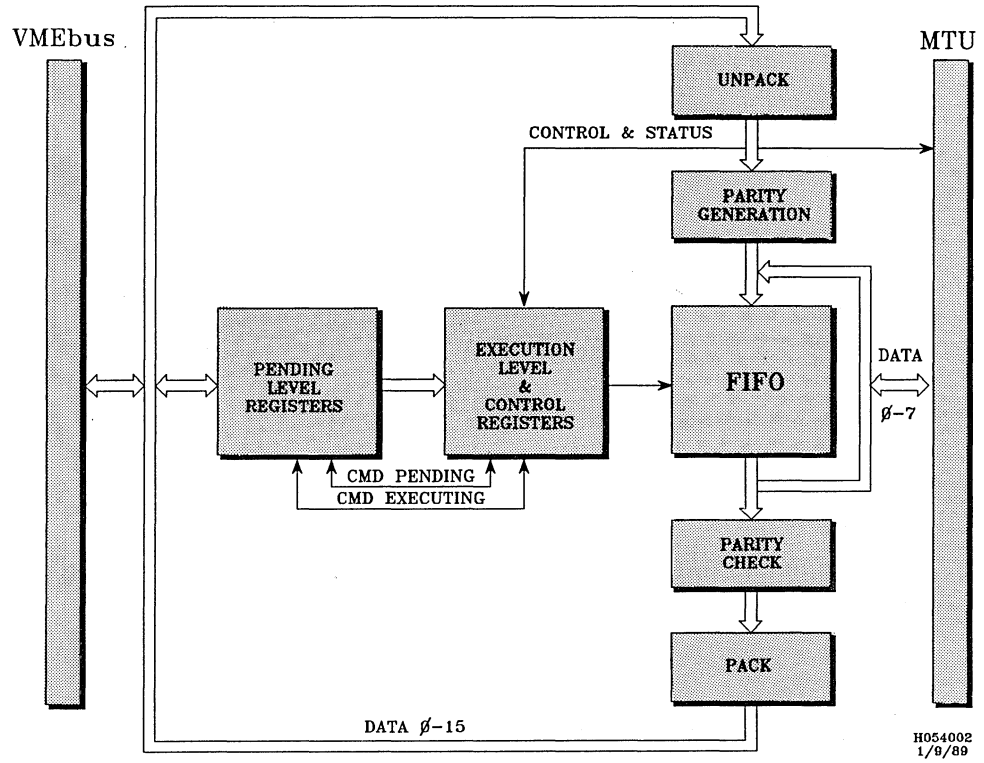
## 2.1 Overview

The main function of the VBTC is interfacing asynchronous 16-bit VMEbus words to the asynchronous 9-bit I/O bus (8 data bits and a parity bit) of the tape drive formatter. This chapter describes VBTC operation under the following major headings:

- Data Transfer
- Operation Levels
- Command Pipelining
- Normal Blocked Mode Operation
- Chain Mode Operation
- Interrupts
- Error Reporting
- Diagnostics
- VBTC Functional Block Diagram

The major functional areas of the VBTC's operation are illustrated in Figure 2–1:

**Figure 2–1, VBTC Operational Diagram**




### 2.1.1 Data Transfer

When writing data to a tape drive, the VBTC receives a 16-bit word from the VMEbus, unpacks the word into 8-bit bytes, generates parity, and loads the data into the 1,024 x 9-byte VBTC FIFO buffer. The FIFO buffer's output is controlled by the tape drive formatter, i.e., the tape drive/formatter determines the rate (speed) that data is taken from the FIFO buffer.

When reading data from a tape drive, the tape formatter outputs data (8 bits and a parity bit) to the VBTC's FIFO buffer. When the data from a tape drive exits the FIFO buffer, parity is checked, stripped, and successive bytes are packed into 16-bit words for transfer over the VMEbus.

> **NOTE**
>
> The VBTC is the VMEbus *master* when reading from, or writing to, a tape drive. All data transfers between the VBTC and the VMEbus adhere to VMEbus Specifications C.1 (IEEE Specification P1014/D1.2).

### 2.1.2 Operation Levels

The VBTC has two levels of operation: *pending level* and *execution level.* Operations begin with the VBTC as a VMEbus *slave* and the VIOP loading data into the VBTC pending level registers. The pending level registers contain operation control information, tape drive setup information, and the transfer address and count. Writing any value to the start flag in the pending level registers signals the VBTC that loading is complete. The command pending flag is set if a command is currently executing.

If no command is being executed, the contents of the pending level registers are copied into the execution level registers. This process clears the command pending flag (if set) and sets the command executing flag. The pending level registers can then be loaded with the next command.

Table 2–1 shows the contents of the pending level registers. Refer to Appendix B for a complete description of the pending level registers.

**Table 2–1, Pending Level Register Layout**

| Byte Number | Register Contents |
|---|---|
| 0 | Start Flag |
| 1 | Control |
| 2 | Tape Unit Select |
| 3 | Density Select |
| 4 | Command Select |
| 5 | Transfer Count |
| 6 | |
| 7 | |
| 8 | Buffer A Transfer Address |
| 9 | |
| A | |
| B | Clear Interrupt |
| C | Buffer B Transfer Address |
| D | |
| E | |
| F | Reset |

Once the command executing flag in the execution level registers is set, the information copied from the pending level registers to the execution level registers is available to the various internal state machines on the VBTC. When these internal state machines begin receiving execution level register data, VBTC operations become independent of the VIOP. These on-board VBTC operations include the tape drive setup, tape reads and writes in blocked and chain modes, VMEbus operations, error reporting, and diagnostics.

Table 2–2 shows the contents of the execution level registers. Refer to Appendix C for a complete description of the execution level registers.

**Table 2–2, Execution Level Register Layout**

| Byte Number | Register Contents |
|---|---|
| 10 | Status |
| 11 | Transfer Address Status |
| 12 | |
| 13 | |
| 14 | Transfer Count Status |
| 15 | |
| 16 | |
| 17 | Command Status |
| 18 | Density and Error Status |
| 19 | Formatter Error Status |
| 1A | Real Time Drive Status |
| 1B | Reserved |
| 1C | Residual Transfer Count |
| 1D | |
| 1E | |
| 1F | Reserved |

### 2.1.3 Command Pipelining

In order to maximize the efficiency of the tape drives/formatters, the VBTC relays tape drive commands using a *pipelining* technique. Pipelining refers to the process of queueing and transmitting commands in a near continuous manner. This rapid relaying of commands greatly reduces or eliminates the time delays that are caused by the tape drive starting and stopping at tape inter-block gaps, allowing the tape to move through them at maximum velocity.

Pipelining commands to the controller provides the ability to meet the re-instruct time of the tape drive and allows a new command to be issued very quickly.

The basic process of VBTC command pipelining is outlined below:

1. A command is loaded into the pending level registers.
2. When the start flag (in the pending level registers) is written, the command in the pending level registers is transferred to the execution level registers. After the transfer is completed, the next command (if there is one) is loaded into the pending level registers.
3. The start flag is written a second time (without waiting for the first command to complete). If the first command completes without error, the new command in the pending level registers will be transferred to the execution registers.
4. This pipelining process can continue until an error condition is detected by the VBTC or the tape drive/formatter.

Data can be written to the pending level registers whenever the command pending bit (in the execution level VBTC Status registers) is *not* set. If the command pending bit *is* set, then a command is already pended and the software must wait until the current operation is completed to write to the pending level registers.

If an error occurs during the execution of a command, and if another command is pended behind it, the pended command is aborted (the command aborted bit is set) to allow recovery procedures to be invoked. In this case the command pending bit remains set so the software will know the error occurred during execution of the first command (and not the pended command). Once an error occurs, all subsequent commands are aborted until the error status is cleared. Error status is cleared by issuing a new command when there is no interrupt pending, i.e., when the interrupt request bit has been reset.

The VBTC latches error status at the time the formatter goes *nonbusy*. If no error has occurred and another command has been pended, the pended command is copied to the execution level, the command pending flag is cleared, and the execution of the new command begins.

Transfer count registers are copied to the residual transfer count registers when a command completes. This process preserves the information required by the software.

### 2.1.4 Normal Blocked Mode Operation

Normal blocked mode is selected when the VIOP resets (clears) the chain mode bit in the pending level registers. At the end of each data transfer in normal blocked mode, interblock tape gaps are *expected* when reading data from tape, and *generated* when writing data to tape. Tape motion may be continuous in normal blocked mode, allowing the interblock gap to be read or written at maximum tape speed.

In normal blocked mode, each command is completed before execution of the next command begins. The completion of a command is signaled by the command in execution bit being reset; and, if the interrupt enable bit is also set, an interrupt request is signaled to the VIOP.

### 2.1.5 Chain Mode Operation

Chain mode is used when reading and writing very long records, i.e., records greater than 64 Kbytes. Chain mode is enabled when the VIOP sets the chain mode bit in the pending level registers. In chain mode, the execution level transfer address registers and transfer count are

reloaded from the pending level registers each time the transfer count goes to zero. The start flag is used to initiate the chained command; this register is not referenced again until the command completes execution.

There are two transfer address buffers (A and B) in the pending level registers. In chain mode, an internal pointer forces each successive load of the execution level register to use these address buffers alternately. If the interrupt enable bit is set, an interrupt signals the VIOP to write a new transfer address to the pending level registers after each load. This mechanism allows the VIOP to load one address buffer while the VBTC accesses the other.

Chain mode is terminated by clearing the chain mode bit. Whenever the pending level registers contents are copied to the execution level with a reset (cleared) chain mode bit, the operation mode reverts to normal blocked mode and terminates as described in the previous section.

When the VBTC has completed the current execution level operation, issued an interrupt, transferred the pending level to execution level, and completed that operation without clearing the previous interrupt, the VBTC halts operation and resets the chain mode bit. This procedure prevents the VBTC from rewriting data onto the tape during a write process or writing over data in CONVEX memory on tape reads.

### 2.1.6 Interrupts

Interrupts are enabled by setting the interrupt enable bit in the pending level register. When the interrupt enable bit is set, an interrupt request is generated upon completion of the new command. The interrupt is serviced by the VBCU interrupt acknowledge cycle, but the interrupt requesting bit remains set until software clears it by writing to the *clear interrupt* address in the pending level register.

### 2.1.7 Error Reporting

Errors may be detected by either the VBTC or the tape drive/formatter. Errors detected by the VBTC are reported in the same manner for all tape drives. Errors reported by a tape drive require knowledge of the specific drive.

When a tape drive error is detected, a global error flag is set in the execution level registers, along with additional error flags describing the nature of the error. All these error flags are reset by the execution of a new command after the clear interrupt register has been written.

#### 2.1.7.1 Controller Errors

The VBTC recognizes three types of errors:

1. If the VBTC attempts a VMEbus data transfer that is not acknowledged by the VIOP, a timeout occurs that releases the VMEbus and sets the VMEbus error bit in the VBTC execution level registers. The Transfer Address (TA) and Transfer Count (TC) will not be incremented; these values may be examined via the DMA status bytes. The data transfer failure may starve the tape unit and result in a data overrun error; however, the VMEbus error should be regarded as the cause of the failure.

2. Parity errors detected by the VBTC when reading tape data are reported as controller errors. (Parity is generated on data written to the tape and checked on data read from the tape.) Since the formatter checks parity on transfers to and from the VBTC, parity errors may be reported by either the VBTC or the formatter.
3. The third type of controller error is the *read under run* condition. The read under run bit is set when a data record on the tape is larger than the data record that was requested.

#### 2.1.7.2 Formatter/Tape Drive Errors

The STC 2921 and 2922 tape drives use an *embedded* formatter. The VBTC will hang if an access is attempted to a nonexistent tape drive; it is also a nonexistent formatter. In this situation the VBTC is waiting for the nonexistent formatter to go *busy*. A similar condition occurs if an operation is attempted with a drive that is offline.

The STC 1968 tape drive uses a single formatter to connect to multiple tape drives. An attempt to access a nonexistent STC 1968 drive is detected by the formatter and the appropriate error status is set.

> **NOTE**
>
> Consult the appropriate tape drive manuals listed under "Associated Documents" in the Preface for additional information on formatter and tape drive errors. Refer to Appendixes C and D for the status register addresses of the specific tape drives.

### 2.1.8 Diagnostics

Diagnostic program *dev5210* is used to verify proper operation of the VBTC. The VBTC contains loopback and forced parity error diagnostic capabilities. These diagnostics are available for checking parity errors and data path operations during transfers. Each diagnostic is executed by software setting the appropriate bit in the pending level register.

Refer to the *CONVEX Diagnostic Documentation (C200 Series)* for additional information on the dev5210 diagnostics program.

#### 2.1.8.1 Forced Parity Errors

The tape drives used with the VBTC expect and transmit *odd* parity. Setting the forced parity bit in the execution level registers causes *even* parity generation in the data transfer to the tape drive. The VBTC then expects even parity when data is read back from the tape drive.

#### 2.1.8.2 Loopback Test

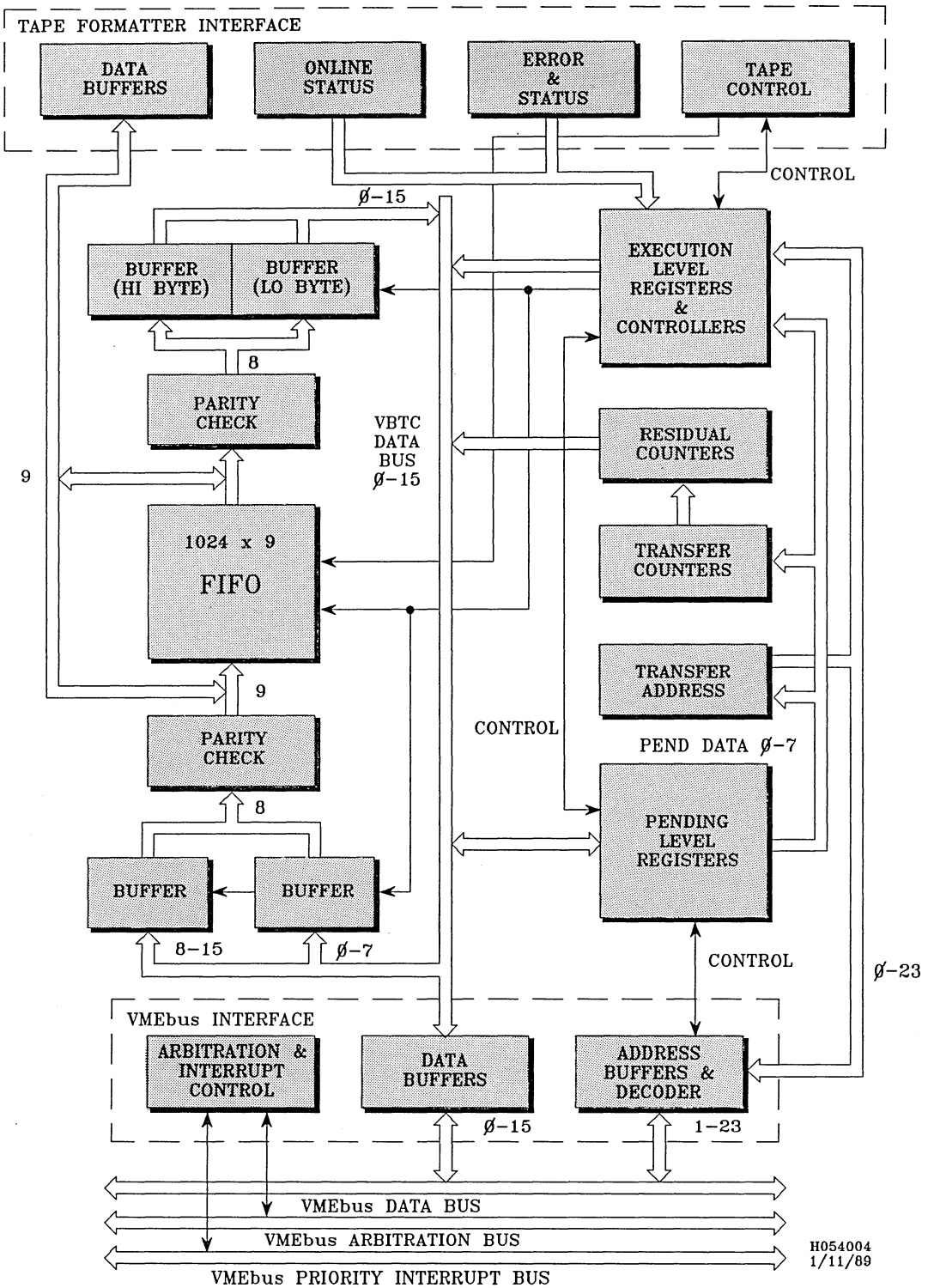
During loopback testing, normal data reads and writes occur between the VMEbus and the VBTC FIFO buffer; however, the tape drive is not started. This test verifies the data path functionality into and out of the FIFO buffer. Parity error checking is also enabled in diagnostic loopback mode.

Due to the FIFO management control PAL requirements, it is not possible to completely fill the FIFO buffer in diagnostic loopback mode. A maximum of 960 bytes can be written to the FIFO buffer during diagnostic loopback mode.

## 2.2 VBTC Functional Block Diagram

The VBTC Functional Block Diagram is illustrated in Figure 2–2:

## Figure 2–2, VBTC Functional Block Diagram



THIS PAGE INTENTIONALLY LEFT BLANK

# Chapter 3
# Configuration and Installation

## 3.1 Overview

This chapter contains configuration and installation procedures for the VMEbus Tape Controller (VBTC). Tape drive cabling and controller jumper settings are described in detail. A software integration procedure describes the process used to add the VBTC to the CONVEX operating system's */ioconfig* file.

## 3.2 Inspection

> **CAUTION**
>
> The VBTC is extremely sensitive to Electrostatic Discharge (ESD). Use appropriate measures when handling the board. Wear a wrist ground strap or other grounding device when installing or performing maintenance on the VBTC.

The specially designed package for shipping the VBTC controller protects the board against electrostatic damage. Inspect the package upon receipt for signs of damage during shipment. Remove the board from the package and carefully examine the board for damaged components. If any damage is found by the visual inspection, document the damage and contact the carrier immediately for damage claims.

> **NOTE**
>
> Save all packaging material until after operational checkout of the board. This enables the board to be returned should problems exist.

## 3.3 Damage Claims

Damage claims should be completed by the customer and given to the shipping representative. Damage claims forms may be obtained from the shipping representative.

## 3.4 Electrostatic Prevention

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

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

Due to the electrostatically sensitive devices used within the circuitry, the VBTC is sensitive to static electricity. The controller can be damaged by Electrostatic Damage (ESD) caused during maintenance procedures, e.g., installation. Use proper care when handling or performing maintenance on or around the controller board. To avoid damage to electronic devices, service personnel must observe the following warning when servicing the VBTC:

> **CAUTION**
>
> The VBTC is extremely sensitive to Electrostatic Discharge (ESD). Use appropriate measures when handling the board. Wear a wrist ground strap or other grounding device when installing or performing maintenance on the VBTC.

The static charge levels based on various activities and humidity levels are shown in Table 3–1:

**Table 3–1, Static Charge Levels and Relative Humidity**

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

## 3.5 Configuration

> **NOTE**
>
> The configuration information presented in this section is subject to change. Refer to the *VMEbus Tape Controller Configurator* document in Appendix E for updated configuration information.

The VBTC's base address, interrupt level, and bus request level must be properly configured

before it is installed in the VMEbus chassis. All three of these settings can be selected on two jumper blocks on the VBTC. The relationships of the jumper blocks to their signals are shown in Tables 3–2 and 3–3:

### Table 3–2, Jumper Block/Address Bit Relationship

| JUMPER/ADDRESS BITS | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Address | A6 | A7 | A8 | A9 | A10 | A11 | A12 | A13 |
| Jumper | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

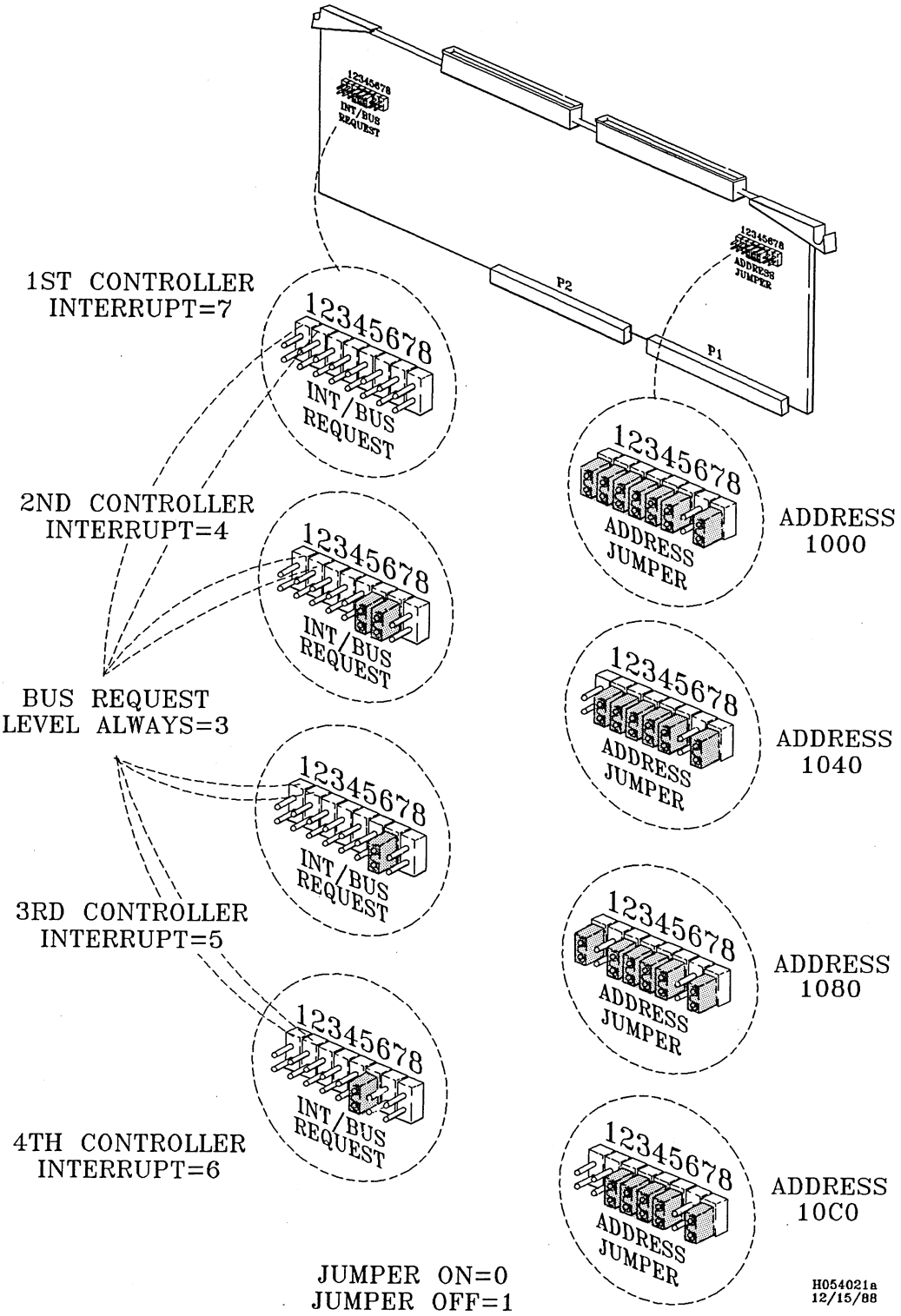
### Table 3–3, Jumper Block/Bus Request and Interrupt Bit Relationships

| JUMPERS/ADDRESS BITS | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Lower Nibble Bus Request | | | | Upper Nibble Interrupt Request | | | |
| Address | BR0 | BR1 | XX | XX | XX | INT0 | INT1 | INT2 |
| Jumper | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Address, interrupt level, and bus request level jumper settings for four VBTCs are shown in Figure 3–1. The addresses and interrupt levels for optional VBTCs are selected by modifying the jumpers to represent the appropriate value. All VBTC bus request levels should be set to **3**.

Up to four VBTCs can be installed in a VMEbus chassis. The addresses shown in Figure 3–1 for the first, second, third, and fourth VBTCs are 0x1000, 0x1040, 0x1080, and 0x10C0, respectively. The example interrupt levels for the first, second, third, and fourth VBTCs are 7, 4, 5, and 6, respectively. Field engineers may select any interrupt level that is not currently assigned to another VMEbus board.

## Figure 3–1, VBTC Configuration



# 3.6 VBTC Installation and Removal

> **NOTE**
>
> The following sections contain procedures for extending and replacing the VMEbus chassis. Refer to the *CONVEX VIOP/VBCU Service Guide* for VMEbus chassis removal and installation procedures.

The following sections provide procedures on VBTC removal and installation. The following procedures are described:

- VMEbus Chassis Extension
- VBTC Installation
- Cable Connections
- VBTC Removal
- VMEbus Chassis Replacement

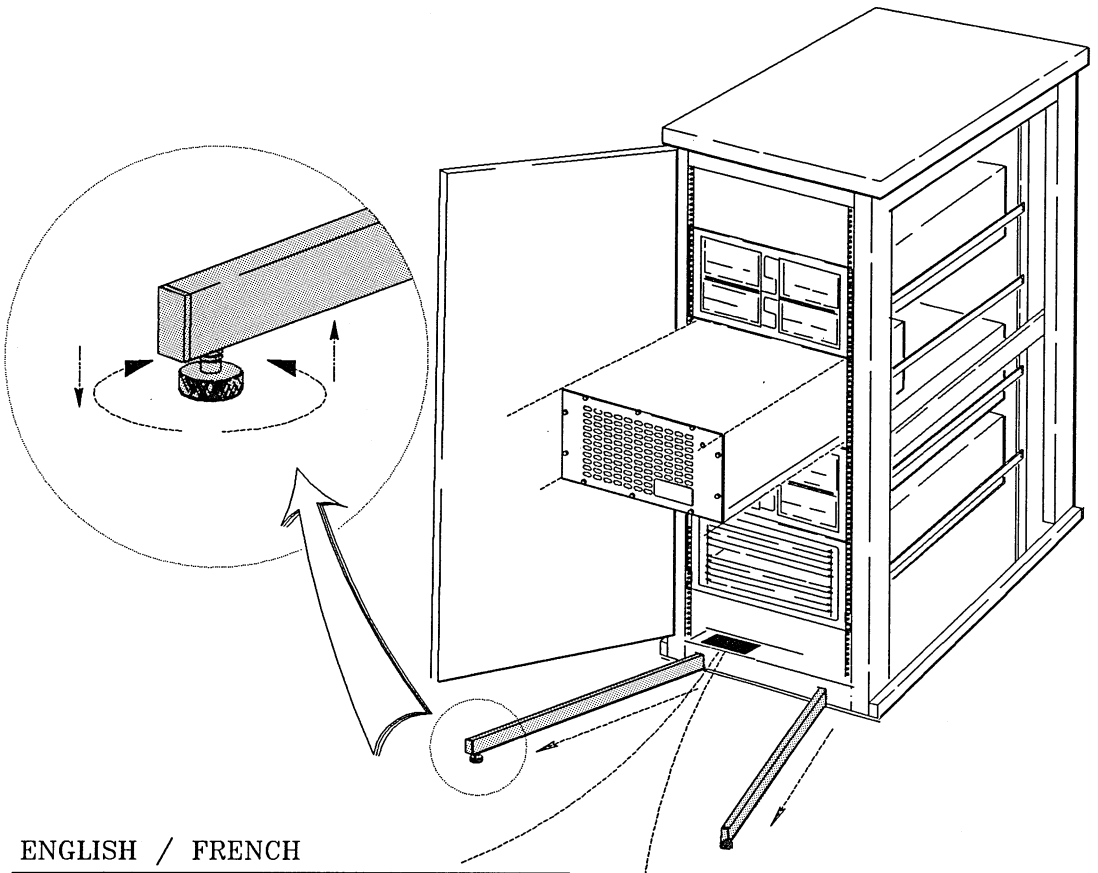
## 3.6.1 VMEbus Chassis Extension

The VMEbus chassis must be extended before installing or removing a controller. Use the following procedures for extending the VMEbus chassis.

> **WARNING**
>
> Expansion cabinet stabilizer bars must be extended before performing any maintenance or replacement of controllers in the VMEbus chassis. Failure to do so will make the expansion cabinet unstable, increasing the possibility of it falling forward. This can cause injury to personnel and will cause damage to equipment.

1. Extend the expansion cabinet stabilizer bars and adjust the feet until they are in firm contact with the floor as shown in Figure 3–2:

## Figure 3–2, Expansion Cabinet Stabilizer Bars



ENGLISH / FRENCH

| CAUTION | ATTENTION |
|---|---|
| TO REDUCE RISK OF POSSIBLE INJURY DUE TO UNSTABLE UNIT, ACTUATE STABILIZER BEFORE ANY PERIPHERAL IS EXTENDED.<br>1. TO ACTUATE STABILIZER, FULLY EXTEND ANTITILT CHANNELS AND LOWER CHANNEL SUPPORT FEET FIRMLY TO THE FLOOR.<br>2. INSURE THAT LOCKING MECHANISMS ARE INSTALLED IN ALL OTHER EXTENDABLE UNITS.<br>3. NEVER EXTEND MORE THAN ONE UNIT AT A TIME | POUR REDUIRE LE RISQUE D'ACCIDENT ATTRIBUABLE A L'INSTABILITE DE L'UNITE, DEPLOYER LES STABILISATEURS AVANT DE SORTIR LES PERIPHERIQUES.<br>1. POUR DEPLOYER LES STABILISATEURS, TIRER COMPLETEMENT LES BRAS ANTI-BASCULEMENT ET ABAISER LES PATTES DE FACON QU'ELLES REPOSENT SOLIDEMENT SUR LE SOL.<br>2. S'ASSURER QUE TOUS LES PERIPHERIQUES SON MUNIS DE VIS DE BLOCAGE.<br>3. NE JAMAIS SORTIR PLUS D'UN PERIPHERIQUE A UN MOMMENT DONNE. |

ENGLISH / GERMAN

| CAUTION | ACHTUNG |
|---|---|
| TO REDUCE RISK OF POSSIBLE INJURY DUE TO UNSTABLE UNIT, ACTUATE STABILIZER BEFORE ANY PERIPHERAL IS EXTENDED.<br>1. TO ACTUATE STABILIZER, FULLY EXTEND ANTITILT CHANNELS AND LOWER CHANNEL SUPPORT FEET FIRMLY TO THE FLOOR.<br>2. INSURE THAT LOCKING MECHANISMS ARE INSTALLED IN ALL OTHER EXTENDABLE UNITS.<br>3. NEVER EXTEND MORE THAN ONE UNIT AT A TIME | ZUR VERMEIDUNG VON GEFAHRDUNG DURCH EIN INSTABILES GERAT SIND VOR DER HERAUSNAHME VON PERIPHERALS DER STABILISIERUNGSMECHANISMUS BETATIGT WERDEN.<br>1. UM DIE STABILISIERUNGSEINRICHTUNG ZU BETATIGEN, SIND DER "ANTITILT KANAL" GANZ HERAUS ZU ZIEHEN UND DER UNTERE STUTZFUSS AUF DEN BODEN ZU FUHREN.<br>2. OBERPRUFEN SIE, OB IN ALLEN ANDEREN VERSCHIEBBAREN GERATEN DER SICHERUNGSMECHANISMUS BETATIGT IST.<br>3. ZIEHEN SIE NIE MEHR ALS EIN GERAT HERAUS. |

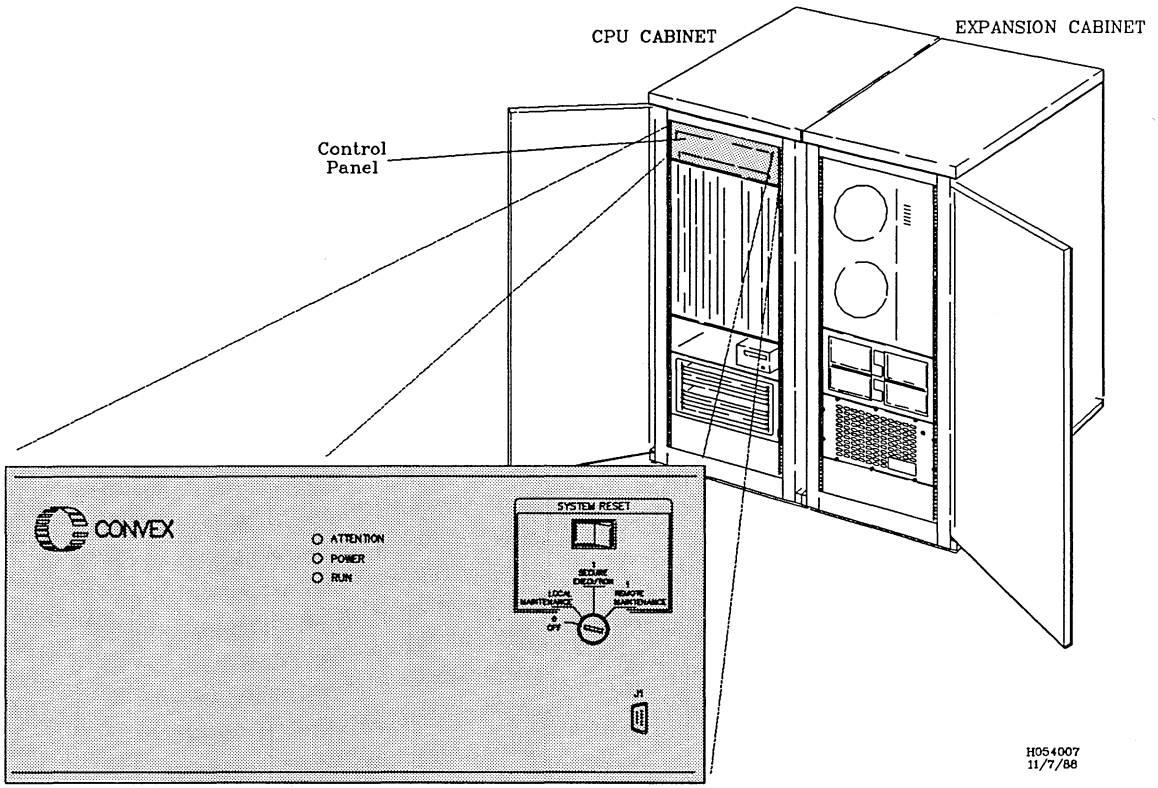
H054006a
12/15/88

**CAUTION**

System power must be OFF before performing any type of maintenance on the VMEbus chassis. Failure to do so may cause severe damage to system equipment. Refer to the *CONVEX Processor Operation Guide (C1, C120, C130, C210, C220)* for the shutdown procedures for a CONVEX computer.

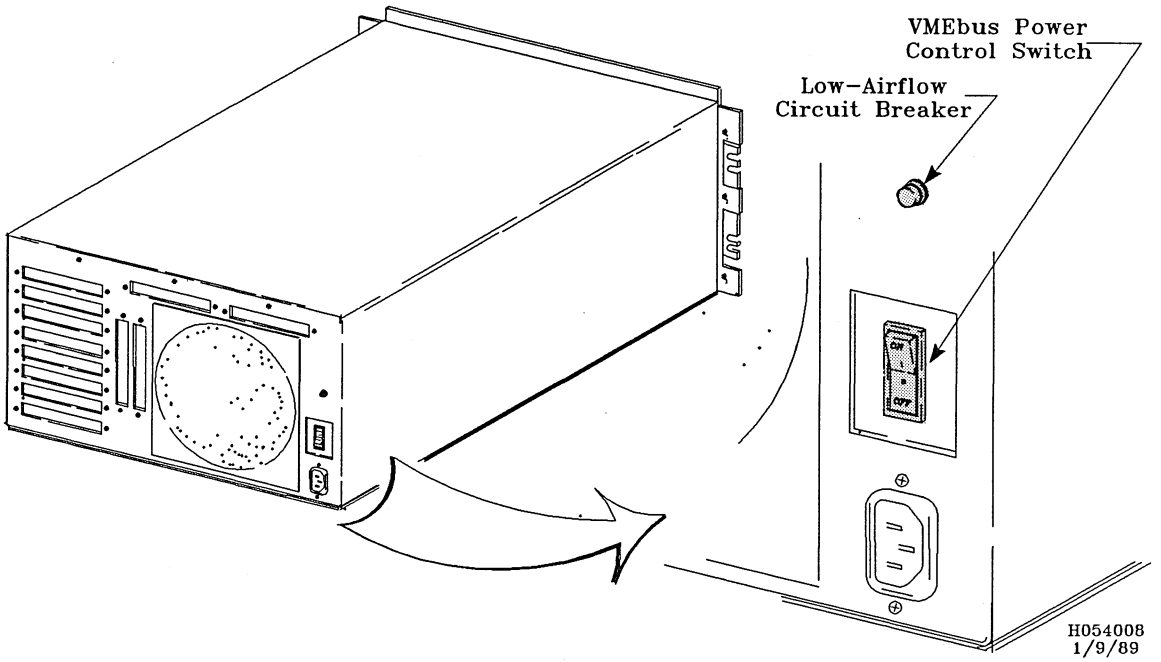
2. Turn the processor's front panel key switch to the **OFF** position as shown in Figure 3–3:

**Figure 3–3, Typical Front Panel Power Control Switch**



3. Set the VMEbus chassis power control switch to the **OFF** position as shown in Figure 3–4:

**Figure 3–4, VMEbus Chassis Power Control Switch**




4. Remove the 2 front panel locking screws and extend the VMEbus chassis.
5. Loosen the top panel captive-lock screws and remove the top panel from the VMEbus chassis.

## 3.6.2 VBTC Board Installation

This installation procedure reflects the initial installation of the VBTC. Additional steps, such as the cable routing or cable connections to the tape drive, may not be necessary when replacing a VBTC. However, each step should be read to ensure proper installation.

> **CAUTION**
>
> The VBTC is extremely sensitive to Electrostatic Discharge (ESD). Use appropriate measures when handling the VBTC board. Wear a wrist ground strap or other grounding device when installing or performing maintenance on the VBTC.

> **CAUTION**
>
> Do not install the VBTC in ***slot 9*** of a single VMEbus chassis; the VBTC will not function in slot 9. Refer to the *CONVEX VIOP/VBCU Service Guide* for additional information on the VMEbus chassis types.

Install the VBTC in an available slot in the VMEbus chassis, gently pushing down evenly on both ends of the board to prevent damage to the board pins. There are no lock screws to secure the board; the board is secured by the guide rails and the tension of the card connector.

### 3.6.3 VBTC To Tape Drive Cable Connections

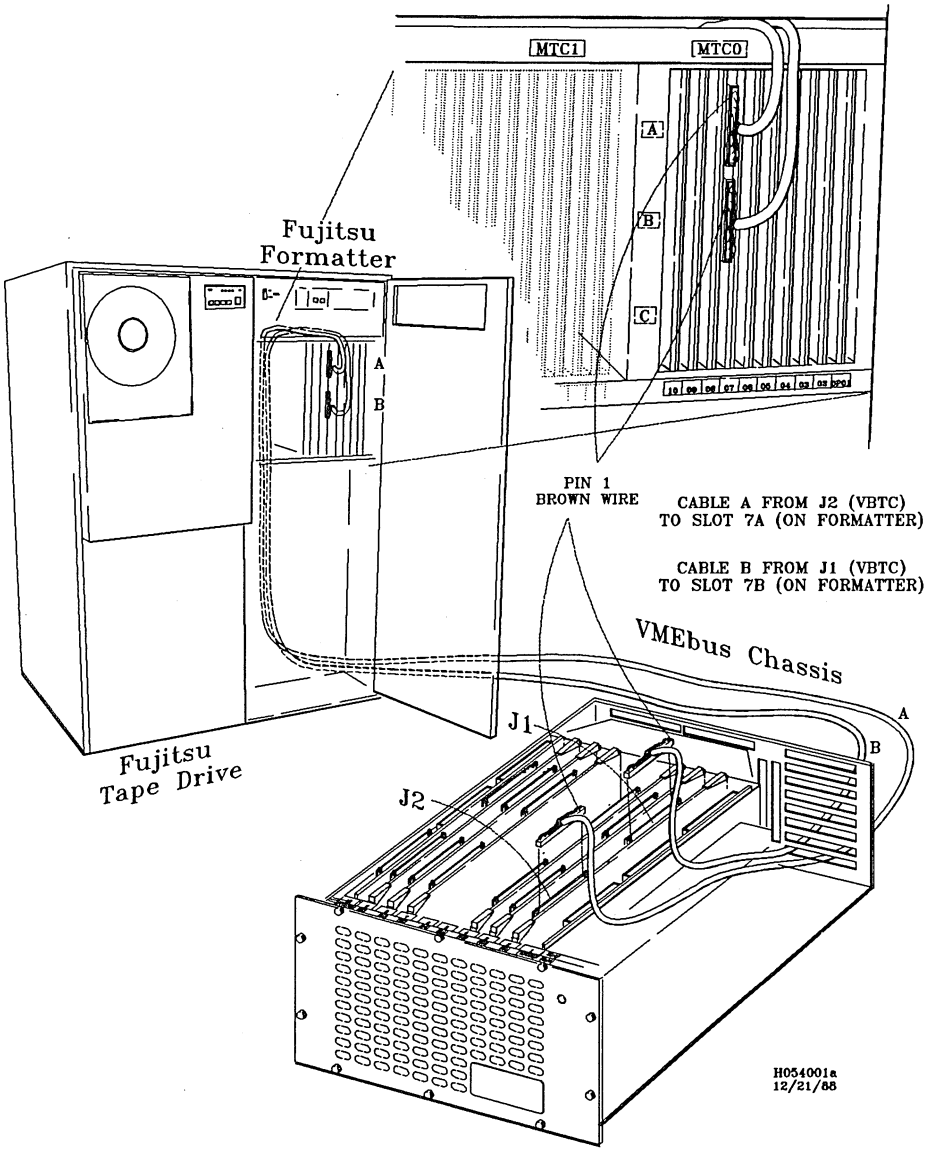
Table 3–4 contains basic cable connection information for the VBTC and four commonly used tape drives: Fujitsu 2436, STC 1968, STC 2921, and STC 2922. The table shows the tape drive model, cable **A** or **B** designation, VBTC board connector **J1** or **J2** designation, and the tape drive/formatter connector designation.

**Table 3–4, VBTC To Tape Drive Cable Connections**

| Drive Name | Cable(s) | From VBTC Connector | To Drive/ Formatter Connector |
|---|---|---|---|
| Fujitsu 2436 | A | J2 | Slot 7 A |
| Fujitsu 2436 | B | J1 | Slot 7 B |
| STC 1968 | A | J2 | A4 |
| STC 1968 | B | J1 | B4 |
| STC 2921 | A | J2 | J7 |
| STC 2921 | B | J1 | J6 |
| STC 2922 | A | J2 | J7 |
| STC 2922 | B | J1 | J6 |

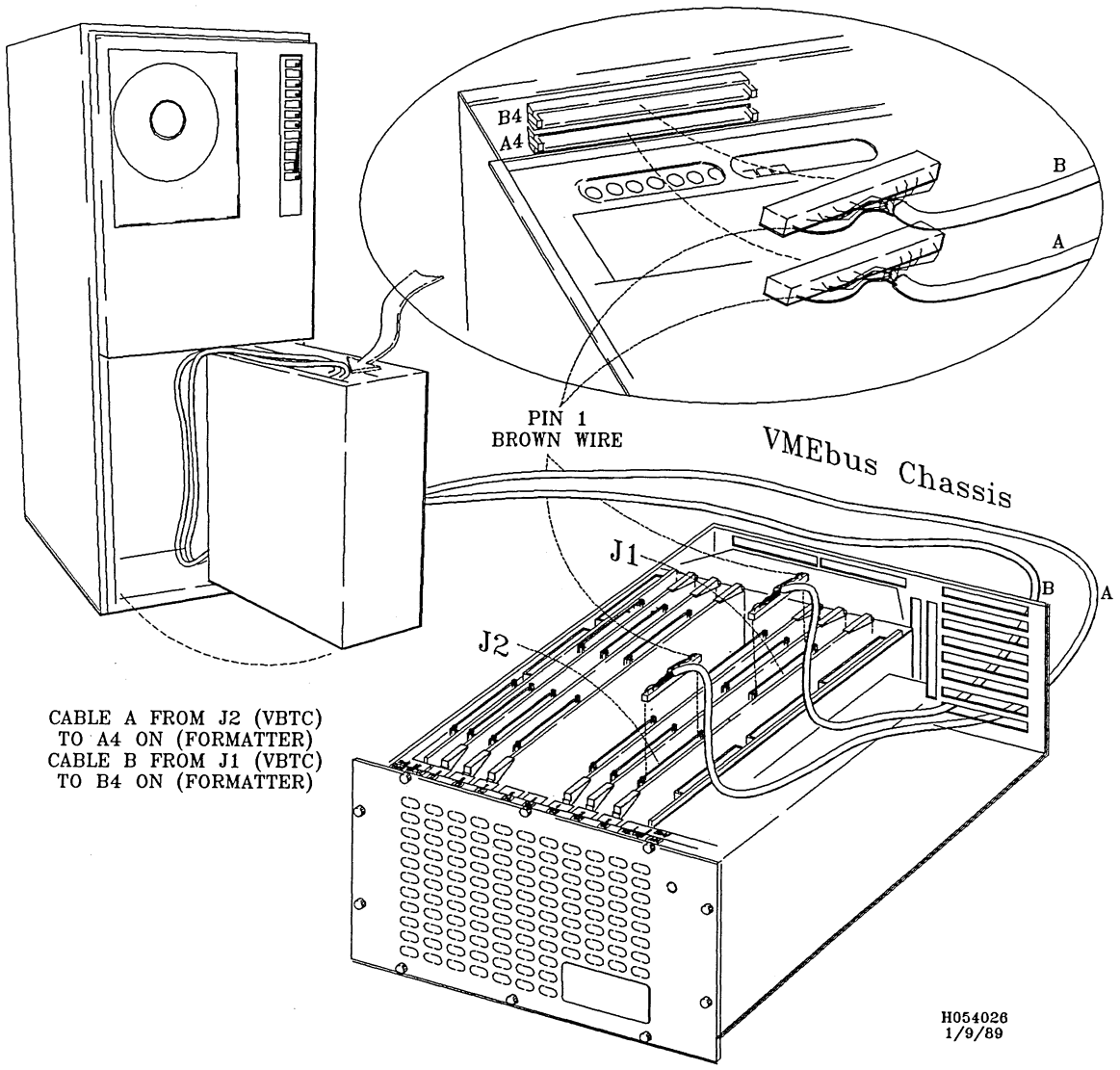
The cable connections between a VBTC and a Fujitsu 2436 tape drive/formatter are illustrated in Figure 3–5:

**Figure 3–5, Fujitsu 2436 Tape Drive Cable Connections**



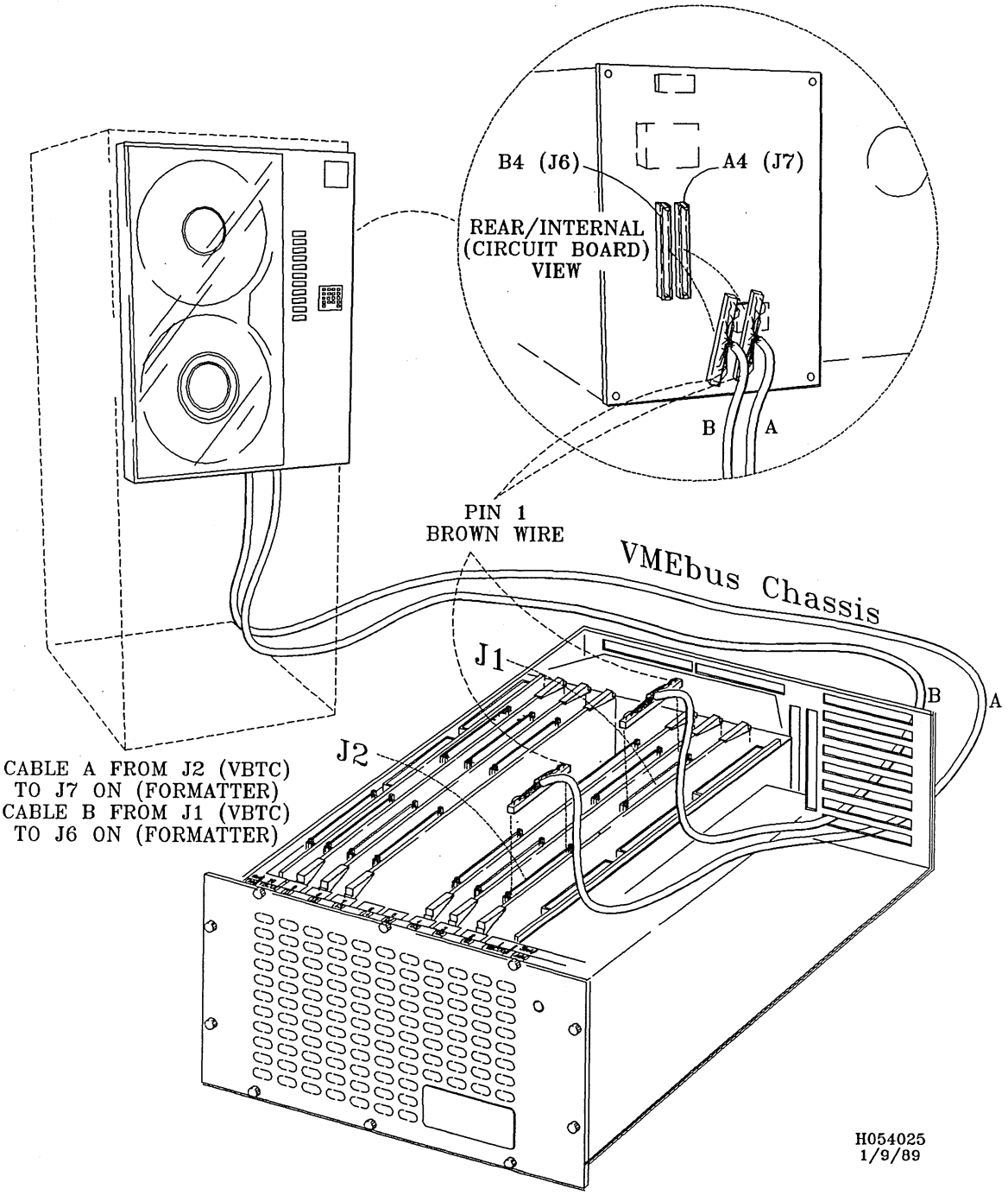
The cable connections between a VBTC and an STC 1968 tape drive/formatter are illustrated in Figure 3–6:

## Figure 3–6, STC 1968 Tape Drive Cable Connections



The cable connections between a VBTC and the STC 2921 and 2922 tape drives/formatters are illustrated in Figure 3–7:

**Figure 3–7, STC 2921, 2922 Tape Drive Cable Connections**



### 3.6.4 VMEbus Chassis Cable Routing

The VMEbus backplane slot positions are labeled on the front of each VMEbus chassis. VMEbus controller cables exit at the rear of the VMEbus chassis through cable openings. Cable opening numbers are stamped on the rear panel of each VMEbus chassis. Cable routing, from the controller to the rear of the VMEbus chassis, should always follow a prescribed sequence. Cables from a given controller backplane slot position should always exit the VMEbus chassis at the same hole position. Cable openings and device types for each type of VMEbus chassis are defined in Table 3–5:

**CAUTION**

Do not install the VBTC in *slot 9* of a single VMEbus chassis; the VBTC will not function in slot 9. Refer to the *CONVEX VIOP/VBCU Service Guide* for additional information on the VMEbus chassis types.

**Table 3–5, Cable Opening Numbers for VMEbus Chassis**

| Cable Opening Number | Dual VMEbus | Single VMEbus | Combo VME/Multibus |
|---|---|---|---|
| **1** | VBCU-0 | VBCU | VBCU |
| **2** | VME-0 Ctlr 1 | Ctlr 1 | VMEbus Ctlr 1 |
| **3** | VME-0 Ctlr 2 | Ctlr 2 | VMEbus Ctlr 2 |
| **4** | VME-0 Ctlr 3 | Ctlr 3 | VMEbus Ctlr 3 |
| **5** | VME-0 Ctlr 4 | Ctlr 4 | VMEbus Ctlr 4 |
| **6** | VME-1 Ctlr 4 | Ctlr 5 | VMEbus Ctlr 5 |
| **7** | VME-1 Ctlr 3 | Ctlr 6 | Multibus Ctlr 3 |
| **8** | VME-1 Ctlr 2 | Ctlr 7 | Multibus Ctlr 2 |
| **9** | VME-1 Ctlr 1 | Ctlr 7[1] | Multibus Ctlr 1 |
| **10** | unassigned | unassigned | Multibus Ctlr 0 |
| **11** | unassigned | unassigned | unassigned |
| **12** | VBCU-1 | unassigned | MBCU |

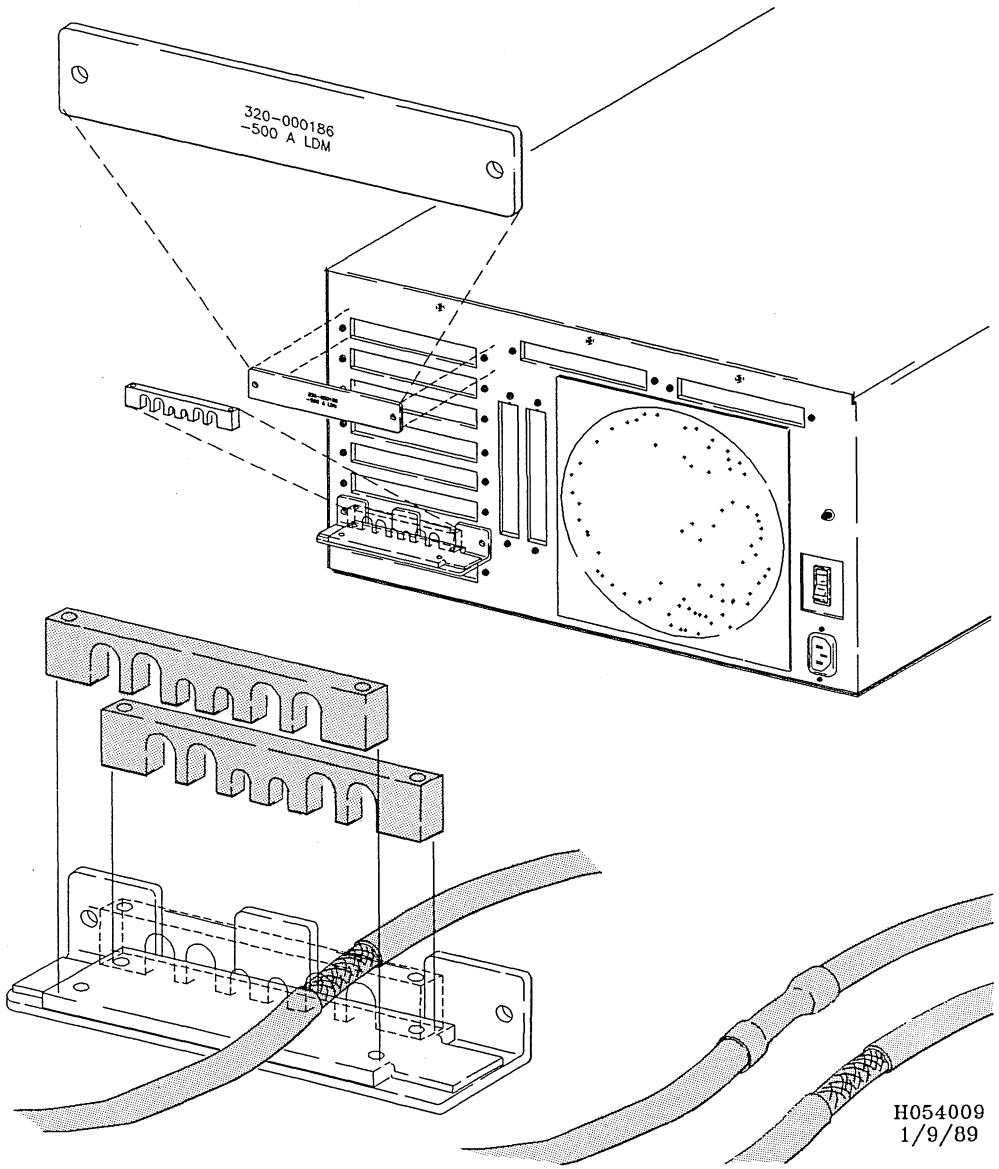
[1] This is the second board (not addressable) of a two-board controller set. The first board (addressable) of the two-board set is Ctlr number 7 in slot 8.

**NOTE**

The VMEbus backplane slot positions are labeled on the front of each chassis. Cable opening numbers are stamped on the outside rear panel on all CONVEX VME chassis. VMEbus controller cables exit the chassis at the rear through cable openings. Cables from a given controller should always exit the VMEbus chassis at the same hole position.

1. Refer to Table 3–5 and remove the appropriate cable cover plate(s) on the rear of the VMEbus chassis as shown in Figure 3–8:

## Figure 3–8, Cover Plate, Cable Clamp, and Shielded Cables

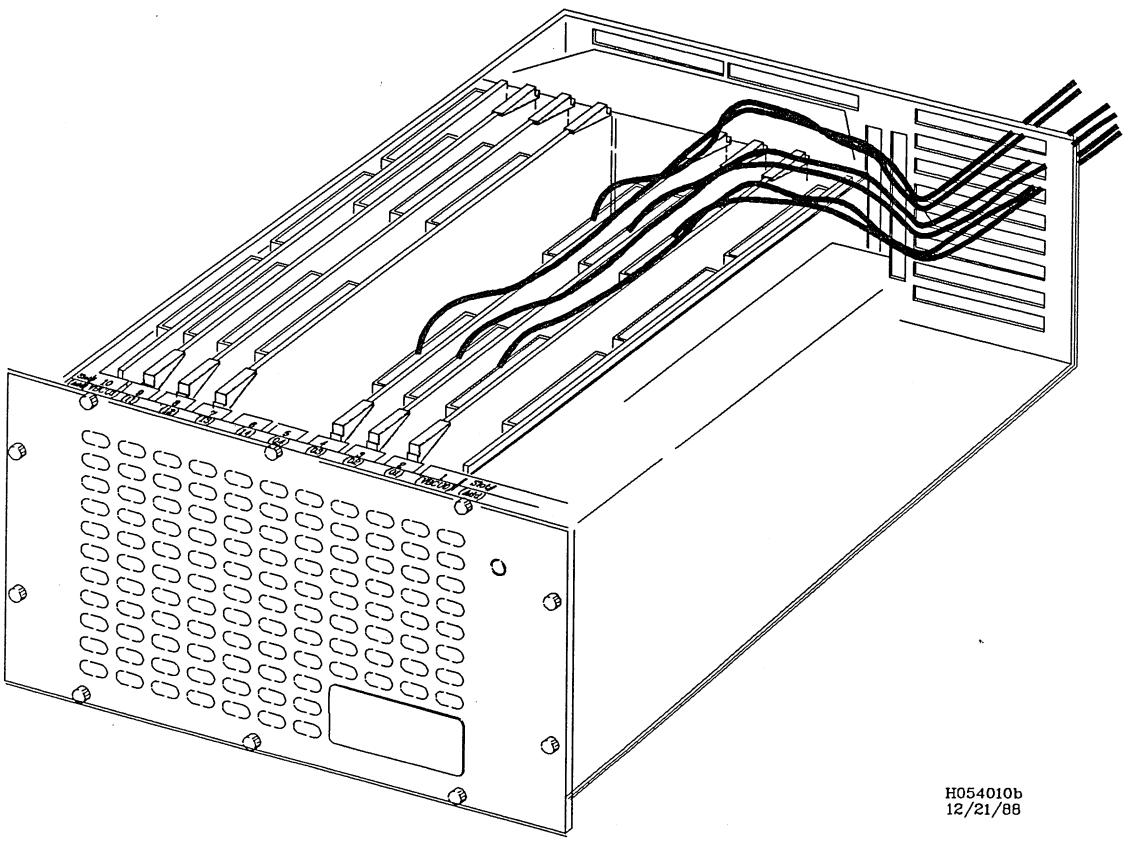


**NOTE**

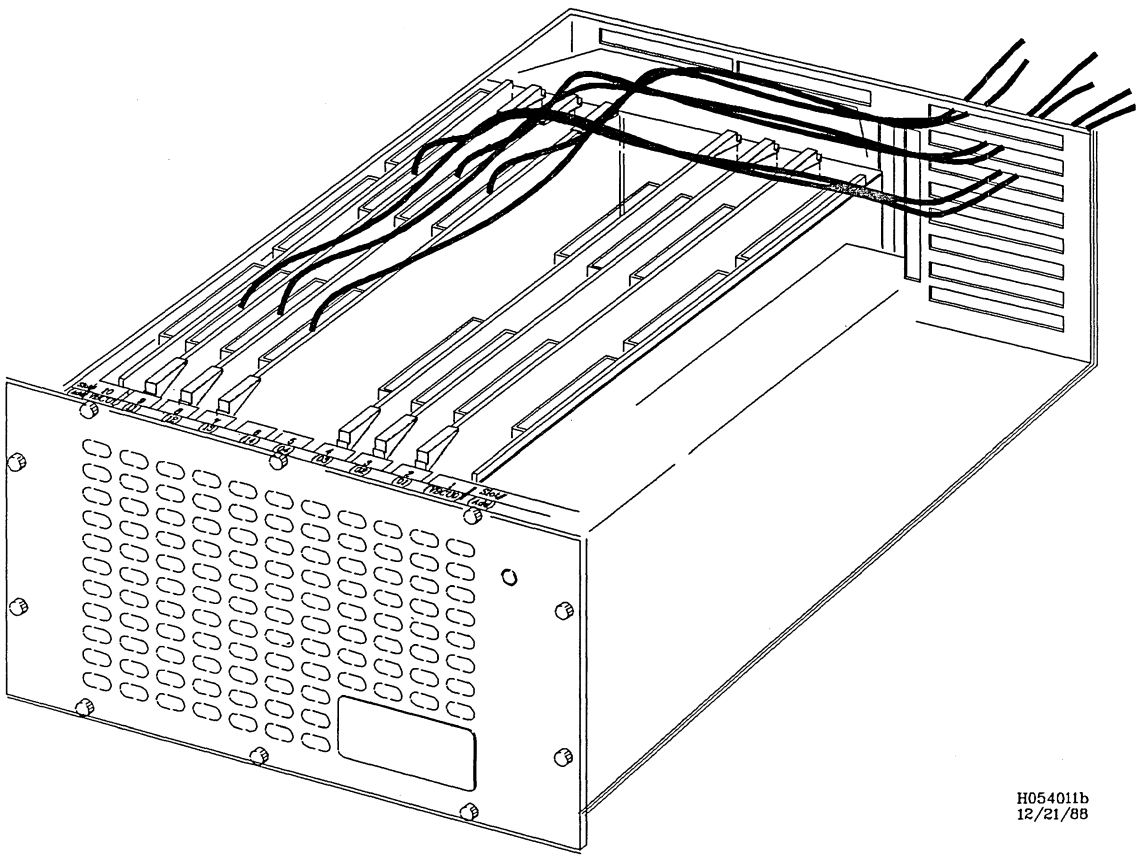
Figures 3–9 and 3–10 illustrate cabling on a *dual* VMEbus chassis. Refer to the *CONVEX VIOP/VBCU Service Guide* for cable routing information on a single VMEbus chassis and a combo VME/Multibus chassis.

The cable routings in a *dual* VMEbus chassis (with VME-0 and VME-1) are shown in Figures 3–9 and 3–10, respectively:

## Figure 3–9, VME-0 Positions and Cable Routing



## Figure 3–10, VME-1 Positions and Cable Routing

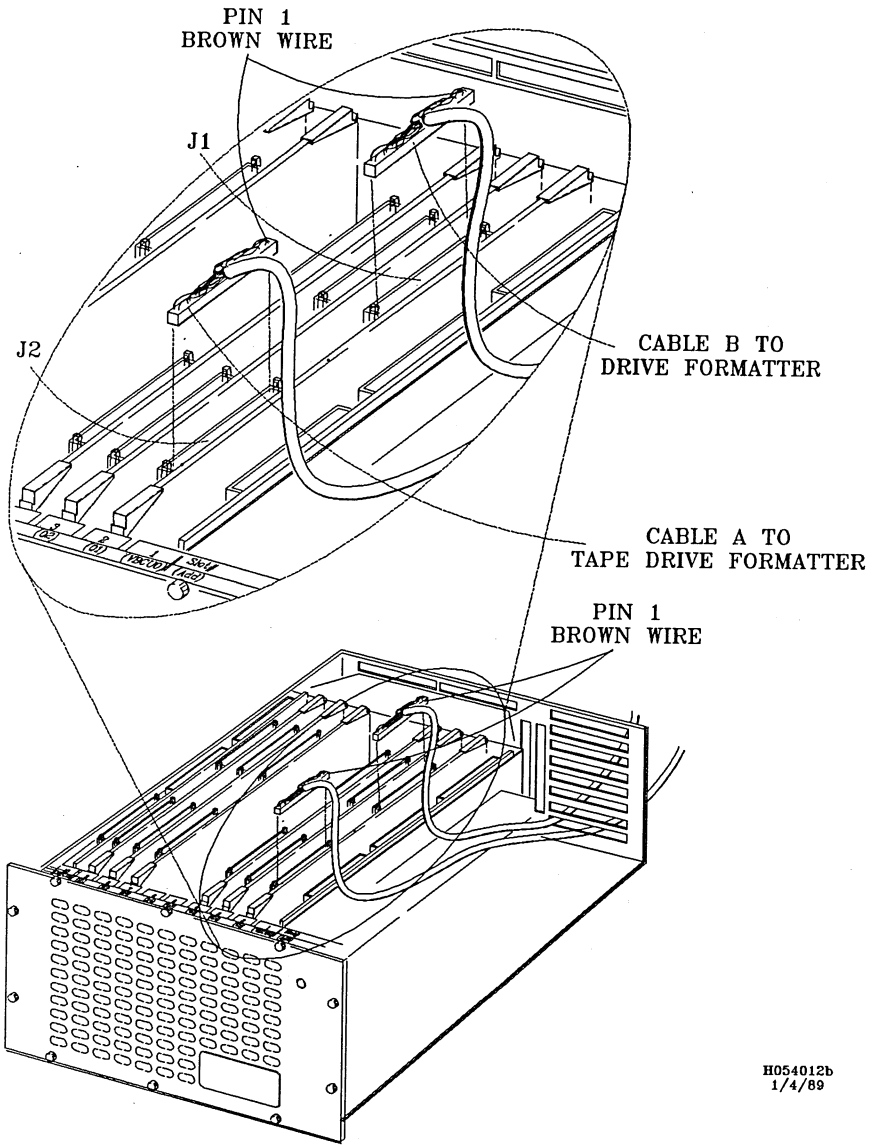



2. Route the cable end labeled **P1** of cable **B** and the cable end labeled **P2** of cable **A** through the appropriate cable opening at the rear of the VMEbus chassis (as shown in Figures 3–9 and 3–10.)

> **CAUTION**
>
> When connecting a cable to a VBTC connector, always ensure that pin 1 on the cable matches pin 1 on the connector. Pin 1 on the cable is identified by a shallow groove on the outside corner of the end clip. Pin 1 for the connector is identified by a small triangle on the outside corner of the connector. Failure to connect the appropriate port will render the controller inoperable.

3. Connect the cable **B** end labeled **P1** to connector **J1** of the VBTC and connect the cable **A** end labeled **P2** to connector **J2** of the VBTC as shown in Figure 3–11:

## Figure 3–11, VBTC Cable Connections




4. Strip the cable shields on cables **A** and **B** and secure the cables to the rear of the VMEbus chassis as shown in Figure 3–8.

NOTE

Refer to the installation procedures in the appropriate STC and Fujitsu maintenance manuals (listed under "Associated Documents" in the Preface) for additional cable routing information.

5. Connect cables **A** and **B** to the tape drive. (Refer to Table 3–4, "VBTC To Tape Drive Cable Connections", and Figures 3–5 through 3–7 for specific connection information.)

### 3.6.5 VBTC Removal

NOTE

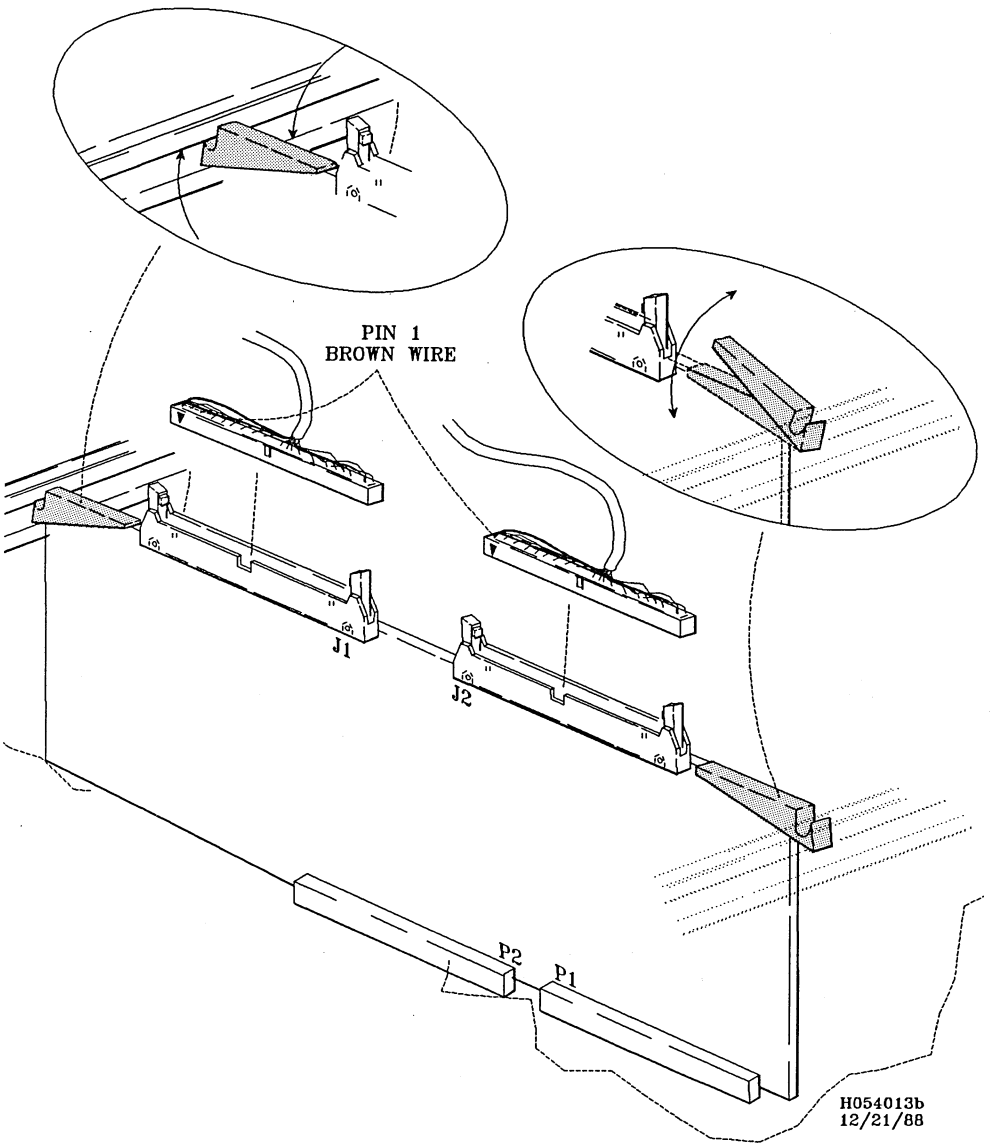
This section presumes the VMEbus chassis has been extended in accordance with the procedures presented in this chapter.

CAUTION

The VBTC is extremely sensitive to Electrostatic Discharge (ESD). Use appropriate measures when handling the VBTC. Wear a wrist ground strap or other grounding device when installing or performing maintenance on the VBTC.

1. Disconnect cables A and B from the VBTC board receptacles by pressing outward on the ejectors and lifting straight up on the cable connectors.
2. Lift the board from the chassis by slowly pulling up on the ejectors located on each side of the board. This method gently lifts the board from the VMEbus chassis connector as shown in Figure 3–12:

## Figure 3–12, VBTC Removal




3. Lift the VBTC board straight up and out of the VMEbus chassis.
4. Place the board in a static bag or on a static mat to protect it from electrostatic damage.

### 3.6.6 VMEbus Chassis Replacement

> **NOTE**
>
> The procedures in this section are used when replacing the VMEbus chassis back in the cabinet. Refer to the *CONVEX VIOP/VBCU Service Guide* for original VMEbus chassis installation procedures.

Use the following procedures when replacing the VMEbus chassis back into the cabinet:

1. Place the top panel on the VMEbus chassis, and tighten the panel captive-lock screws.
2. Push in on the lock buttons on each side of the VMEbus chassis slide guides and gently push the VMEbus chassis back into the cabinet.
3. Tighten the captive-lock screws on the front of the VMEbus chassis.
4. Push the expansion cabinet stabilizer bars back into the cabinet.

## 3.7 Software Integration

The CONVEX operating system contains all software drivers for the hardware supported by CONVEX. This means that a system generation is not required when the VBTC is installed in a system.

Controllers are identified to the CONVEX operating system by a mnemonic controller code that is entered into the */ioconfig* configuration file located on the SPU disk. Once the software identification process is completed, the operating system will be able to use the controllers. The device codes for the various tape drives that connect to the VBTC are:

- **MTD–201**: STC 2921
- **MTD–202**: STC 1968
- **MTD–203**: Fujitsu 2436
- **MTD–204**: STC 2922

These codes, along with other information, are entered into the */ioconfig* file that is contained on the SPU disk. The I/O configuration file contains entries, such as VMEbus I/O Processor (VIOP) number, VMEbus chassis number, controller type, communication status register (CSR) address, interrupt number, and peripheral device type and number. The */ioconfig* file describes, in hierarchical fashion, the connections between the VIOP, VMEbus controller, and peripheral devices. The operating system uses this information for accessing a device.

The information in the hardware section of the configuration file (*/ioconfig)* must be changed whenever a VBTC is added to, or removed from, a CONVEX system. System operation problems will occur if the configuration file is not updated after a hardware change is made.

The */ioconfig* file must be edited at the SPU level to add the VBTC to the operating system. The *csr* and *interrupt level* formats for a single VBTC are shown below. The text in **boldface** indicates a variable value.

*ctlr MTC-201 csr* **1000** *int* **7**
*unit* **0** *type MTD-20x*

Refer to the *CONVEX System Manager's Guide* for additional information on modifying the */ioconfig* file after installing or removing a controller.

### 3.7.1 Example */ioconfig* File

Figure 3–13 shows an example of a typical */ioconfig* file that includes an entry for a single VMEbus Tape controller ( shown in *italics*). In this example, the VBTC is located in VME 0, the VME tape controller is specified as MTC-201, the csr is 1000, the interrupt level is 7, the unit designation is 0, and the device code is MTD-203 (Fujitsu 2436 drive).

The relative position of the VBTC data within the *ioconfig* file determines which physical drive is assigned to which logical drive by the operating system. In the example shown in Figure 3–13, the first tape drive connected to the VBTC will be assigned to logical unit 1, because it is the second tape drive specified in */ioconfig*. The Multibus tape controller MTC-001 will be assigned to the first logical unit (unit 0), since it is the first tape drive listed in */ioconfig*.

Refer to Appendix B in the *CONVEX System Manager's Guide* for additional information on how the operating system makes logical drive assignments.

**Figure 3–13, Example */ioconfig* File**

```
iop 7
    mbus 0
      ctlr DKC-001 csr 0x3f0 int 2
        unit 0 type DKD-005
      ctlr ACM-001 csr 0x3c0 int 6
        unit 0 type TTY
        unit 1 type TTY
        unit 2 type TTY
      ctlr MTC-001 csr 0x3d0 int 5
        unit 0 type MTD-001
viop 6
   vme 0
     ctlr MTC-201 csr 0x1000 int 7
       unit 0 type MTD-203
     ctlr DKC-203 csr 0x800 int 1
       unit 0 type DKD-214
     ctlr DKC-203 csr 0xa00 int 3
       unit 0 type DKD-214
```

# Appendix A
# Glossary

## A.1 Overview

The following terms are defined as they are used at CONVEX. Standard acronyms are also included. Boldfaced terms within a definition are defined in separate entries.

## A.2 Terms

**address** A number used by the hardware, operating system, or process to identify a storage location.

**asynchronous**
Data transmission that requires the use of start and stop elements for each character; necessary because of the variable time intervals between characters.

**bit** A binary digit.

**bus** A channel or path for transferring data and electrical signals.

**chain mode**
An operational mode that facilitates: (1) the reading of records of unknown length, and (2) the writing of records of unlimited length.

**chassis** The physical enclosure that houses the computer or subsystem.

**clear interrupt**
A register at offset 0xB which, when written to will clear a pending interrupt.

**control registers**
Registers in the VBTC address space that control VBTC operation.

**DMA status bytes**
Several registers in the VBTC contain the current transfer address, transfer count, and residual transfer count.

**electrostatic discharge**
The release of static electricity from a charged object to an object at a different electrostatic potential.

**enabled** A condition or bit is said to be enabled when it is *true* or set to *1*.

**error mux** The method of presenting tape drive secondary status information. The error mux begins at address 0x20.

**ESD** *See* **electrostatic discharge**.

**execution level registers**
Registers between addresses 10 and 1F containing information relating to the currently executing command or to the immediately previous command.

**FIFO** *See* **first-in first-out.**

**first-in first-out**
Refers to a data structure in which entries are made at one end and deletions are made at the other.

**flag** A 1-bit operand that is generally used to indicate the results of an operation. The results are in the form true or false.

**formatter** The portion of a tape drive that accepts data and commands from the VBTC, encodes the data for the appropriate recording format, and transmits the data to a tape drive.

**IEEE** Acronymn for the Institute of Electrical and Electronics Engineers, a professional engineering organization with a strong interest in computer systems and their uses.

**interrupt** An occurrence, other than an exception, that changes the normal flow of instruction execution. An interrupt originates from hardware, such as an I/O device.

**jump** Departure from normal one-step incrementing of the program counter.

**jumper selectable**
The ability to select address, interrupt level, and bus request level on the VBTC by attaching or removing jumpers.

**load** An instruction that moves data from memory to a register.

**loopback mode**
An operational mode of the VBTC where "tape writes" write only the FIFO and "tape reads" only read the FIFO. No data is transmitted to or from the tape drive and no drive need be present.

**master** A functional module that initiates DTB cycles in order to transfer data between itself and a **slave** module.

**nonbusy** An operational condition where the tape drive is not executing a command.

**normal blocked mode**
A method of writing records on the tape of a fixed length. *See also* **chain mode.**

**pack** The process of combining two bytes to form a 16-bit word.

**parity** A method used to detect a single bit error in a byte of data. A parity bit is appended to a byte of data; its value depends on the sum of bits set in the data byte.

**parity error**
The condition occurring when the parity of a byte did not match its parity bit.

**pending level registers**
A register between addresses 00 and 0F containing information to be transferred to execution level whenever the start flag is written.

**pointer** An address or other indication of a location.

**read** A memory operation in which the contents of a memory location are accessed and passed to another part of the machine.

**read operation**
A read operation in CONVEX equipment is always referenced to the device or system that initiated the read operation or is the bus master. Data always *goes to* the device or system that initiates the read operation, or is the bus master.

**register** A programmer-visible hardware entity used to contain addresses, operands, or status.

**reset** The process of establishing a known state in a machine register.

**slave** A functional module that detects DTB cycles initiated by a **master** module and transfers data between itself and the **master**.

**SEG-D** A tape standard developed by the Society for Exploration Geophysicists (SEG). The -D suffix indicates field data, i.e., data that has not been demultiplexed. The SEG standard defines the contents of the data records and the headers which typically precede the records.

**synchronous**
Referring to the transmission of data related to a clock signal.

**tape density**
The density, in bits per inch, that data is recorded on magnetic tape.

**unpack** The process of splitting a 16-bit word into two 8-bit words.

**VBCU** Abbreviation for VMEbus Control Unit.

**VBTC** *See* VMEbus Tape Controller.

**VIOP** Abbreviation for VMEbus Input/Output Processor.

**VMEbus** An industry-standard, 32-bit bus used in computer systems.

**VMEbus chassis**
A chassis containing one or two VMEbuses.

**VMEbus tape controller**
A circuit board used to interface the CONVEX VMEbus to a tape drive.

THIS PAGE INTENTIONALLY LEFT BLANK

# Appendix B
# Pending Level Registers

## B.1 Overview

This appendix documents the VBTC pending level registers. The VIOP reads and writes the pending level registers under software device driver control.

The layout of the pending level registers is shown in Table B–1. All of the register addresses listed in Table B–1 are hexadecimal offsets from the VBTC's base address. Individual byte and bit descriptions of the pending level registers are provided in subsequent sections.

**Table B–1, Pending Level Registers**

| ADDR | MSB | | | | | | | LSB |
|---|---|---|---|---|---|---|---|---|
| 00 | Start Flag | | | | | | | |
| 01 | Int Ena | Diag Loop Ena | Force Par Err | Chain Mode | Speed Select | Addr Mod | Resv | Resv |
| 02 | 0 | 0 | 0 | 0 | 0 | Tus2 | Tus1 | Tus0 |
| 03 | 0 | 0 | 0 | 0 | 0 | 0 | Den1 | Den0 |
| 04 | 0 | 0 | 0 | Cmd4 | Cmd3 | Cmd2 | Cmd1 | Cmd0 |
| 05 | Transfer Count Byte 2 | | | | | | | |
| 06 | Transfer Count Byte 1 | | | | | | | |
| 07 | Transfer Count Byte 0 | | | | | | | |
| 08 | Transfer Buffer A Address Byte 2 | | | | | | | |
| 09 | Transfer Buffer A Address Byte 1 | | | | | | | |
| 0A | Transfer Buffer A Address Byte 0 | | | | | | | |
| 0B | Clear Interrupt | | | | | | | |
| 0C | Transfer Buffer B Address Byte 2 | | | | | | | |
| 0D | Transfer Buffer B Address Byte 1 | | | | | | | |
| 0E | Transfer Buffer B Address Byte 0 | | | | | | | |
| 0F | Reserved | | | | | | | Reset |

## B.2 Start Flag — Byte 00

VBTC operations begin when any value is written to this pending level register. Writing to the start flag byte enables the transfer of the data in the pending level registers to the execution level registers and starts the operation. Reading this byte has no effect.

## B.3 VBTC Control Byte — Byte 01

The VBTC Control byte contains information used to control the execution of a command when it moves from the pending level to the execution level. This byte has no effect while still at the pending level. Each bit in the VBTC Control byte is described below.

### B.3.1 Interrupt Enable — Bit 7

If the Interrupt Enable bit is reset, completion of a VBTC command will set the appropriate status bits, but an interrupt will *not* be generated to the VIOP. If this bit is set, completion of a command will set the status bits and generate an interrupt request to the VIOP. Completion of a chain mode buffer will also generate an interupt if this bit is set.

### B.3.2 Diagnostic Loopback Enable — Bit 6

The VBTC operates normally if the Diagnostic Loopback Enable bit is not set. If this bit is set, the 1,024 byte data FIFO buffer will not output data to, or take data from, the tape formatter; data will be transferred to and from the FIFO buffer only. A maximum of 960 bytes can be written to the FIFO buffer in diagnostic loopback mode.

### B.3.3 Force Data Parity Errors — Bit 5

If the Force Data Parity bit is set, all data transferred to the tape drive will be forced to have bad parity. Data transferred from the tape drive with good parity will cause bad parity to be detected.

### B.3.4 Chain Mode — Bit 4

If the Chain Mode bit is not set, the controller will expect or generate an inter-block gap to terminate the data transfer. If this bit is set, double buffering will be used, allowing the transfer of very large (greater than 64 Kbytes) data blocks.

If the VBTC is in chain mode and an interrupt is pending when another interrupt occurs, the VBTC will reset the chain mode bit and the operation will be terminated. This process prevents the VBTC from recopying data to or from the same location in memory.

### B.3.5 Speed Select — Bit 3

Setting the Speed Select bit, issuing the Diagnostic Mode Select (DMS) command, and the Write (WRT) command with a *specified data pattern* to the STC 2922 tape drive allows the speed to be set to 100 ips streaming or 50 ips start/stop operation. After the speed has been selected, the VBTC must be reset via software before further commands are issued.

> **CAUTION**
>
> Using the Speed Select bit on a tape drive other than the STC 2922 will bring *unpredictable* results. Consult the *STC 2920 Maintenance Manual* for a description of the specific data bytes used to select speed/gap size combinations on the STC 2922 drive.

### B.3.6 Address Modifier — Bit 2

The Address Modifier bit is used during Direct Memory Access (DMA) transfers; this bit is set before the DMA transfer occurs. If this bit is set, the controller responds with a modifier of 0x39, Standard NonPrivileged Data Access. If this bit is no set, the VBTC responds with a 0x3D modifier, Standard Supervisory Data Access.

### B.3.7 Reserved — Bits 1,0

These bits are reserved.

## B.4 Formatter Control Bytes — Bytes 02–04

The Formatter Control bytes control the operation of the formatter(s) attached to the controller. Information written to these registers via the VMEbus is loaded only into the pending level. Actual control of the formatter occurs when the start flag is set and the pending level is transferred to the execution level. The three Formatter Control bytes are described below.

### B.4.1 Tape Unit Select — Byte 02

The Tape Unit Select byte is used to select the tape drive addressed by the formatter (or by the VBTC in the case of an embedded formatter, e.g., STC 2921 and 2922 drives). Table B–2 shows the eight possible tape unit numbers assigned by the Tape Unit Select byte.

- **Bits 7-3** — Must be zero
- **Bits 2-0** — Tape unit select field

**Table B–2, Tape Unit Select Codes**

| Tape Unit Number | Tus<2> | Tus<1> | Tus<0> |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 |
| 7 | 1 | 1 | 1 |

### B.4.2 Density Select — Byte 03

The Density Select byte is used to select the tape density. Each bit in the Density Select byte is described below.

- **Bits 7-2** — Must be zero
- **Bits 1,0** — These bits select the tape write density, if enabled by the tape unit controls, and if the selected tape is at Bottom of Tape (BOT). Otherwise, this field is ignored by the tape drive. The various options for these bits are:
  - **0** — 1600 BPI Phase Encoded
  - **1** — 6250 BPI Group Coded
  - **2** — 800 BPI NRZI

Table B–3 shows the three possible values for the Density Select byte:

**Table B–3, Density Select Codes**

| Density Select | Den<1> | Den<0> |
|---|---|---|
| 1600 bpi | 0 | 0 |
| 6250 bpi | 0 | 1 |
| 800 bpi | 1 | 0 |

### B.4.3 Command Select — Byte 04

The Command Select byte is used to select tape operation commands. Refer to the specific tape drive manual for a detailed explanation of the command (CMD) acronyms shown in Table B–4.

- **Bits 7-5** — Must be zero
- **Bits 4-0** — Command select code

**Table B–4, Command Line Select Codes**

| CMD<4> | CMD<3> | CMD<2> | CMD<1> | CMD<0> | HEX CODE | CMD |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 00 | NOP |
| 0 | 0 | 0 | 0 | 1 | 01 | CLR |
| 0 | 0 | 0 | 1 | 0 | 02 | DMS |
| 0 | 0 | 0 | 1 | 1 | 03 | SNS |
| 0 | 0 | 1 | 0 | 0 | 04 | RDF |
| 0 | 0 | 1 | 0 | 1 | 05 | RDB |
| 0 | 0 | 1 | 1 | 0 | 06 | WRT |
| 0 | 0 | 1 | 1 | 1 | 07 | LWR |
| 0 | 1 | 0 | 0 | 0 | 08 | BSF |
| 0 | 1 | 0 | 0 | 1 | 09 | BSB |
| 0 | 1 | 0 | 1 | 0 | 0A | FSF |
| 0 | 1 | 0 | 1 | 1 | 0B | FSB |
| 0 | 1 | 1 | 0 | 0 | 0C | WTM |
| 0 | 1 | 1 | 0 | 1 | 0D | ERG |
| 0 | 1 | 1 | 1 | 0 | 0E | REW |
| 0 | 1 | 1 | 1 | 1 | 0F | RUN |

## B.5 DMA Control Bytes — Bytes 05–07, 08–0A, 0C–0E

The DMA Control bytes provide a starting byte address and byte count to the VBTC for data transfers. The byte count is assumed to be a positive integer and is the actual number of bytes to be transferred.

A VMEbus write to these registers loads the pending level only; it has no effect until the pending level is transferred to the execution level. Loading the execution level does not alter or destroy the contents of the pending level.

The DMA control bytes are organized into three fields of three bytes each:

- **Byte transfer count** — Bytes 05-07
- **Starting byte transfer address register A** — Bytes 08-0A
- **Starting byte transfer address register B** — Bytes 0C-0E

## B.6 Clear Interrupt — Byte 0B

Writing any value to the Clear Interrupt byte address clears the VBTC's interrupt request flip-flop.

## B.7 Board Reset — Byte 0F

The Board Reset byte resets the VBTC. A reset is performed by writing a 1 to this register, followed by a 0.

# Appendix C
# Execution Level Registers

## C.1 Overview

This appendix describes the execution level registers. The VIOP, under software device driver control, can only read the execution level registers to obtain VBTC status.

The layout of the execution level registers is shown in Table C–1. All of the register addresses listed in Table C–1 are hexadecimal offsets from the VBTC base address. Individual byte and bit descriptions of the execution level registers are provided in subsequent sections.

**Table C–1, Execution Level Registers**

| ADDR | MSB | | | | | | | LSB |
|---|---|---|---|---|---|---|---|---|
| 10 | Cmd Exec | Cmd Pend | Int Req | Next Buf Ptr | Int Ena | Diag Loop Ena | Force Par Err | Chain Mode |
| 11 | Transfer Buffer Current Address Byte 2 | | | | | | | |
| 12 | Transfer Buffer Current Address Byte 1 | | | | | | | |
| 13 | Transfer Buffer Current Address Byte 0 | | | | | | | |
| 14 | Current Transfer Count Byte 2 | | | | | | | |
| 15 | Current Transfer Count Byte 1 | | | | | | | |
| 16 | Current Transfer Count Byte 0 | | | | | | | |
| 17 | Exec Tus2 | Exec Tus1 | Exec Tus0 | Exec Cmd4 | Exec Cmd3 | Exec Cmd2 | Exec Cmd1 | Exec Cmd0 |
| 18 | Exec Den1 | Exec Den2 | File Mark | Read Under Run | Cmd Abrt | VME Bus Error | Read Par Error | Tape Error |
| 19 | Cmd Inc | Cmd Reject | Over Run | Data Check | Prom Error | Corr Error | Tape Dbus Parerr | Resv |
| 1A | Online Status | Ready Status | Write Status | Slave Status Change | End of Tape | Begin of Tape | File Protect | Rewind Status |
| 1B | Reserved | | | | | | | |
| 1C | Residual Transfer Count Byte 2 | | | | | | | |
| 1D | Residual Transfer Count Byte 1 | | | | | | | |
| 1E | Residual Transfer Count Byte 0 | | | | | | | |
| 1F | Reserved | | | | | | | |

# C.2 VBTC Status Byte — Byte 10

The contents of the VBTC Status byte describe the state of the VBTC and are independent of the specific tape units attached to it. Each bit in the VBTC Status byte is described below.

## C.2.1 Command in Execution — Bit 7

If the Command in Execution bit is set, it indicates that a command is currently executing.

## C.2.2 Command Pending — Bit 6

If the Command Pending bit is set, it indicates that a command has been loaded into the pending level registers and will begin executing if the currently executing command completes without error.

## C.2.3 Interrupt Requesting — Bit 5

The Interrupt Requesting bit is set when the VBTC is currently requesting an interrupt from the VIOP. This bit is reset by writing to the pending level clear interrupt register.

## C.2.4 Next Buffer Pointer — Bit 4

If the Next Buffer Pointer bit is set, it indicates that the next transfer address parameter will be loaded from the pending level *A* transfer buffer. This bit is always set during normal blocked transfers; the *B* transfer buffer is used only for chain mode transfers. During chain mode transfers, the Next Buffer Pointer bit will toggle (set and reset) each time the execution level is reloaded. This bit has no meaning except during chain mode operations.

## C.2.5 Interrupt Enable — Bit 3

If the Interrupt Enable bit is set, it indicates that an interrupt request will be generated upon completion of the currently executing command.

## C.2.6 Diagnostic Loopback Enable — Bit 2

If the Diagnostic Loopback Enable bit is set, the execution level is operating in *diagnostic wrap-around mode.* Direct Memory Access (DMA) transfers may be executed to and from the VBTC's FIFO buffer, but will not be transferred to or from the attached tape drive.

## C.2.7 Force Parity Error — Bit 1

If the Force Parity Error bit is set, *even* parity will be generated and checked for on data transfers. Since the tape drive formatter checks for *odd* parity, a tape databus parity error will be generated if the tape drive parity circuits are operating correctly. Reading valid data from

tape will generate a read parity error when this bit is set.

### C.2.8 Chain Mode — Bit 0

If the Chain Mode bit is set, the VBTC is operating in chain mode. This bit will reset if a chain buffer completes before the previous chain buffer's interrupt is cleared.

## C.3 DMA Status Bytes — Bytes 11–16

The DMA Status bytes contain the current Transfer Address and Transfer Count parameters. The six DMA Status bytes are described below.

### C.3.1 Transfer Address Bytes — Bytes 11-13

The three Transfer Address bytes contain an address pointer to the next location to be accessed by the DMA channel. In case of an error, the transfer address will be frozen at the location of the error. This register is not cleared by a reset.

### C.3.2 Transfer Count Bytes — Bytes 14-16

The three Transfer Count bytes contain the number of bytes remaining to be transferred by the current DMA operation. If an error occurs, the transfer count is frozen at the value in place when the error occurred. The transfer in error is ignored; therefore, it is included in the frozen transfer count. This register is not cleared by a reset.

## C.4 Command Status Byte — Byte 17

The Command Status byte contains the execution level tape drive address and command.

### C.4.1 Tape Drive Address — Bits 7-5

The Tape Drive Address bits contain the address of the drive or formatter currently being addressed by the VBTC. Only the two least significant bits are meaningful if STC 2921 or 2922 drives are attached.

### C.4.2 Tape Drive Command — Bits 4-0

The Tape Drive Command bits contain the last command presented to the tape drive referenced by the tape drive address.

# C.5 Density and Error Status Byte — Byte 18

The Density and Error Status byte contains tape density status and VBTC-generated error status. Each bit in the Density and Error Status byte is described below.

## C.5.1 Density Select Field — Bits 7-6

The Density Select Field bits contain the density select field value most recently sent to the formatter. This value is not necessarily the density select currently in use; the formatter will ignore this control field, *except* when it begins a write operation at Beginning Of Tape (BOT), and then only if software control of density is enabled.

The current density select information is located in the mux bytes.

## C.5.2 File Mark Status — Bit 5

If the File Mark Status bit is set, it indicates that the tape is positioned just past a tape mark.

## C.5.3 Read Under Run — Bit 4

If the Read Under Run bit is set, it indicates that the number of bytes requested in the Transfer Count (TC) register was less than the actual number of bytes in the record during a tape read operation.

## C.5.4 Command Aborted — Bit 3

If the Command Aborted bit is set, it indicates that a pended command was aborted due an error during execution of the preceding command. This bit will be reset by the execution of a new command or by a reset.

## C.5.5 VMEbus Timeout Error — Bit 2

If the VMEbus Timeout Error bit is set, it indicates that a VMEbus error occurred during an attempted DMA transfer. The Transfer Address (TA) and Transfer Count (TC) parameters are frozen at their values that existed at the point the timeout error occurred. This bit will be reset by execution of a new command. An interrupt is not issued when this error occurs.

## C.5.6 Read Parity Error — Bit 1

If the Read Parity Error bit is set, it indicates that the VBTC detected bad parity on a data byte received from the FIFO buffer. This bit will be reset by execution of a new command.

### C.5.7 Tape Error — Bit 0

If the Tape Error bit is set, it indicates that the tape formatter has detected an error. To determine the nature of the error, tape status must be examined. This bit will be reset when the error condition no longer exists.

## C.6 Formatter Error Status Byte — Byte 19

The Formatter Error Status byte contains formatter error status for the last executed command. This information is latched on the trailing edge of *formatter busy* signal. If an error occurred, any pending operation will be aborted to prevent overwriting status information. Each bit in the Formatter Error Status byte is described below.

### C.6.1 Command Incomplete — Bit 7

If the Command Incomplete bit is set, it indicates the formatter and tape drive initiated the previous command, but the operation was not completed.

### C.6.2 Command Reject — Bit 6

If the Command Reject bit is set, it indicates the formatter or tape drive was unable to initiate or complete the previous command.

### C.6.3 Overrun — Bit 5

If the Overrun bit is set, it indicates the VBTC was unable to transfer data to or from the tape drive at the required rate, and data was lost during the transfer.

### C.6.4 Data Check — Bit 4

If the Data Check bit is set, an error has occurred. The formatter status mux bytes must be examined to determine the nature of the error.

### C.6.5 PROM Error — Bit 3

If the Prom Error bit is set, it indicates the formatter control store parity checker detected an error during execution of the previous command.

However, if an STC 2921 or 2922 tape drive is connected to the VBTC, it indicates that a *checksum error* was detected in the control store of the STC 2921 or 2922 drive during power-up self-test.

### C.6.6 Correctable Error — Bit 2

If the Correctable Error bit is set, it indicates that a correctable data error was detected. The uncorrectable error status bit in the mux byte will also be set if the correction attempt failed.

### C.6.7 Data Bus Parity Error — Bit 1

If the Data Bus Parity Error bit is set, it indicates that a parity error occurred on the bi-directional data bus between the formatter and the tape drive. This error is, by definition, *not* correctable.

### C.6.8 Reserved — Bit 0

Bit 0 of this byte is reserved.

## C.7 Real Time Drive Status Byte — Byte 1A

The Real Time Drive Status byte contains real time status information received from the tape drive. This information is valid anytime the tape drive is selected. (These status signals are also present in the tape drive status mux.)

Each bit in the Real Time Drive Status byte is described below.

### C.7.1 Online — Bit 7

The Online bit is set by the tape drive when it is in *online* status. If the tape drive is *not* online, all other status is invalid. A tape drive accepts commands only when it is online.

### C.7.2 Ready Bit — Bit 6

The Ready bit is set when the tape drive is in *ready* status. The tape drive is in *ready* status when the tape is loaded; but *not* when rewinding or when in *machine check* status.

### C.7.3 Write — Bit 5

The Write bit is set when the tape drive is in *write* status.

### C.7.4 Slave Status Change (SSC) — Bit 4

The Slave Status Change (SSC) bit is set by the tape drive to indicate that it has gone online, offline, or gone from *not ready* to *ready*. The SSC bit is reset when the tape drive accepts any command other than No-Operation (NOP) or Sense Drive Status (SNS).

### C.7.5 End of Tape — Bit 3

The EOT bit is set when the tape is positioned at the End Of Tape (EOT) marker.

### C.7.6 Beginning of Tape — Bit 3

The BOT bit is set when the tape is positioned at the Beginning Of Tape (BOT) marker.

### C.7.7 File Protect — Bit 1

The File Protect bit is set when a tape has been loaded and the tape reel does not contain a write enable ring.

### C.7.8 Rewind — Bit 0

The Rewind bit is set when the tape drive is rewinding to the BOT.

## C.8 Residual Transfer Count Bytes — Bytes 1C–1E

The Residual Transfer Count bytes contain the byte count that was left in the transfer count registers (bytes 14–16) after a read or write command terminated. These registers are useful when pipelining read commands. When a read command is issued to the VBTC, the VBTC continues reading until the transfer count is zero or until the end of a record.

If the transfer count is greater than the actual number of bytes in a record, the transfer count that existed at the end of the record is copied to the Residual Transfer Count registers. Since the Transfer Count registers are written over with each loading of the execution level registers, software can determine the actual byte count of the just completed read operation from the Residual Transfer Count registers.

THIS PAGE INTENTIONALLY LEFT BLANK

# Appendix D
# Error Mux Status Bytes

## D.1 Overview

This appendix contains drive-specific information about the failures indicated in the Error Status byte of the execution level registers.

The method used to multiplex the VBTC status bytes varies among the formatters. The status bytes of the various formatters used with the VBTC are described below.

### D.1.1 STC 2921 and 2922

Four status bytes are available on the STC 2921 and 2922 drives by addressing the appropriate status byte in the VMEbus I/O register space. Bit <9> of these status bytes (parity) is folded into Bit 0 of the immediately succeeding byte.

The error mux byte and bit layouts for the STC 2921 and 2922 formatters are shown in Table D-1:

### D.1.2 Fujitsu 2436

The Fujitsu 2436 drive uses the same four status bytes as defined for the STC 2921 and 292x formatters. Sixteen additional *Drive Status* bytes are available by successively executing an *SNS* command and reading Mux Byte 3. Refer to the *Fujitsu 243XL/R Magnetic Tape Subsystem Installation Instructions* manual for additional information.

The error mux byte and bit layouts for the Fujitsu 2436 formatter are shown in Table D-1:

## Table D–1, STC 292x and Fujitsu 2436 Error Mux Status Bytes

| ErrMux Byte | ErrMux Bit | Signal Description | VBTC Register Address |
|---|---|---|---|
| 0 | 0 | Dead track, Bit 0 | Base Address + 20 |
| 0 | 1 | Dead track, Bit 1 | |
| 0 | 2 | Dead track, Bit 2 | |
| 0 | 3 | Dead track, Bit 3 | |
| 0 | 4 | Dead track, Bit 4 | |
| 0 | 5 | Dead track, Bit 5 | |
| 0 | 6 | Dead track, Bit 6 | |
| 0 | 7 | Dead track, Bit 7 | |
| 0 | P | Dead track, Parity Bit | Base Address + 21 |
| 1 | 0 | Diagnostic mode latch | Base Address + 22 |
| 1 | 1 | Velocity error | |
| 1 | 2 | End of data check | |
| 1 | 3 | Reserved | |
| 1 | 4 | Multiple track error | |
| 1 | 5 | Partial record | |
| 1 | 6 | Uncorrectable error | |
| 1 | 7 | Write tape mark check | |
| 1 | P | Cyclic redundancy character error | Base Address + 23 |
| 2 | 0 | DA0 | Base Address + 24 |
| 2 | 1 | DA1 | |
| 2 | 2 | DA2 | |
| 2 | 3 | DA3 | |
| 2 | 4 | DA4 | |
| 2 | 5 | DA5 | |
| 2 | 6 | DA6 | |
| 2 | 7 | DA7 | |
| 2 | P | Tachometer | Base Address + 25 |
| 3 | 0 | Online status | Base Address + 26 |
| 3 | 1 | Ready status | |
| 3 | 2 | High density select <1> | |
| 3 | 3 | Reverse status | |
| 3 | 4 | File protect status | |
| 3 | 5 | Reserved | |
| 3 | 6 | BOT status | |
| 3 | 7 | EOT status | |
| 3 | P | Write status | Base Address + 27 |

### D.1.3 STC 1968 Error Mux Bytes

Five mux bytes are available to the STC 1968 drive by addressing the appropriate status byte in the VMEbus I/O register space. Byte 4 contains CRC information unique to the STC 1968 drive.

Refer to the *STC 1960 Maintenance Manual* for additional information.

The error mux byte and bit layouts for the STC 1968 formatter are shown in Tables D–2 and D–3:

## Table D–2, STC 1968 Error Mux Status Bytes

| ErrMux Byte | ErrMux Bit | Signal Description | VBTC Register Address |
|---|---|---|---|
| 0 | 0 | Dead track, Bit 0 | Base Address + 20 |
| 0 | 1 | Dead track, Bit 1 | |
| 0 | 2 | Dead track, Bit 2 | |
| 0 | 3 | Dead track, Bit 3 | |
| 0 | 4 | Dead track, Bit 4 | |
| 0 | 5 | Dead track, Bit 5 | |
| 0 | 6 | Dead track, Bit 6 | |
| 0 | 7 | Dead track, Bit 7 | |
| 0 | P | Dead track, Parity Bit | Base Address + 21 |
| 1 | 0 | Diagnostic mode latch | Base Address + 22 |
| 1 | 1 | Velocity error | |
| 1 | 2 | End of data check | |
| 1 | 3 | Reserved | |
| 1 | 4 | Multiple track error | |
| 1 | 5 | Partial record | |
| 1 | 6 | Uncorrectable error | |
| 1 | 7 | Write tape mark check | |
| 1 | P | Cyclic redundancy character error | Base Address + 23 |
| 2 | 0 | DA0 | Base Address + 24 |
| 2 | 1 | DA1 | |
| 2 | 2 | DA2 | |
| 2 | 3 | DA3 | |
| 2 | 4 | DA4 | |
| 2 | 5 | DA5 | |
| 2 | 6 | DA6 | |
| 2 | 7 | DA7 | |
| 2 | P | Tachometer | Base Address + 25 |
| 3 | 0 | Online status | Base Address + 26 |
| 3 | 1 | Ready status | |
| 3 | 2 | High density select <1> | |
| 3 | 3 | Reverse status | |
| 3 | 4 | File protect status | |
| 3 | 5 | Reserved | |
| 3 | 6 | BOT status | |
| 3 | 7 | EOT status | |
| 3 | P | Write status | Base Address + 27 |

## Table D–3, STC 1968 Error Mux Status Bytes, cont'd

| ErrMux Byte | ErrMux Bit | Signal Description | VBTC Register Address |
|---|---|---|---|
| 4 | 0 | CRC-F | Base Address + 28 |
| 4 | 1 | CRC-F | |
| 4 | 2 | CRC-F | |
| 4 | 3 | CRC-F | |
| 4 | 4 | CRC-F | |
| 4 | 5 | CRC-F | |
| 4 | 6 | CRC-F | |
| 4 | 7 | CRC-F | |
| 4 | P | CRC-F | Base Address + 29 |

# Appendix E
# VBTC Configurator Document

This appendix contains a copy of the VMEbus Tape Controller Configurator document.

> **NOTE**
>
> The *VMEbus Tape Controller Configurator* document contains basic configuration information for the VBTC. In the event of changes regarding VBTC configuration, an updated version of the document will be made available. Configurator document updates should be inserted into this appendix.

THIS PAGE INTENTIONALLY LEFT BLANK

# VME Bus Tape Controller Configurator

## 1.1 Scope

The purpose of this document is to provide configuration information for the VBTC (Convex P/N 410-001152-200).

## 1.2 Configuration

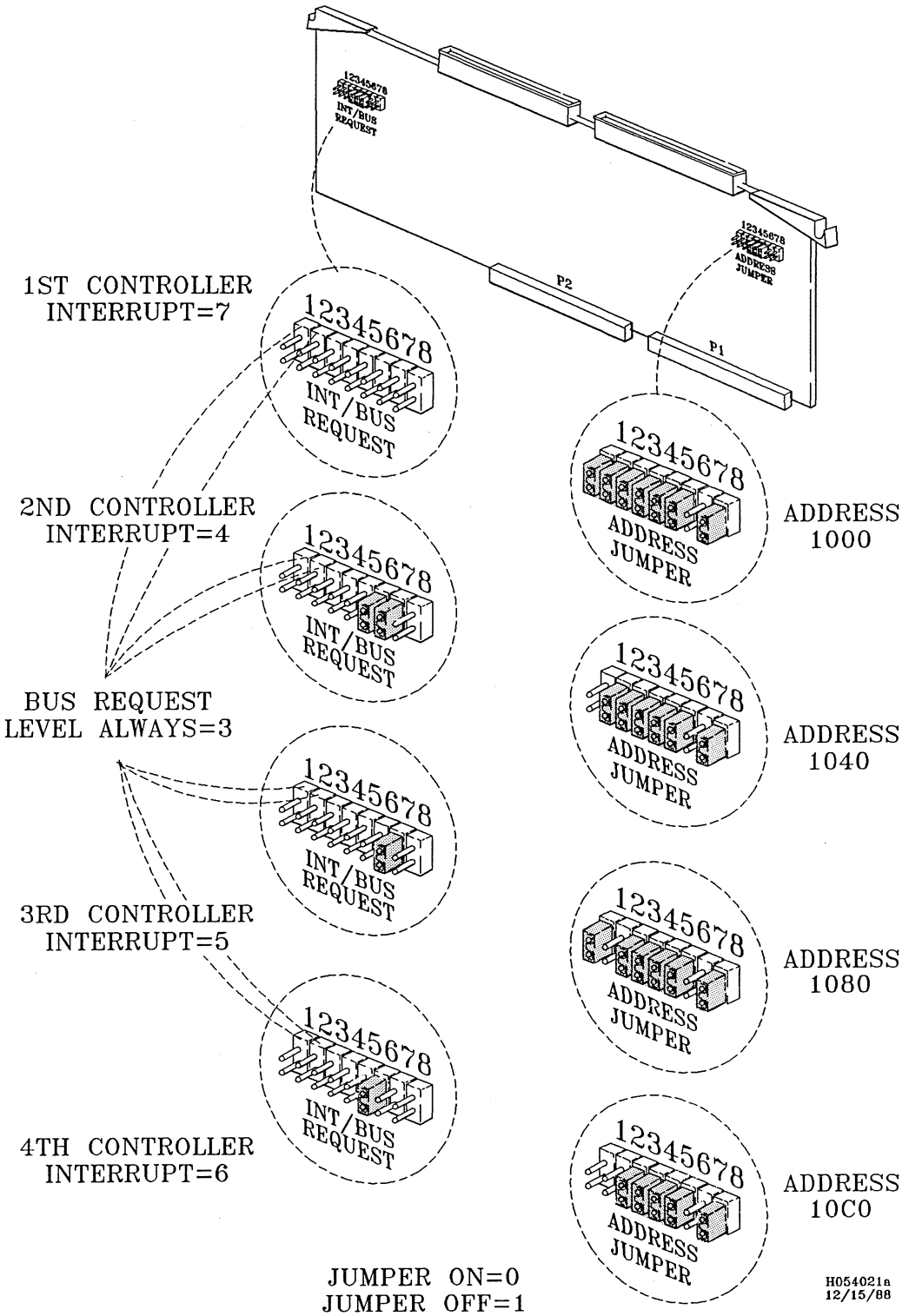
The VBTC has two jumper blocks which are used to set the boards base address, interrupt level, and bus request level. The jumper blocks are defined below.

| Jumper block #1 - Upper left side of board | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Address | A6 | A7 | A8 | A9 | A10 | A11 | A12 | A13 |
| Jumper | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

| Jumper block #2 - Upper right side of board | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Lower Nibble Bus Request | | | | Upper Nibble Interrupt Request | | | |
| Bit | BR0 | BR1 | XX | XX | XX | INT0 | INT1 | INT2 |
| Jumper | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Assigned address and interrupt levels for the VMEbus tape controller are as follows. See the figure on page 2 of this document. Interrupt levels for the second, third, and forth controllers are not assigned. They should be set upon installation in the target VMEbus to a unique level.

| Standard Address and Interrupt levels | | | |
|---|---|---|---|
| Controller # | Address | Interrupt Level | Bus Request Level |
| 1 | 0x1000 | 7 | 3 |
| 2 | 0x1040 | None Assigned | 3 |
| 3 | 0x1080 | None Assigned | 3 |
| 4 | 0x10C0 | None Assigned | 3 |






| | | |
|---|---|---|
| CONVEX | TITLE: VBTC Configurator | REV: B.0 1/07/89 |
| | | ENGR: Rob Carruthers |
| | Part Number: 410-001152-600 | PAGE: 2 |

# Appendix F
# Reporting Problems

## F.1 Overview

The *contact* utility is the recommended way to report minor hardware deficiencies and technical documentation problems to the Technical Assistance Center (TAC). This utility is an interactive tool that prompts the user for the information to properly file a problem report.

> NOTE
>
> The *contact* utility is not intended for requesting customer service for hardware failures. To restore your CONVEX equipment to operational status, faster service can be obtained by directly telephoning the TAC (refer to "Technical Assistance" in the preface).

To use the *contact* utility, there must be a phone connection to the TAC. A UNIX-to-UNIX Communication Protocols (UUCP) allows communication between UNIX systems by either dial-in or hard-wired communication lines. For more information, refer to *uucp(1)* or to the *info(1)* entry in the UNIX man pages.

The name and version number of the product involved is required. Use the *vers* command to ascertain the program or utility name and version. The syntax for the command is **vers** *filename*, where *filename* is the full pathname of the program. If the full pathname of the program is not known, enter **which** *program*. For more information, refer to the *vers(1)* and *which(1)* entries in the UNIX man pages.

## F.2 Information Required to Report a Problem

The *contact* utility requires the following information:

1. The customer name, title, phone number, and corporate name
2. The hardware nomenclature, part number, and revision level, or the technical manual name, document number, and version

> NOTE
>
> Use *vers* and *which* to identify product name and version.

3. A short (one line) summary of the problem
4. The more information provided, the more quickly the problem can be isolated and solved. At a minimum, include a detailed description of the problem (including page references, if applicable), the source code, and a stack backtrace whenever possible.

> **NOTE**
>
> See the *adb(1)* or *csd(1)* man pages for information on obtaining stack backtraces.

5. The priority of the problem, selected from a list of six levels
6. Instructions on how to reproduce the problem, including the command syntax used, any flags invoked, or anything else attempted to make the program run
7. Any other comments about the problem or files to be submitted

The *contact* user has a chance to review and edit the report prior to submitting it. If the user decides to delay submitting the report, the session can be aborted. The report is automatically saved in the user's top-level directory in a file named *dead.report*.

See the following figure for a sample *contact* session. User input is in bold lettering, and the system response is in monospace type.

## Figure F–1, Sample *contact* Session

```
%contact (RETURN)
Welcome to contact version 0.11 ()

Enter your name, title, phone number, and corporate name (^D to terminate)
> Margaret Atwood, systems programmer, 814-4444, University r
> of Chicago (RETURN)
> (CTRL-D)

Enter the name of the product involved
> CONVEX UNIX Programmer's Manual, Part I (RETURN)

Enter the version number (in the form X.X or X.X.X.X) of the product
> Revision 4.0 (RETURN)

Enter a short (1 line) summary of the problem
> The finger command manual page lists nonexistent bug (RETURN)

Enter a detailed description of the problem (^D to terminate)
> The finger(1) man page says, under the BUGS section, that "Only the first
line of the .project file is printed."  Happily, this is not true! (RETURN)
> (CTRL-D)

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

Enter the instructions by which the problem may be reproduced (^D to terminate)
> a) put more than one line in .project (RETURN)
> b) read the man page for finger(1) (RETURN)
> (CTRL-D)

Enter any comments that are applicable (^D to terminate) (RETURN)
> (CTRL-D)

Do you have any suggestions or comments on the documentation that you
referenced when you were trying to resolve your problem (for example,
additions, corrections organization, accessibility)? (^D to terminate)
> The man page should be updated. (RETURN)
> (CTRL-D)

Are there any files that should be included in this report (yes | no)?
> no (RETURN)

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

Problem report submitted.
%
```

THIS PAGE INTENTIONALLY LEFT BLANK

# Index

## L



## M



## N



## O



## P



## R



## S



## T



## U



## V

## W

THIS PAGE INTENTIONALLY LEFT BLANK

# Electronic Mail

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

```
cnvxhwdoc@convex.COM
```

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

# What to Include in an Email Message

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

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

# Reader's Forum

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

THIS PAGE INTENTIONALLY LEFT BLANK

CONVEX VMEbus Tape Controller
Service Guide
081-001330-200, First Edition

## Reader's Forum

_____

_____

_____

_____

_____

_____

_____

_____

**From:**

Name_____Title_____

Company_____Date_____

Address and Phone No._____

**FOR ADDITIONAL INFORMATION OR DOCUMENTATION:**

| Location | Phone Number |
|---|---|
| In Texas | (214)952-0200 |
| Other continental locations | 1(800)952-0379 |
| Outside continental US | Contact local CONVEX office |

Direct mail orders to:

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

(Fold Here First)


NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

# BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1046 RICHARDSON, TEXAS

POSTAGE WILL BE PAID BY ADDRESSEE

CUSTOMER SERVICE
CONVEX Computer Corp.
P.O. Box 833851
Richardson, TX 75083-3851

(Fold Here Second)

(Tape or Staple)