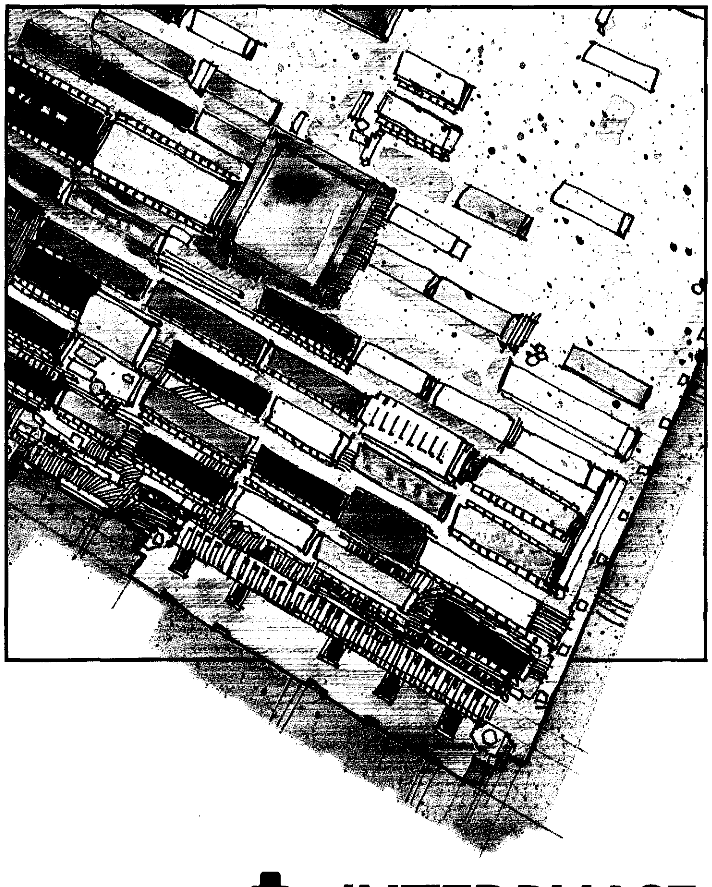
V/ESDI 4201 Panther

High-performance VMEbus Enhanced Small Device Interface (ESDI) Disk Controller

User's Guide



INTERPHASE
CORPORATION

V/ESDI 4201 Panther

High-performance VMEbus Enhanced Small Device Interface (ESDI) Disk Controller

User's Guide

Document Number UG-0740-000-040 Revision 040 Follows Revision X0F

Release Date October, 1988

CAUTION

Specific configurations of the V/ESDI 4201 use rows A & C of the connector on P2 for I/O signals. To avoid possible damage to this board, the backplane, or other system components, verify compatibility with the system before installing this board and applying power.

FOR ASSISTANCE IN USING THE V/ESDI 4201 OR ANY

OTHER INTERPHASE PRODUCT CALL:

INTERPHASE APPLICATIONS ENGINEERING DEPARTMENT

(214) 350-9000 919-9000

IN THE U.K., CALL:

(0296) 435661

Information in this user document supersedes any preliminary specifications and/or data sheets that may have been made available. Every effort has been made to supply accurate and complete information. However, Interphase Corporation assumes no responsibility for its use. In addition, Interphase reserves the right to make product improvements without prior notice. Such improvements may include, but are not limited to, command codes and error codes.

NOTE

This document is confidential proprietary information of Interphase Corporation and shall not be disclosed or copied or used for any purpose other than for which it is specifically furnished without the prior written consent of Interphase Corporation.

TABLE OF CONTENTS

1. INTRODUCTION TO THE V/ESDI 4201

2. INSTALLATION OF THE V/ESDI 4201

3. INITIALIZING THE V/ESDI 4201

4. NON-MACSI OPERATION OF THE V/ESDI 4201

5. INPUT/OUTPUT PARAMETER BLOCK (IOPB)

6. MULTIPLE ACTIVE COMMAND SOFTWARE INTERFACE (MACSI)

FIGURES

TABLES

1 INTRODUCTION

WARNING

Catastrophic DAMAGE can result if improper connections are made. Therefore, those planning to connect power sources to the VMEbus for the purpose of feeding the user defined pins of P2 (Rows A and C) should FIRST CHECK to ensure that all boards installed are compatible with those connections.

The Panther offers two modes of operation: Non-MACSI, which is explained in this user's guide primarily for backwards compatibility with previous versions of Interphase drivers and products; and MACSI, the command queing mode of operation, which offers the best possible performance from the board.

Non-MACSI operation is explained in Chapter 4. Input/Output Parameter Blocks are explained in Chapter 5 and MACSI operation is explained in Chapter 6.

Overview

The V/ESDI 4201 Panther is an extremely fast, intelligent, Enhanced Small Device Interface (ESDI) controller/formatter used to interface VMEbus-based systems to ESDI compatible disk drives. The V/ESDI 4201 can support any combination of four ESDI drives, hard or soft sectored.

Its intelligence is based on a M68000 16/32 bit microprocessor running with no wait states. The real key to the V/ESDI 4201, however, is the proprietary BUSpacket Interface^{SM} and the Multiple Active Command Software Interface (MACSI).

The V/ESDI 4201 achieves a throughput far superior to that of any competitive controller by sending preformatted data packets across the VMEbus at rates up to 30 megabytes per second.

BUSpacket Interface

It is the BUSpacket Interface of the V/ESDI 4201 that enables it to operate at speeds of 30 megabytes/second on the VMEbus. Previous generations of disk controllers transferred data across the bus as fast as the internal buffer, the bus interface and the system memory would allow, but this generally resulted in data rates in the five to ten megabytes/second range. The BUSpacket Interface bursts preformatted data packets across the bus as fast as the system memory will allow.

With a bus data rate up to 30 megabytes/second the V/ESDI 4201 comes close to fully utilizing the 40 megabytes/second theoretical bandwidth of the VMEbus. This high bus data rate means that the V/ESDI 4201 spends less time on the bus than other controllers, conserving bus bandwidth for other processes.

There are two keys to the speed of the BUSpacket Interface: a deep FIFO buffer and a delay-line-based asynchronous state machine.

The FIFO Buffer

The deep FIFO buffer of the BUSpacket Interface is a very high speed 512-byte FIFO that can write a BUSpacket to (or read a BUSpacket from) the Panther's 128 Kbyte buffer with a fast block transfer. The FIFO itself is very fast so that once it is connected to the VMEbus, the DMA can proceed at 30 megabytes/second. Thus, the speed of the controller (on the VMEbus) is totally decoupled from the speed of the 128K RAM buffer and the peripheral device attached to the controller.

During a read from the disk, for example, the FIFO buffer is filled with a packet of data from the buffer memory prior to the controller acquiring the bus. Once on the bus, the packets are burst across the bus (in DMA mode). When the transfer is complete the bus is immediately released for other uses. This is continued until all of the information requested in the read or write command has been transferred to/from system memory. During a DMA operation to/from the BUSpacket FIFO no other functions (such as caching) are impeded.

The Asynchronous State Machine

The V/ESDI 4201 uses an asynchronous state machine (in lieu of a traditional variety) in the BUSpacket Interface to control the DMA transfer of data to/from the VMEbus.

Synchronous state machines, which are typically used to control the entire DMA operation, must first synchronize the inherently asynchronous VMEbus control signals to avoid metastable states. This synchronization cycle will, on the average, delay the signal by one and a half periods of the synchronizing clock. This means that a synchronous state machine running at 20 MHz, for example, will delay any control signal off of the bus by an average of 75 nanoseconds before it uses it as an input to the state machine. Such a delay is an unacceptable portion of the 100 nanosecond cycle time required to achieve a 40 megabyte/second data rate on the VMEbus.

In contrast, the high-speed DMA transfer of the BUSpacket Interface on the V/ESDI 4201 is controlled by an asynchronous state machine that runs off of a tapped delay line. A tapped delay-line-based state machine avoids metastable states without incurring the delays caused by synchronizing the VMEbus signals. Since the VMEbus is an asynchronous bus, the tapped delay line state machine can essentially drive the bus as fast as system memory allows.

Virtual Buffer Architecturesm

The M68000, running under a proprietary, multitasking real time firmware operating system, is used to regulate controller activity, such as sequencing commands and setting up data transfer operations. Its most important function, however, is to manage the pool of virtual buffers that is composed from the V/ESDI 4201's 128 Kbytes of RAM.

At any given time individual buffers may be allocated to either the disk, the VMEbus, or the cache. The 68000 microprocessor will dynamically allocate and deallocate buffers as they are requested or released by the VMEbus or the disk. During a disk read, for example, the disk read task brings data into a free buffer. Simultaneously, as soon as any sector of interest has been captured, the bus write task begins moving data into system memory; thus, the sector buffer is freed. If the VMEbus cannot keep up with the drive, the buffers will hold the data until the bus is ready. The buffer thus prevents either overrun or underrun of data. The V/ESDI 4201 is able to apply many high-performance techniques which can reduce or eliminate disk rotational latencies. These methods include zero latency reads and writes, and intelligent caching.

Zero Latency

A traditional controller, upon receiving a multisector request from the operating system, will wait until it encounters the first required sector before beginning to read and transfer the data. Thus, it will incur an average rotational latency equivalent to one half of one track. If the request is for a full track of data, even with a 1:1 interleave factor, the traditional controller will take an average of one-and-half revolutions to complete the transfer.

The V/ESDI 4201 begins reading data as soon as the heads land on the track (i.e., zero rotational latency) and begins transferring data as soon as it encounters a sector of interest; it does not wait to rotate around to the beginning of the requested string. In this manner, the V/ESDI 4201 will never require more than a single revolution to transfer an entire track of data.

Zero latency writes are accomplished using the same principle. This technique is most effective for large disk transactions which may access many sectors of data per transaction.

Prefetch Caching

To further reduce latency, the V/ESDI 4201 uses an intelligent caching scheme (prefetch caching) to anticipate which data sectors will be requested next. When the V/ESDI 4201 has completed a read operation and has transferred the requested data, it will continue to read sectors into the cache until all available buffers have been filled, or until it receives a command from the host requesting a head movement. And with the large 128 Kbyte RAM buffer, the V/ESDI 4201 can even cache across track boundaries. Thus, if subsequent requests from the operating system are for sectors logically contiguous with the previous sectors, these requests can be satisfied directly from the cache without having to access the disk. In disk intensive applications this can greatly improve overall system throughput.

This form of caching is particularly useful for UNIX and UNIX-like operating systems. Tests have shown improvements in disk operation averaging greater than 40 percent over operation with 1:1 interleave only. Caching shows the greatest improvement when disk activity is characterized by a large number of short transactions.

V/ESDI 4201 Optional SCSI Port

The V/ESDI 4201 can be equipped with an optional SCSI port to provide convenient archiving and tape backup capabilities. This addition allows the V/ESDI 4201 to control not only four ESDI disk drives, but at the same time, to control up to seven SCSI devices.

When a V/ESDI 4201 is equipped with a SCSI port, the Short I/O memory format is modified to accomodate the addtional command set, status registers, et cetera, that are required for SCSI device operations. In addition, 16 Kbytes of the 128 Kbytes of on-board RAM are used as a buffer between the tape devices and system memory. This buffer enables the V/ESDI 4201 to operate concurrent bus and tape operations. So while the tape is processing a transaction, the V/ESDI 4201 is reading/writing to/from the buffer to prepare for the next tape operation. This aids in providing tape streaming on a command basis.

SCSI operation is detailed in the V/ESDI 4201 Panther SCSI Port supplemental user's guide, UG-0780-000-XXX.

UNIX is a registered trademark of AT&T Bell Laboratories.

V/ESDI 4201 Features And Functions

The V/ESDI 4201 provides many options which are either under software control or are mechanically selectable by switch or jumper. For example, sector sizes are programmable varying from 128 bytes to 2048 bytes. The most important features and functions of the V/ESDI 4201 are outlined as follow:

- The BUSpacket Interface boosts DMA throughput in excess of 30 megabytes per second.
- The V/ESDI 4201 is controlled using a simple macro-level software interface.
- The on-board M68000 16/32 bit microprocessor relieves the system CPUs of disk handling tasks.
- Optional SCSI port to control up to seven SCSI devices.
- The V/ESDI 4201 can control up to four ESDI disk drives and seven SCSI devices concurrently.
- The V/ESDI 4201 supports 8-, 16-, or 32-bit wide data transfers, and provides 16-, 24-, or 32-bit addressing capability.
- 128 Kbytes of memory is provided; most of which is treated as a pool of virtual buffers.
- The V/ESDI 4201 supports disk data rates up to 24 Mb/s with a 1:1 interleave factor.
- Virtual Buffer Architecture reduces or eliminates data transfer delays caused by disk rotational latency and data overrun/underrun.
- A prefetch caching scheme with dynamic buffer allocation and deallocation is provided. Caching algorithms are optimized for UNIX, RMX/86 and similar operating systems.
- Automatic error correction is provided using a 32-bit or 48-bit error correction code (ECC).
- The V/ESDI 4201 can control four ESDI drives (hard or soft sectored).
- Self-diagnostics are performed after each hardware power-up and each software reset.
- Scatter/Gather comands allow the user to place contiguous disk data in noncontiguous areas of system memory or vice versa.

RMX/86 is a trademark of Intel Corporation.

- Overlapped and implied seeks are supported.
- Seven software programmable interrupt levels are provided.
- Bus priority is jumper selectable from zero to three.
- Defective media replacement on a sector or track basis is provided when formatting the disk.
- The disk can be addressed by either physical or logical sectors.
- Zero latency reads and writes insure maximum throughput.
- An interrupt on drive status change is a programmable option for such applications as overlapped seeks.

Command Queuing

In a typical computer (Figure. 1-1) there is a host CPU, some system memory, and a peripheral controller. The host issues commands to the peripheral controller which in turn moves data from the peripheral into (or out of) system memory. The host begins an operation by issuing a command to the controller. If there is no command queue, the controller is idling until the "GO" bit is set in the command. Once the "GO" bit is set, the controller can "decode" the command and start the peripheral in operation. When the controller/peripheral pair finishes the operation (which will typically involve some data movement) the controller issues a command complete interrupt to the host. The host, after servicing the interrupt, starts the cycle over by issuing another command to the controller.

This scenario, with little modification, has been used by controller manufacturers for years. But as computer manufacturers begin pushing for better and better performance, the problems inherent in this scheme begin to seriously limit performance.

Specifically there are two "times" inherent in this serial"command - wait for response" scheme that limit performance: command response times and command parsing times. By overlapping command response and command parsing with peripheral operation, queuing schemes significantly improve system performance.

Command Response

In the typical system described above, it may be many microseconds before the "GO" bit in the next command can be set after the host receives the command complete interrupt from the previous command. The sequence is: the controller asserts the command complete interrupt, the host responds by doing a context switch, servicing the interrupt, loading the next command into the controller and setting the "GO" bit. While all of this is going on the controller and peripheral are idle. The controller cannot start processing a command until the "GO" bit of the command has been set.

With a command queue inside the controller the host can load a command into the controller and set the "GO" bit while the controller and peripheral are still processing the previous command. Thus all of the command response time is overlapped with actual peripheral activity. This not only improves performance of the controller/peripheral pair but it also removes time pressure from the operating system so that interrupt response times and context switch times are no longer in the "critical path" of controller operation. This may allow the system to operate the controller at a lower priority without noticeable degradation of peripheral performance.

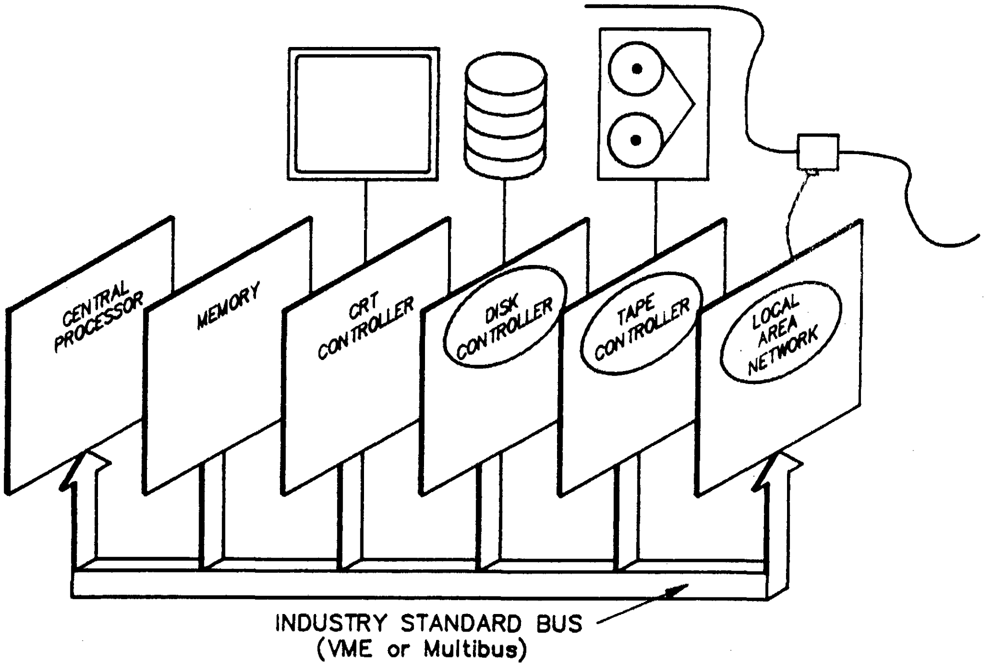


Figure 1-1. Typical Computer System

Command Parsing

Once the host has set the "GO" bit in a command the controller must parse (decode) the command before it can be issued to the peripheral. It is not unusual for this command parsing to take hundreds of microseconds. And in a conventional system, command parsing cannot be done until after the "GO" bit has been set.

A controller without a command queue cannot parse the command until the "GO" bit is set either. But with a command queue, the "GO" bit is set long before the command complete interrupt of the previous command is issued. Thus commands can be parsed while previous commands are being executed. This overlaps command parsing with peripheral operation.

Performance Advantages

In addition to the performance advantages gained by overlapping certain controller and host activity with peripheral activity, as described above, an efficient command queuing scheme can improve system performance by optimizing the manner in which the peripherals themselves are controlled. This is true of disk operations in particular, because they are subject to mechanical rotational and head movement delays. Thus improved operation can often be achieved by modifying the "way" in which commands are passed from the controller to the disk.

There are five generalized techniques that can be applied:

- Command Grouping
- Overlapped Seeks
- Command Sorting
- Search Cache on Reads
- Command Latency

Command Grouping

Command Grouping is a scheme whereby the controller searches the command queue for commands that can be "grouped" into a single command before issuing the single, concatenated command to the disk. Thus only a single command is issued to the peripheral. Upon completion of this one "macro" command, the controller reports command completion of each of the individual, original commands.

Overlapped Seeks

Once a controller has issued a command to a disk, if there is a seek implied in the command, there will be some seek delay before the head is on cylinder and the disk is capable of transferring data. By issuing seek commands to all of the disks at once these seek times are overlapped so that only one of the seek delays is actually incurred, the others are overlapped with data movement and other activity.

Command queuing facilitates the overlapping of seeks because it makes it possible for the controller to search through the queue and find the next command for each disk.

Command Sorting

Disk seek times can also be minimized if the controller sorts the commands beforehand so that the disk is always seeking a nearby cylinder. To ensure that seeks to "outside" cylinders do get issued, the typical algorithm will not allow the head travel to "turn around" until it gets to an extreme. Because this is similar to the method used by elevators in deciding which floor to stop at next, this scheme is called an "elevator sort" algorithm.

Command queuing facilitates this type of command sorting because the controller can easily sort through all of the command in the queue and then issue them to the disk in a prescribed order.

Search Cache on Reads

Disk seek times and rotational delays can be eliminated entirely from read commands if the controller can use data that has already been read into the onboard cache. If the controller finds that the read command is requesting data that is already in the cache, then the command can be satisfied immediately without any waiting. This will work only in RAM-based caching controllers. FIFO based controllers have obvious problems implementing any type of useful cache search on reads.

Command Latency

Not all command queue schemes are the same, and even if the controller has a command queue, there may be differences. For example, another key parameter that affects overall system operation is command queue latency. This is a measure of the minimum time required (for the host to wait) between issuing commands to the queue. In general this time will be overlapped with peripheral operation, but it still slows the host down if it must wait a long time before it can issue the next command to the controller.

Ideally the controller can accept commands as fast as the host can normally write to memory. Anything less than this slows the host down.

Other Considerations

In addition to these performance advantages, there are other considerations that affect how useful a command queuing scheme is in a given situation.

In general there should be some mechanism to tailor the queue depth to the application. Longer queues require more manipulation by the controller and are thus slower but a queue that is too short for the application may not offer significant advantage over a simpler nonqueuing scheme.

Occasionally there is a need to be able to issue a command that does not pass through the queue at all, or at least to issue one that is placed at the top of the queue.

In the final analysis there are good reasons for wanting a queuing mechanism that offers all of the performance advantages of short command queues and also provide the flexibility of multiple command queues. Interphase has developed Multiple Active Command Software Interface (MACSI) and has implemented it on several VMEbus controllers. It provides all of the performance advantages of simpler first generation command queuing schemes while still offering the power and flexibility that can only be provided by a multiple queue mechanism.

Utilization Of VMEbus Facilities

The evolution of control applications from 8- and 16-bit buses has progressed up to the development of the VMEbus. The VMEbus is the first bus structure to support a true 32-bit processor. It is comprised of separate 32-bit data and address lines that not only support 32-bit transactions, but also the traditional 8- and 16-bit transactions as well. This makes the VMEbus particularly appropriate for stand-alone or remote process controller applications. In addition, the VMEbus provides the flexibility for expansion within the microcomputer environment. Given all of these advantages, the V/ESDI 4201 is able to exploit the potential of the VMEbus and provide performance necessary in today's systems.

VMEbus Structure

There are three basic signals which determine the operation of the VMEbus: the Bus Request signal (for getting on the bus), the Bus Grant signal (for allocating the bus for use), and 32 bits of addressing. The addressing is supplemented with six address modifier bits which are used to partition memory spaces. The address modifiers are used to expand memory space by breaking it up into functional blocks.

The VMEbus has seven prioritized interrupts for fully vectored operation. This means that during an interrupt cycle, an interrupt vector provides the CPU with a unique value that describes the interrupt. In nonvectored systems, the CPU must search registers for the cause of the interrupt. The advantage of the vectored interrupt, therefore, is to greatly decrease the amount of time the CPU must spend servicing interrupts.

Data Transfers

All VMEbus data transfers are performed over the Data Transfer Bus (DTB). The DTB is 32 bits wide and is only used for data transfers. The VMEbus supports data transfers of eight, 16, or 32 bits. All data transfers take place between a bus master and a bus slave. The VMEbus supports multiple bus masters which provide the capability of moving large blocks of data without CPU intervention; thus, the CPU is not tied up for long periods of time during data transfers.

The V/ESDI 4201 uses the DTB to transfer commands, status and data between the host system, disk drive, and itself. The V/ESDI 4201 supports 8-, 16-, and 32-bit data transfers.

Bus Master

A DTB master can be any board that is capable of requesting the VMEbus. The V/ESDI 4201 performs all data transfers as a DTB master.

Bus Slave

A bus slave is a device that responds to the DTB master during data transfers. When commands are issued to the V/ESDI 4201, or status is read from the board, the V/ESDI 4201 is acting as a slave device.

Data Transfer Operations

During a data transfer cycle several things must happen . First, a bus master must request and be granted the bus. The master then puts the address of the slave to which the data will be transferred on the bus, and if it is a write operation, the data is also put on the bus. Since the VMEbus is completely asynchronous, the master must know when the address is valid and if the slave has either received the data or is ready for more data.

Once the address is valid, the master asserts an Address Strobe signal. Then, the master commands the data on the DTB bus with the Data Strobes. That tells the slave that the address is valid and the data is directed at that particular slave. Once the slave has received the data (write operation) or placed the data on the bus (read operation), the slave tells the master that the transfer is complete. To do this, the slave asserts a signal called Data Transfer Acknowledge (DTACK).

There are no restrictions on the VMEbus as to how long data transfers can be. The only real restriction is that the address strobe must precede the data strobe.

To start a command, a bus master (typically a CPU) will issue commands to the V/ESDI 4201 which is acting as a bus slave. After the V/ESDI 4201 has received the command and responded to it, it will then request the VMEbus, and when the bus is granted, the V/ESDI 4201 will transfer the data as a bus master.

VMEbus Addressing

The memory space accessible to the V/ESDI 4201 and the host CPU contains several subdivisions. Each subdivision requires a specific type of addressing. This partitioning of the memory space allows more efficient access and utilization of the memory. The addressing options include:

- ShortAddressing(16-bit)
- StandardAddressing(24-bit)
- ExtendedAddressing(32-bit)

The type of addressing is specified using a 6-bit address modifier code that also indicates the nature of the access. For example, address modifiers are commonly used to specify supervisory (restricted) memory access, and user (nonprivileged) memory access. The V/ESDI 4201 supports 16-, 24-, and 32-bit addressing as well as all address modifiers (as a bus master).

Short I/O

The V/ESDI 4201 requires that all data transfers of control information, command parameters, and status that occur with the host CPU acting as the bus master must be done through the short address space using either address modifiers 29 (user) or 2D (supervisor). This short address space is commonly referred to as Short I/O space. (A 512-byte block of this address space resides in the V/ESDI 4201 on-board RAM.)

The Short I/O is the window through which the host can send commands and parameters, and can review controller status and error codes. The reason that it is called Short I/O space is because any time a board sees a transaction with an address modifier of 29 or 2D in the address strobe, it only looks at the first (lower) 16 address lines even though the VMEbus supports the full 32 address lines. The upper 16 address lines in Short I/O are ignored.

By looking only at the lower 16 address lines, less hardware is needed and less time is spent decoding lines that are not necessary for functions such as reading command registers and status registers. The obvious result is improved performance and reduced cost.

The base address of this block of memory is set by the on-board DIP switches (see section 5).

The V/ESDI 4201 provides Short I/O space on-board as an added feature for improved performance. The 512 bytes of Short I/O provided can accommodate several commands at once. This proves advantageous to system performance by eliminating the need to DMA every command across the bus. So not only is system performance improved, but bus bandwidth is also conserved.

NOTE

Short I/O space limitations dictate that all accesses be either 8 or 16 bits. Therefore, if the CPU card generates a 32-bit data path and you are using the C programming language, then it is necessary to cast the 32-bit IOPB fields (i.e., buffer address and bytes/sector) as two 16-bit data types (unsigned shorts on MC68020). This is because the VMEbus short I/O space is 16 bits and will not respond properly to high-order data strobes.

Arbitration

At some point, either on its own or when it receives a command, every bus master device needs to get on the VMEbus and must request access. This process of requesting and granting the bus is referred to as bus arbitration.

The VMEbus provides four levels of bus request (0-3) to provide prioritized requests. In addition, bus priority is provided by position of the board in the card cage. The closer the board is to slot one (system controller), the higher priority it is given.

The V/ESDI 4201 bus request priority is selectable via on-board jumpers. The factory setting is bus priority 3.

This system of prioritizing means that if two devices with the same priority simultaneously request the bus, the one closest to the system controller is granted bus access first. However, before any bus grant can be issued, the bus must first be cleared (i.e., the current bus master must release the Bus Busy signal).

A VMEbus system may utilize one of three different schemes of bus arbitration based on the two methods of prioritizing bus requests. The first method is called single level arbitration. In this scheme, every device on the bus is assigned the same priority level (e.g., level 3), and the bus requests are prioritized by slot only. This scheme is most useful in low-cost systems where the system controller is on the CPU board. In addition, it takes fewer gates to implement and is less expensive than the other two methods of arbitration. However, the remaining two options offer higher performance than single level arbitration.

The second type is called priority arbitration. Priority arbitration utilizes a signal called Bus Clear. When the system controller sees a request from a higher priority than the one already on the bus, the system controller drives the Bus Clear signal which is a recommendation to the bus master to get off of the bus. When the V/ESDI 4201 receives a bus clear, it will get off of the bus within two bus cycles. This provides better system performance by allowing the V/ESDI 4201 to be run at a low priority level and still not worry about it "hogging" the bus. Finally, the third type of arbitration is called round robin. With this type, there is no fixed priority based on level (i.e., three is not always the highest priority). Instead, priority is determined in a cyclical fashion. In other words, if a priority three is on the bus, the next bus grant will be given to priority two (even if another priority three is present), the next will be given to a priority one and the next to a zero and then back to three. It is useful in a system that is using many similar devices. For example, if four disk controllers are on the backplane and all need equal access to the bus, the need for "position" priority is eliminated. In this scheme, each board will have equal access regardless of board position.

The V/ESDI 4201 supports all three types of arbitration.

Priority Interrupts

Interrupts are a convenient way for a bus master to tell the CPU that it needs to communicate with another device across the bus. The VMEbus supports seven levels of interrupts with seven being the highest and one being the lowest. The interrupt operation is very similar to the bus grant operation, except the signals involved are the interrupt request from the board and the Interrupt Acknowledge

(IACK) signal from the system controller. Each V/ESDI 4201 command can be programmed with any one of the seven interrupt levels.

As in arbitration, interrupts are also prioritized by slot position. Therefore, if multiple boards request interrupts, the system controller drives the IACK signal and it is passed along serially by each board in the system until the interrupting board reads it.

Interrupt Operation

After reading a file, a disk controller may want to tell the CPU that it (the CPU) may now use that data. The disk controller gives up the bus and generates the interrupt request on the specified interrupt level. At some point, when the bus is clear, the system controller passes down the IACK signal. When the board that generated the interrupt reads the signal, it responds by issuing an interrupt vector (if enabled). An interrupt vector describes the interrupt to the system controller. When the interrupt vector is valid on the bus, the CPU is in master mode, and the interrupting device will respond with a data acknowledge signal. This tells the system controller it has the vector for the CPU. The V/ESDI 4201 supports unique error codes and vectors for several different types of operation. In addition, a different vector can be used to indicate normal completion or completion with error. This can significantly reduce system overhead by instantly identifying the status of each command as it is completed. The V/ESDI 4201 has a separate status change interrupt and vector to facilitate interrupt processing during overlapped seek operations. Because a unique vector can be used, the system can immediately identify the source of the interrupt with a minimum of processing.

By using unique vectors, the system driver does not need to access registers to take care of post-interrupt activity. Instead, the vector can tell the host exactly what happened, so the host can proceed without any futher delay.

The following table summarizes specific functionality for each of the V/ESDI 4201 boards. For details, please contact Interphase Customer Service.

Table 1-1. Product Variations

TABLE OF PRODUCT VARIATIONS

Dash #	Description
- 0	No longer in production. Standard product as per 4201 User's Guide.*
- 1	No longer in production. "B" Row only on P2 connector.* Added 175ns delay line on the Bus Packet for slower systems.
- 3	No longer in production. Customer specific. Berg Disk I/O connectors.
- 4	No longer in production. Customer specific. Specific firmware required.
- 5	Added 175ns delay line on the Bus Packet for systems requiring slower VME accesses. "B" Row only on P2 connector. For Sun 3E and ISI users.
- 6	Full P2 connector.
- 7	"B" Row only on P2 connector.
4201 - DC/SE	SCSI Port daughter card. Single-ended driver. Cannot be converted to a differential.

* (Requires XOL or greater firmware)

2 INSTALLATION OF THE V/ESDI 4201

Cabling Options

The ESDI drive cables connect to the V/ESDI 4201 through vertically mounted headers (J1-J5) on the V/ESDI 4201 card.

J1 is a 34-pin connector for the ESDI "A" cable. J2-J5 are 20-pin connectors for "B" cable connection of up to four ESDI drives. J2 is the header for drive zero and J3 is the header for drive one, J4 is the header for drive two, and J5 is the header for drive three. Connector pinouts for both the "A" cable and "B" cables are shown in Appendix B of the user's guide.

If connectors J2-J5 are used to connect the ESDI drives, the cables may be routed through the front of the VMEbus card cage.

For information concerning the installation of the optional V/ESDI 4201 SCSI port, refer to the supplemental user's guide: UG-0780-000-XXX. Details of SCSI connectors, installation procedures and other installation considerations are provided.

Figure 2-1 on the following page shows the position of the cable connections, the jumpers, and the location of the option switches on the V/ESDI 4201 printed circuit card. Please refer to the diagram for information as indicated in the remainder of this section of the user's guide.

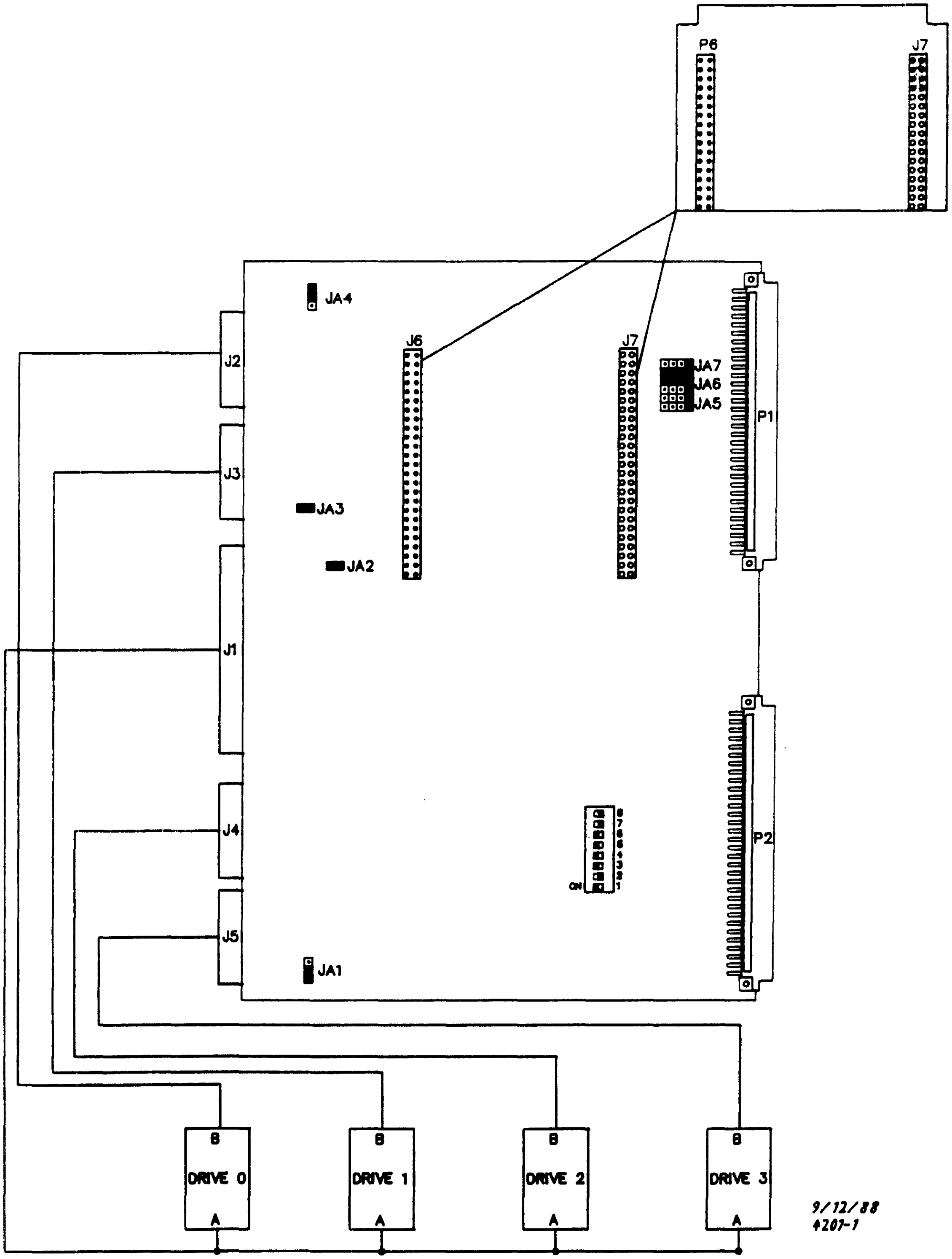
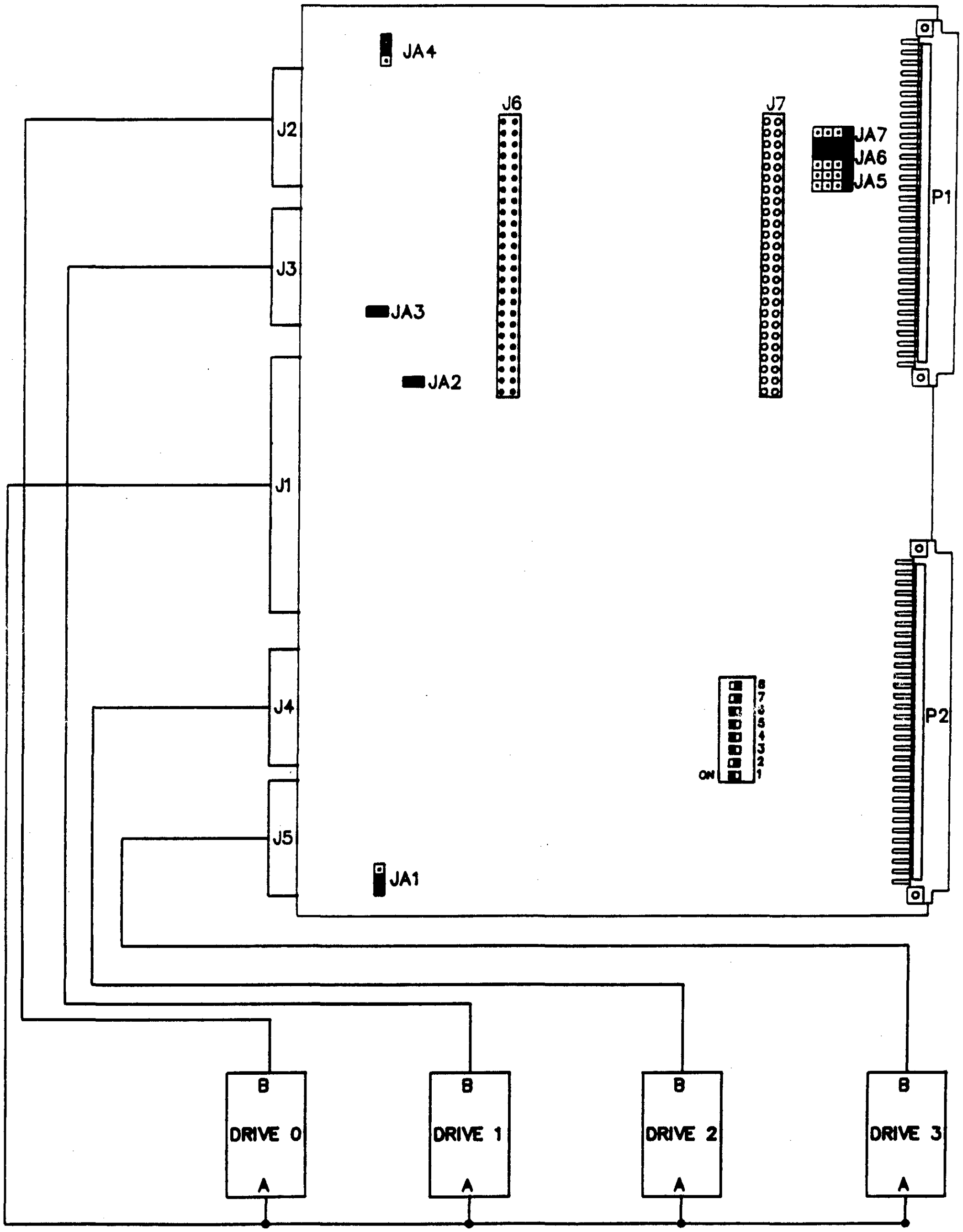


Figure 2-1. Board Layout



9/12/88
4201-1

Figure 2-2. Cable Installation

Table 2-1. P2 Connector Scheme

VMEbus P2 Connector Scheme

Pin	Row A	Row B	Row C
1	GND	+5V	HS3*
2	DIFFEN 0	GND	HS2*
3	DIFFEN 1	RESERVED	HS0*
4	DIFFEN 2	A24	HS1*
5	DIFFEN 3	A25	XREQ*
6	DIFFEN 4	A26	ESD1*
7	DIFFEN 5	A27	ESD2*
8	DIFFEN 6	A28	ESD3*
9	DIFFEN 7	A29	UB2-6
10	DIFFEN 8	A30	WATGATE*
11	DIFFEN 9	A31	ADGATE*
12	DIFFEN 10	GND	GND
13	SCTL 0	+5V	+12V
14	SCTL 1	D16	CMDCPL1
15	SCTL 2	D17	CMDCPL2
16	SCTL 3	D18	CSDATA*
17	SCTL 4	D19	INDEX*
18	SCTL 5	D20	XACK*
19	SCTL 6	D21	SECTOR*
20	SCTL 7	D22	READY*
21	SCTL 8	D23	ATTN*
22	SCSID* 0	GND	UD1-2
23	SCSID* 1	D24	UD1-1
24	SCSID* 2	D25	UD1-6
25	SCSID* 3	D26	UD1-7
26	SCSID* 4	D27	UD1-10
27	SCSID* 5	D28	UD1-9
28	SCSID* 6	D29	UD1-14
29	SCSID* 7	D30	UD1-15
30	SCSID* 8	D31	UM9-2
31	+12V	GND	UB2-12
32	GND	+5V	GND

Option Switch Settings

The V/ESDI 4201 has a switch block (S1) that contains eight switches (numbered 1-8). (see Figure 2-1).

Switches one through seven correspond to VMEbus address lines A9-A15 respectively. An OFF switch has a value of '1' and an ON switch has a value of '0', which are used to select the base address of the controller. Table 2-2 represents all possible base address and switch settings for the controller.

Switch 8 is used to select the address modifiers for the V/ESDI 4201 Short I/O space. If switch 8 is on, only supervisor accesses are permitted (address modifier 2D only). If switch 8 is off, then both 2D and 29 address modifiers are selected.

Table 2-2. Base Address Switch Settings

BASE ADDRESS SWITCH SETTINGS

O = ON / CLOSE
F = OFF / OPEN

ADDRESS	SWITCH SETTING 7 6 5 4 3 2 1	ADDRESS	SWITCH SETTING 7 6 5 4 3 2 1
0000	OOOOOOO	8000	FOOOOOO
0200	OOOOOOF	8200	FOOOOOF
0400	OOOOOFO	8400	FOOOOFO
0600	OOOOOFF	8600	FOOOOFF
0800	OOOOFOO	8800	FOOOFOO
0A00	OOOOFOF	8A00	FOOOFOF
0C00	OOOOF FO	8C00	FOOOFFO
0E00	OOOOF FF	8E00	FOOOFFF
1000	OOOFOOO	9000	FOOFOOO
1200	OOOFOOF	9200	FOOFOOF
1400	OOOFOFO	9400	FOOFOFO
1600	OOOFOFF	9600	FOOFOFF
1800	OOOFFOO	9800	FOOFFOO
1A00	OOOFFOF	9A00	FOOFFOF
1C00	OOOFFFO	9C00	FOOFFFO
1E00	OOOFFFF	9E00	FOOFFFF
2000	OOFOOOO	A000	FOFOOOO
2200	OOFOOOF	A200	FOFOOOF
2400	OOFOOFO	A400	FOFOOFO
2600	OOFOOFF	A600	FOFOOFF
2800	OOFOFOO	A800	FOFOFOO
2A00	OOFOFOF	AA00	FOFOFOF
2C00	OOFOFFO	AC00	FOFOFFO
2E00	OOFOFFF	AE00	FOFOFFF
3000	OOFFOOO	B000	FOFFOOO
3200	OOFFOOF	B200	FOFFOOF
3400	OOFFOFO	B400	FOFFOFO
3600	OOFFOFF	B600	FOFFOFF
3800	OOFFFOO	B800	FOFFFOO
3A00	OOFFFOF	BA00	FOFFFOF
3C00	OOFFFFO	BC00	FOFFFFO
3E00	OOFFFFF	BE00	FOFFFFF
4000	OFOOOOO	C000	FFOOOOO
4200	OFOOOOF	C200	FFOOOOF
4400	OFOOOFO	C400	FFOOOFO
4600	OFOOOFF	C600	FFOOOFF
4800	OFOOFOO	C800	FFOOFOO
4A00	OFOOFOF	CA00	FFOOFOF
4C00	OFOOFFO	CC00	FFOOFFO
4E00	OFOOFFF	CE00	FFOOFFF
5000	OFOFOOO	D000	FFOFOOO
5200	OFOFOOF	D200	FFOFOOF
5400	OFOFOFO	D400	FFOFOFO
5600	OFOFOFF	D600	FFOFOFF
5800	OFOFFOO	D800	FFOFFOO
5A00	OFOFFOF	DA00	FFOFFOF
5C00	OFOFFFO	DC00	FFOFFFO
5E00	OFOFFFF	DE00	FFOFFFF
6000	OFFOOOO	E000	FFFOOOO
6200	OFFOOOF	E200	FFFOOOF
6400	OFFOOFO	E400	FFFOOFO
6600	OFFOOFF	E600	FFFOOFF
6A00	OFFOFOF	EA00	FFFOFOF
6C00	OFFOFFO	EC00	FFFOFFO
6E00	OFFOFFF	EE00	FFFOFFF
7000	OFFFOOO	F000	FFFFOOO
7200	OFFFOOF	F200	FFFFOOF
7400	OFFFOFO	F400	FFFFOFO
7600	OFFFOFF	F600	FFFFOFF
7800	OFFFFOO	F800	FFFFFOO
7A00	OFFFFOF	FA00	FFFFFOF
7C00	OFFFFFO	FC00	FFFFFFO
7E00	OFFFFFF	FE00	FFFFFFF

Jumper Settings

The user can select bus request priority from 0 (lowest) to 3 (highest) using the on-board jumpers JA5, JA6, and JA7. All of the possible jumper configurations for this jumper block are illustrated below. To locate the jumper block on the V/ESDI 4201, refer back to Figure 2-1.

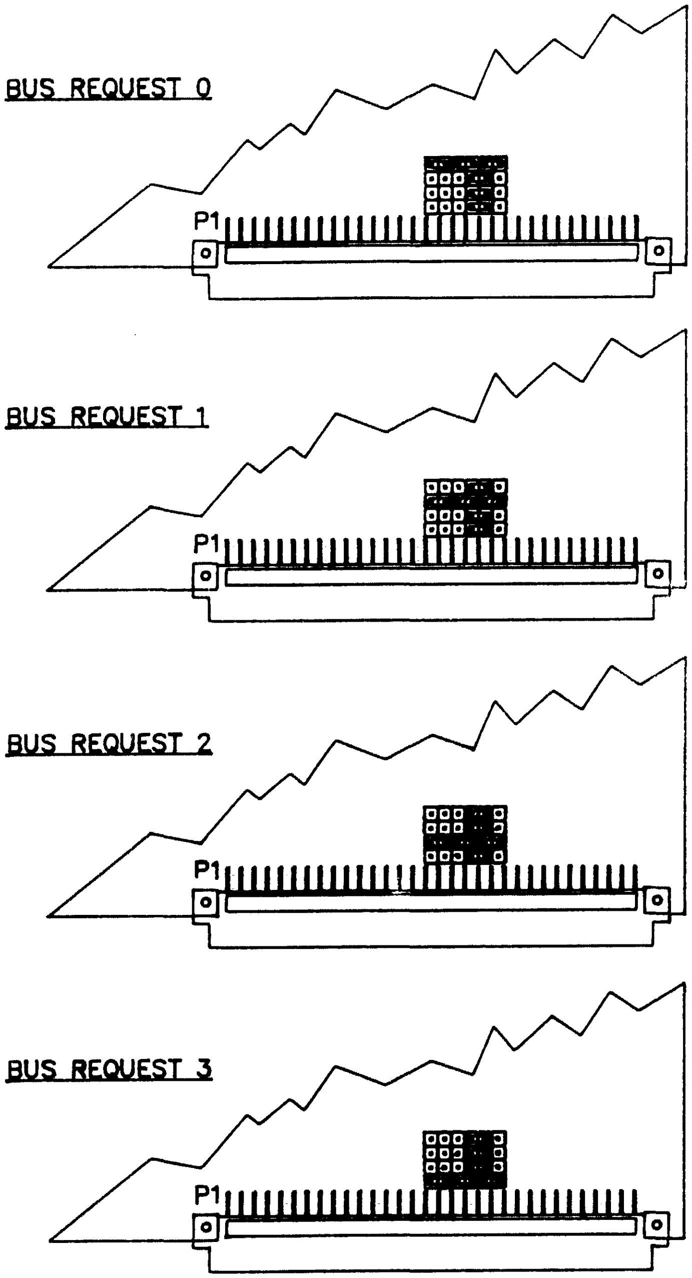


Figure 2-3. Jumper Configurations

In addition to the jumpers for selecting bus priority, there are several other on-board jumpers to configure the V/ESDI 4201 for specific hardware characteristics. Jumper E1 selects EPROM size (either 256K or 512K) and jumper E4 determines V/ESDI 4201 signal output (either to the on-board headers, or to VMEbus P2). For the physical location of these jumpers for this option, refer back to Figure 2-1.

The unmarked jumpers are for factory use only and should not be changed by the user. The board is shipped with the jumpers in the positions shown in Figure 2-1, at the beginning of this section.

Installation

The V/ESDI 4201 is designed to ensure easy installation into the VME system.

Upon receipt of the board, check to make sure that no damage has occurred during shipping. Usually, a thorough visual inspection is sufficient since each board is thoroughly checked at Interphase just prior to shipment.

WARNING

The V/ESDI 4201 is extremely sensitive to electrostatic discharge (ESD), and the board could be damaged if it is handled improperly. Interphase ships the board enclosed in a special anti-static bag. Upon receipt of the board, take the proper measures to eliminate board damage due to ESD (i.e., wear a wrist ground strap or other grounding device).

WARNING

Do NOT install or apply power to a damaged board. Failure to observe this warning could result in extensive damage to the board and/or system.

If the board is undamaged and all parts are accounted for, proceed with the installation.

1. The first step is to set all on-board jumpers so that the V/ESDI 4201 is properly configured for operation within your system. Those options are discussed at the beginning of this section, and should be reviewed before continuing. The board is shipped from the factory set for Bus Request level 3 and with JA4 jumpered 1-2 (for drive cables connected directly to the controller rather than to the optional P2 Adapter Card).

2. Once the board is configured, ensure that both the system power and the disk drive power are OFF.

WARNING

System power and disk drive power must be OFF before the V/ESDI 4201 can be installed. Failure to do so may result in severe damage to the board and/or system board and/or system.

3. When the power is off, connect the "A" cable (see Appendix A for pin-out details) to the disk drive, making sure that the pins are properly oriented. If only one disk drive is used, it must be attached to the last connector on the cable.

4. Then, install terminators on the last drive on the cable. (If only one drive is connected, the terminators must be connected to that drive.)

5. After routing the "A" cable through the VME system to the proper card slot, connect a "B" cable (see Appendix A for pin-out details) to the disk drive. If more than one drive is being used, connect a "B" cable to each one, ensuring that the connectors are properly oriented.

6. Route the "B" cable to the proper VME card slot, and insert the V/ESDI 4201 about one-third of the way into the slot. Carefully connect the cables as follows:

   "A" Cable - J1
   "B" Cable - J2 (Drive 0)
   "B" Cable - J3 (Drive 1)
   "B" Cable - J4 (Drive 2)
   "B" Cable - J5 (Drive 3)

7. Carefully slide the board the rest of the way into the slot. It should slide all the way in without any difficulty. If it doesn't, pull it out and check to make sure that the cables are not in the way.

   If there is not enough clearance for the board, the cable strain relief may be removed and installation continued.

8. Once the board is properly seated in the slot, tighten the captive mounting screws on each end of the board.

9. When the board is installed, run a complete test on the system to ensure system integrity.

3
INITIALIZING THE V/ESDI 4201

Overview

The V/ESDI 4201 can control a wide variety of disk drives with differing formats, sizes, and speeds. To accommodate this diversity, the controller must be told which drive is attached to each of the ports, and which options are to be selected before using the drives. The Initialize command provides this information. Each port must be initialized upon power-up and again after every reset.

The V/ESDI 4201 also provides an Initialize Long command. It is used to load an extended UIB, which is required if multiple sector slipping is going to be used on the drive being initialized. The first nine words of both the standard UIB and the extended UIB are identical. However, the extended UIB has seven additional words appended to the end of a standard UIB.

When an Initialize command is issued, the buffer address in the IOPB points to a list of initialization parameters called a Unit Initialization Block (UIB). There are two different UIBs which may be read depending on whether a normal Initialize command (87 hex) or an Initialize Long command (7C hex) is issued. The standard UIB contains 18 bytes of information. Upon power-up, the V/ESDI 4201 installs default UIB parameters for each port. The default UIB contains the minimum information necessary to read data from a typical ESDI-type drive. This feature allows boot programs to be drive-independent.

Unit Initialization Block

The UIB information for a specific drive can be stored in the first sector of data in that drive. The default UIB allows the host to read this information from the drive and to configure the V/ESDI 4201 to conform to the required drive specifications. The V/ESDI 4201 will not allow any type of Write or Format commands to be executed before an Initialize command has been issued and executed. Note that failure to initialize a drive before a Write or Format operation is attempted will result in an error code of 40 (HEX), unit not initialized. Since the end of the UIB does not fall on a long word boundary, the V/ESDI 4201 must access the UIB in word mode even if the "memory type" IOPB parameter is set to 03 (long word). The memory type will be changed to 02 by the V/ESDI 4201 if this occurs.

The following table shows the format of a standard UIB. This format is also used for the first nine words of the extended UIB. A description of the contents of each byte follows the table. When individual bits within each byte are not specifically noted, it may be assumed that they are "Don't Cares."

Table 3-1. UIB Format

	Format of UIB
Byte #	DESCRIPTION
0	V1SH - VOLUME 0 STARTING HEAD #
1	V1SH - VOLUME 0 NUMBER OF HEADS (2 drives only)
2	V1SH - VOLUME 1 STARTING HEAD # (2 drives only)
3	V1SH - VOLUME 1 NUMBER OF HEADS
4	SCT/TRK - SECTORS PER TRACK
5	SKEW - SPIRAL SKEWING FACTOR
6	BYTES/SCT (MSB) - BYTES PER SECTOR (UPPER BYTE)
7	BYTES/SCT (LSB) - BYTES PER SECTOR (LOWER BYTE)
8	GAP 1 - NUMBER OF WORDS IN GAP 1
9	GAP 2 - NUMBER OF WORDS IN GAP 2
A	SCT INT - SECTOR INTERLEAVE FACTOR
B	RETRY - NUMBER OF RETRIES ON DATA ERROR
C	CYL (MSB) - NUMBER OF CYLINDERS (UPPER BYTE)
D	CYL (LSB) - NUMBER OF CYLINDERS (LOWER BYTE)
E	ATTRIB - ATTRIBUTES
F	ATTRIB - SECOND ATTRIBUTES SET
10	STATUS CHANGE INTERRUPT LEVEL
11	STATUS CHANGE INTERRUPT VECTOR

Volume Specification: UIB Bytes 0-3

Each drive can be separated into two volumes, and each volume can then be treated as an individual drive. This option is particularly useful when dealing with mixed media such as fixed and removable drives. It should be noted, however, that a drive need not have mixed media to use the volume selections. All fixed or removable drives can also be set up as if they contained two volume.

By specifying where the removable and fixed media starts, each medium can be treated as a separate drive. This prevents head increments from crossing medium boundaries. Bytes 0 and 1 specify the starting head and the number of heads for volume 0. Bytes 2 and 3 specify the starting head and the number of heads for volume 1. If volume 1 does not exist, bytes 2 and 3 must be 0.

Sectors/Track: UIB Byte 4

UIB Byte four is the number of sectors with which each track will be formatted. The maximum allowable number of sectors is 160. This number indicates the usable sectors per track, and should not include the spare or runt sector, if either exists.

Spiral Skew: UIB Byte 5

The spiral skew is used by the V/ESDI 4201 to determine the number of sectors that sector zero is offset from the index pulse. Setting the spiral skew factor allows head adjustments on each track without losing a revolution. This byte is used only during formatting and only when the spiral skew in the IOPB is set to zero. For a more detailed explanation of the use of spiral skew with the V/ESDI 4201, please refer to Appendix A.

Bytes Per Sector: UIB Bytes 6 and 7

These two bytes are the number of data bytes per sector. Overhead bytes such as gaps are not included wen the bytes per sector is specified. The number entered in this field must be even (no half words). The minimum sector size is 256 bytes, and the maximum is 2408 bytes.

Gap 1 Words: UIB Byte 8

GAP 1 is all zeros gap before the header information on each sector. Byte 8 represents the number of words of zeros that are written in GAP 1 during a format only. Refer to Appendix A for suggested Gap 2 settings for specific drives.

Gap 2 Words: UIB Byte 9

GAP 2 is the all zeros gap between the header and the data field on each sector. Byte 9 represents the number of words of zeros that is written in GAP 2 during a Format or Write operation. Refer to Appendix A for suggested Gap 2 settngs for specific drives.

Sector Interleave: UIB Byte A

The sector interleave determines how the physical sectors are formatted on disk. With byte A set to '1', the sectors are formatted with the 1:1 interleave. With byte A set to '2', the sectors are formatted with 2:1. Other interleave factors are determined by setting this byte to the desired number (i.e., '3' = 3:1, '4' =4:1, etc.). Interphase suggests that this byte be set to '1' for optimim performance.

Retry Count: UIB Byte B

Byte B is the maximum number of rotational retries attempted when an error occurs in the data field. A maximum number of 15 retries is suggested, since a maximum number of 15 retry attempts can be posted.

Number of Cylinders: UIB Bytes C and D

Bytes D and C represent the number of cylinders that the V/ESDI 4201 knows are on the drive. This number is used to determine if a Seek request is valid. Any Seek or implied Seek request for cylinders greater than the known number will result in an error.

Attributes: UIB Byte E

Attributes such as reseek, move bad data, increment by head, etc. are defined by this byte. Each attribute is designated in the following figure which is then followed by a definition of each.

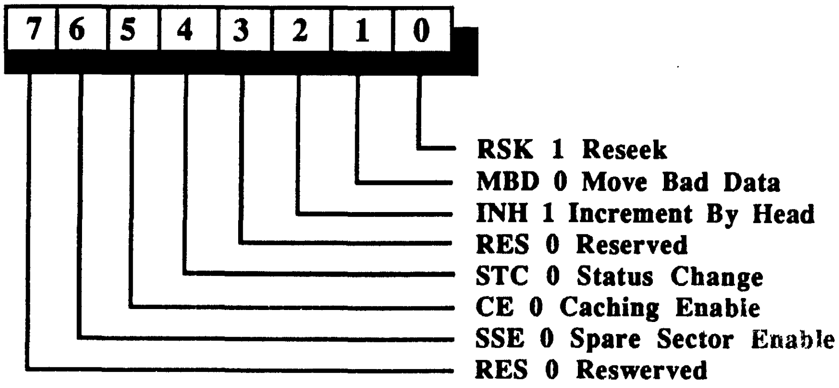


Figure 3-1. UIB Attribute Byte E

Bit 7 Reserved

This bit is reserved and must be set to '0'.

Bit 6 Spare Sector Enable

If this bit is set at '1', it allows the use of a spare sector on each track. The sector is allocated when the track is formatted. Refer to the Map Sector command in Appendix C for a discussion of sector mapping methods.

Bit 5 Caching Enable

If '1', this bit enables sector caching. If '0', data is always transferred from the disk, and no extra data is moved into the on-board buffers. Zero latency operation is not affected.

Bit 4 Status Change

If '1', this bit enables an interupt and updating of the Status Change bit in the CSR (or optional status change register) when any unit's status changes (e.g., ready or not ready, explicit seek complete, etc.).

Bit 3 Reserved

This bit is reserved and must be set to '0'.

Bit 2 Increment by Head

If '1', this bit specifies the increment by head, then cylinder when a track boundary is crossed. If '0', increments are done by cylinder, then head at track boundaries. An increment by head scheme is usually preferred since head switch time is much less than cylinder-to-cylinder seek time.

Bit 1 Move Bad Data

If '1', this bit enables the transfer of possibly bad data into system memory. If the data is still in error after all retries and error correction attempts, and reseeks that are specified have been performed, the V/ESDI 4201 will normally post an error without transferring any data to system memory. if the Move Bad Data bit is set, the V/ESDI 4201 will move the bad data to system memory (set bit six of the error code byte in the IOPB) and return a Command Completed with Exception status in the IOPB.

Bit 0 Reseek

If '1', this bit enables a restore and reseek after the retry count has been exhausted on an erroneous data field. The restore and reseek operation is performed after the retry count has been exhausted and prior to applying ECC (if enabled).

Second Attributes Set: UIB Byte F

This byte is used to select the type of ESDI drive that is being used (i.e., hard sectored, address mark, or embedded servo). In addition, this byte is used to select either 2 or 4 unit operation.

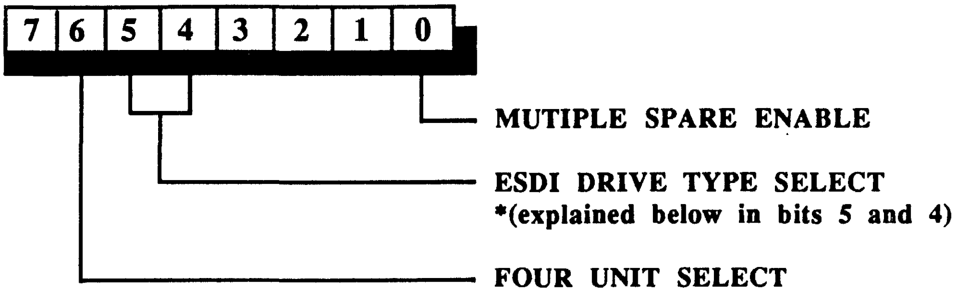


Figure 3-2. UIB Second Attribute Byte F

Bit 6 Four-Unit Select

This bit must be set to '1', if 4 unit operation is required. When this bit is set, the drive number must be selected by setting bits 12 and 13 of the upper byte of IOPB word 8 (or the IOPB interrupt level). If this bit is set, the normal drive number specification bit is no longer used.

For 4 unit operation, the Drive Status Registers reside at the base address plus 1FA (hex) and the base address plus 1FC (hex). The DSR at base address 1FA (hex) supports unit 3 (in the MSB) and unit 2 (in the LSB), and the DSR at base address +1FC (hex) supports unit 1 (in the MSB) and unit 0 (in the LSB). Also, if this bit is set, then bit 7 of UIB byte 10 (selects Optional Status Change Register) must also be set. The Optional Status Change Register resides at base address +1FE (hex).

Bits 5, 4 ESDI Drive Type Select:

These two bits select the type of ESDI drive that is used. The specific types are selected as follows:

00	-	Hard Sector Drive
01	-	Embedded Servo Drive
10	-	Address Mark Drive

When selecting Embedded Servo-type drives (10), the "Sector Count Low" byte in the IOPB (Word 4, LSB) must contain the value calculated for the intersector gap (in bit times divided by 2). The intersector gap time can be found in the drive manual.

Bit 0 - Multiple Spare Enable

This bit must be set to '1' if more than one spare sector will be mapped but may only be set when INIT LONG (command 7C) is used. This bit may not be set if the Spare Sector Enable bit is set in UIB byte E (bit 6). This bit is usually set to '0'.

Status Change Interrupt Level, Register and Vector: UIB Bytes 10 and 11

Bytes 10 and 11 of the UIB set the level and vector for the Status Change Interrupt for the drive being initialized. The level can be set to any number from one to seven. The vector is the data returned during the VME interrupt acknowledge cycle. Any value is acceptable. If bit 7 of UIB byte 10 is '1', this information is returned in the optional status change register at 1FE instead of the CSR. This bit is ignored if bit 4 in the UIB attributes byte is 0.

Extended UIB

The following table shows the format and default settings of the extended UIB which is read when an Initialize Long command (7C) is issued.

Table 3-2. Extended UIB

Extended UIB			
Byte# (hex)	Contents (hex)	Description	Default Setting (DEC)
0	0	VOLUME 0 STARTING HEAD #	0
1	A	VOLUME 0 NUMBER OF HEADS	10
2	0	VOLUME 1 STARTING HEAD #	0
3	0	VOLUME 1 NUMBER OF HEADS	0
4	40	SECTORS PER TRACK	64
5	0	SKEW	0
6	2	BYTES PER SECTOR (MSB)	512
7	0	BYTES PER SECTOR (LSB)	
8	10	NUMBER OF WORDS IN GAP 1	16
9	20	NUMBER OF WORDS IN GAP 2	32
A	1	INTERLEAVE FACTOR	1
B	3	NUMBER OF RETIRES ON DATA ERROR	3
C	2	NUMBER OF CYLINDERS (MSB)	644
D	84	NUMBER OF CYLINDERS (LSB)	
E	5	ATTRIBUTES SET	5
F	0	SECOND ATTRIBUTES SET	0
10	1	STATUS CHANGE INTERRUPT LEVEL	1
11	FF	STATUS CHANGE INTERRRUPT VECTOR	255
12	0	WRITE DELAY	0
13	0	NUMBER OF SPARES	0
14-15	0	RESERVED	0
16	B	COMMAND COMPLETE TIMEOUT (MSB)	3000
17	B8	COMMAND COMPLETE TIMEOUT (LSB)	
18-20	0	RESERVED	0

ATTRIBUTES ENABLED: No Cache, Increment by Head, and Reseek

Byte 14, 15, and Bytes 18 through 20 (hex) of the extended UIB are not used at this time and are all reserved. These bytes may be used in future updates to the V/ESDI 4201.

Write Delay: UIB Byte 12

Number or sectors to skip on a zero latency write before attempting to write. Use full to ensure that sufficient data has been prefetched before WR OP begin to preclude disk underrun.

Number of Spares: UIB Byte 13

Byte 13 of the extended UIB is used to specify the number of sectors that will be mapped as spare sectors. Any number of sectors can be used as spares as long as the number is not greater than the number of sectors per track. Also, byte 4 (Sectors/Track) must be altered to reflect the number of usable sectors. In Byte 4, enter the number of sectors per track minus the number of spares specified.

Command Complete Timeout (MSB and LSB): UIB Byte 16 and 17

The maximum amount of time the controller will wait (in milliseconds) for command complete from the ESDI drive before timing out.

Default UIB

Upon power-up or reset, the V/ESDI 4201 automatically initializes itself to a set of default conditions (default UIB) chosen for a typical ESDI drive. Using the default UIB, the user can then read the customized UIB in from the drive, which then replaces the default UIB. This eliminates the need to actually issue an Initialize command. However, if a write- or format-type command is to be performed as the first operation, the user must issue the Initialize command to the drives.

Table 3-3. Default UIB

Byte# (hex)	Contents (hex)	Description	Default Setting_(DEC)_
0	0	VOLUME 0 STARTING HEAD #	0
1	A	VOLUME 0 NUMBER OF HEADS	10
2	0	VOLUME 1 STARTING HEAD #	0
3	0	VOLUME 1 NUMBER OF HEADS	0
4	40	SECTORS PER TRACK	64
5	0	SKEW	0
6	2	BYTES PER SECTOR (MSB)	512
7	0	BYTES PER SECTOR (LSB)	
8	10	NUMBER OF WORDS IN GAP 1	16
9	20	NUMBER OF WORDS IN GAP 2	32
A	1	INTERLEAVE FACTOR	1
B	3	NUMBER OF RETIRES ON DATA ERROR	3
C	2	NUMBER OF CYLINDERS (MSB)	644
D	84	NUMBER OF CYLINDERS (LSB)	
E	5	ATTRIBUTES SET	5
F	0	4 UNIT SELECT/ESDI DRIVE-TYPE	0
10	1	STATUS CHANGE INTERRUPT LEVEL	1
11	FF	STATUS CHANGE INTERRRUPT VECTOR	255

ATTRIBUTES ENABLED: No Cache, Increment by head, and Reseek

Initialization of Disk Media

Before a disk can be put into service, it must first be formatted. The format provides all header information to be recorded for each sector on a track along with a dummy data field that can later be overwritten. The header information tells the V/ESDI 4201 where the Read/Write head is located so that proper position can be ascertained before data transfers are attempted. The remainder of this section will focus on means to improve disk performance by taking advantage of several V/ESDI 4201 features and functions.

A hard disk drive unit is typically made up of a number of disk platters with two sides (surfaces) each. Each surface contains a number of concentric tracks which are further divided into sectors. The term cylinder refers to the three-dimensional cross section of a given track (radially coincident) on a stack of platters. The term is sometimes used interchangeably with the term track. However, a track is really specified by a combination of a cylinder number and head number.

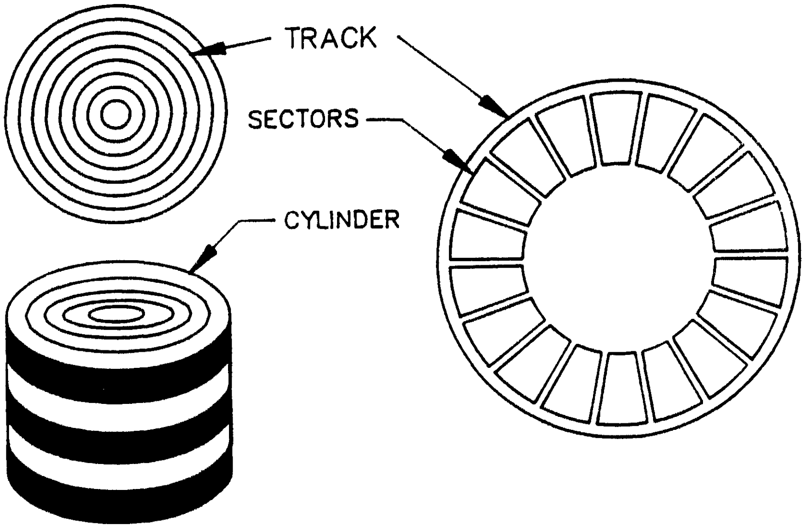
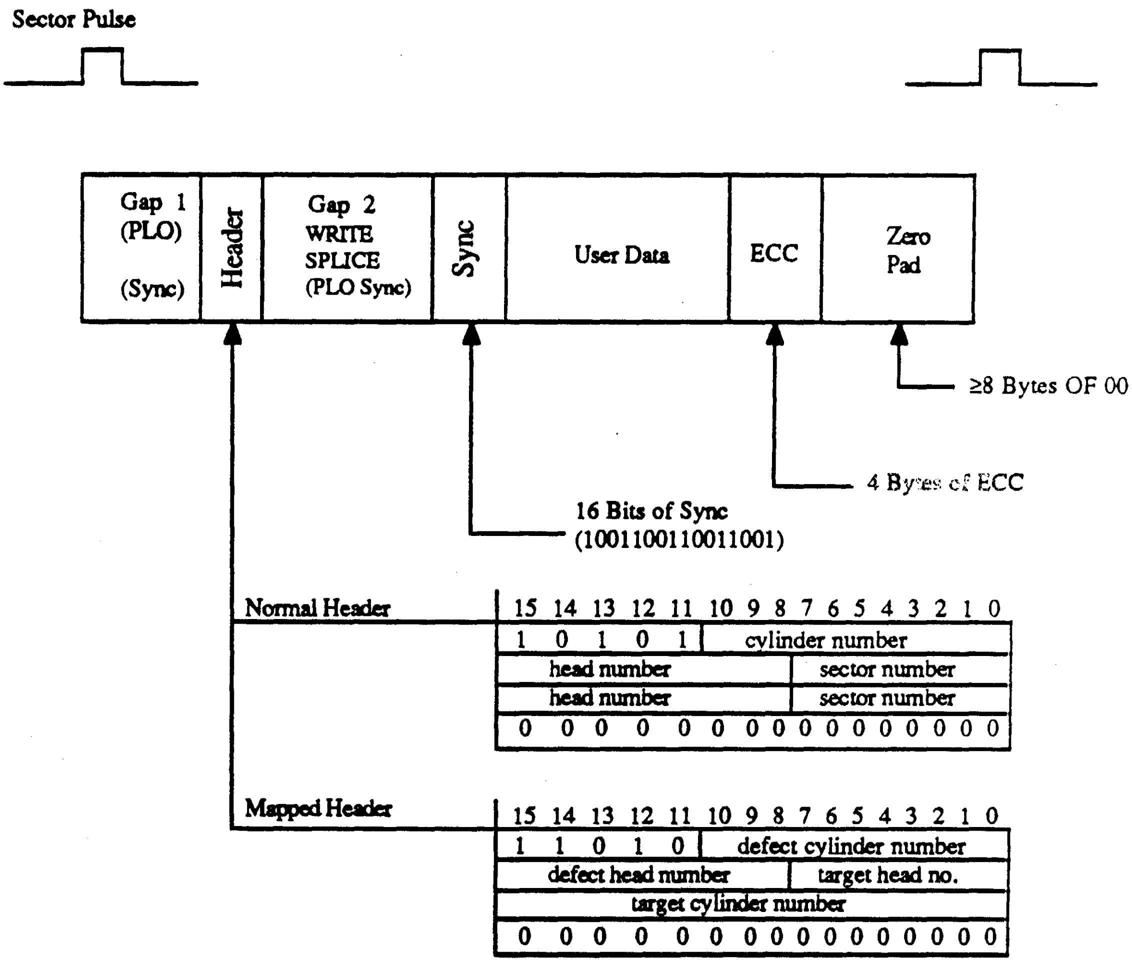


Figure 3-3. Typical Disk Unit

The V/ESDI 4201 automatically formats one track of a disk when it receives a Format command. To format an entire unit, the user must write an iterative program loop with each iteration specifying a different cylinder/head (in the IOPB) until all tracks have been formatted. If the default UIB is used, then the initialization process must precede the format operation since some formatting parameters are defined by the UIB.

In addition, the optimum number of sectors per track should be set before the disk drive is formatted. This parameter is usually specified on ESDI devices by setting switches on the drive (refer to the drive manual for specific details). The number of sectors per track determines the number of bytes in each sector. This is important because, in addition to the data field, each sector must be large enough to accommodate any overhead required by the disk drive and the controller to operate properly. (Drive overhead includes the PLO synchronization time and the length of the write splice. Controller overhead includes the header field, the ECC field, and the space between the sector pulse, header field and data field.)



Normal Header

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	1	cylinder number										
head number								sector number							
head number								sector number							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Mapped Header

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	0	defect cylinder number										
defect head number								target head no.							
target cylinder number															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-4. Sample Formatted Sector

Interphase has compiled a list of the proper parameters required by many popular disk drives. Refer to Appendix A for a current list (as of this printing) of parameters. If the drive that you intend to use is not listed, please call the Interphase Applications Engineering Department for assistance.

The data field can vary from 128 to 2048 bytes but must be an even number. The data placed in the data field during a format command can be either a constant value (specified by byte 15 of the IOPB) or a data field pointed to by the buffer address bytes (IOPB bytes 13, 14, 15). This variable data field is used when the "Format with Sector Data" command is specified instead of the "Format" command.

Formatting Considerations

The format procedure, although simple to implement, must allow for such variables as system and disk latencies, bottlenecks, faulty media, reformatting faulty media, and media verification.

Spiral Skew

The spiral skew controls the number of sectors that sector 0 is offset from the index pulse. This factor is used to reduce latencies during disk transactions. The skew factor can be set in one of two locations (IOPB Word 13 or UIB Word 5).

If Word 13 in the IOPB is 0, then the data in UIB Word 5 is used to calculate the skew. In some systems the normal spiral skew algorithm is not effective because the data is placed on the drive in an unusual way. In this case, the user can specify the skew on a track-for-track basis by setting the skew in Word 13 of the IOPB.

A nonzero value in IOPB Word 13 encountered during a Format Track operation has two effects. It causes the skew in the UIB to be ignored; and it indicates the number of sectors that sector 5 is to be skewed from the index pulse.

How To Determine Skew

If the UIB data is used to indicate skew, the V/ESDI 4201 uses the following algorithm to determine the skew on a track. Multiply the desired skew factor with the number of heads. Take that number and divide it by the number of sectors/track. If the resultant number is not a whole number, the remainder represents the distance that the first sector will be offset from the index pulse.

Example: Skew = 5 Head = 9 Sectors/Track = 16

1. 5 * 9 =45
2. 45/16 =2, remainder 13
3. N =13

Interleave

All of the sectors on a track of data are numbered sequentially from 0 to N minus 1 (N = sectors/track), but are not necessarily physically contiguous. The interleave factor controls the physical separation of these logically sequential sectors. For example, on a disk with an interleave factor of 2:1, logical sectors 1 and 2 are actually one physical sector apart. Interleave factors other than 1:1 are only used when the V/ESDI 4201 is operating without caching. In most cases, caching and zero latency reads/writes provide better system performance than interleaving. If you wish to use interleave factors other than 1:1, please call the Interphase Applications Engineering Department so that we can help you optimize your system.

Physical And Logical Translation

Most operating systems store data in logical addresses, and in order to access or move the data, the host CPU must translate the logical addresses to physical addresses. The V/ESDI 4201 offers a logical translation option which relieves the host of this activity. In many cases, allowing the V/ESDI 4201 to perform the logical-to-physical translation results in improved overall system performance. This command option is selected with bit 4 of the lower byte of IOPB Word 0.

Disk Surface Analysis

Many disk manufacturers will send a media flaw map with the drive. However, there will usually be a difference between the manufacturer's media flaw map and the results of the disk user's surface analysis. This is because some disk manufacturers conduct their testing under environmental extremes and measure internal analog signals to detect bad media. Therefore, some users with very controlled disk environments may choose to ignore the manufacturer's flaw map and only map (or deallocate) areas that his/her surface analysis detects. Interphase recommends recording all of the manufacturer's flaws in addition to any other flaws that may be indicated by the surface analysis. This is because a flaw not indicated by the initial surface analysis may show up after a period of time.

The V/ESDI 4201 has some useful features that facilitate surface analysis. For example, the "Format Track With Data" command allows worst case data patterns to be placed while formatting, and the "Verify Track" command provides a quick way to check data integrity.

In order to detect media flaws that occur over a period of time, some users keep disk statistics. This activity is simplified by the ability of the V/ESDI 4201 to report a command completed with exception status (code 83). If this status is returned in the IOPB, the host can examine the error byte to determine what type of error recovery was required to obtain good data. The host can then log the error condition and the disk location, thus maintaining accurate statistics on the disk.

Faulty Media Mapping

It is not uncommon for a hard disk drive to have imperfect media on the disk surface. It is assumed by the manufacturer that these bad areas will be 'mapped out." Even if perfect media is shipped, bad areas may develop over time.

Bad Track Mapping

Most operating systems allow for this condition through some sort of deallocation scheme. These schemes usually deallocate large blocks of disk space. This method, however, proves to be wasteful. A more efficient method of dealing with this problem is through Bad Track Mapping. When a bad area is detected, the track on which it occurs can be "mapped" to an alternate track on the disk.

When the mapped bad track is encountered during a Read or Write operation, the V/ESDI 4201 automatically seeks to the alternate track. This scheme is efficient in terms of disk space, but a slight speed penalty is incurred due to the extra seek that is required.

Bad Sector Mapping

When speed is more important than disk space, or if a very high number of disk flaws is anticipated, it may be advantageous to use "bad sector mapping." This method of bad area mapping requires that room for a spare sector be left on each track when the disk is originally formatted.

When a bad sector is mapped, the track is reformatted such that the bad sector is moved to the next consecutive sector, which is in turn moved to the next consecutive sector, and so on until the last sector is moved into the spare sector space. This is called "sector slipping."

The V/ESDI 4201 marks the sector containing the bad area as unusable. When the marked sector is encountered during normal Reads and Writes, the V/ESDI 4201 knows to go on to the next sector instead. The V/ESDI 4201 supports both bad track mapping and bad sector mapping.

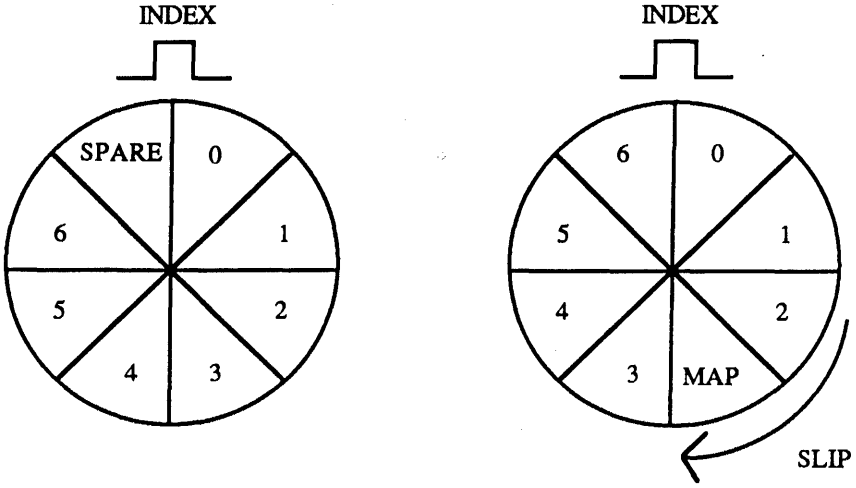


Figure 3-5. Sector Slip

WARNING

No data is saved when slipping sectors. If valid data is on a track that requires sector slipping, it is up to the user to save the data before the slip and to restore the data after the slip.

Multiple Sector Slipping

There are four V/ESDI 4201 commands that are used to implement the multiple sector slipping procedure: 7A, Report Sector ID; 7B, Format with Sector ID; 7C, Initialize Long; 7D, Report Configuration Long. In addition, UIB Byte 13 (Number of Spares) and bit 0 of UIB Byte F (Second Attributes) must be set. Refer to Appendix C for full details of these commands.

There are also two related Error Codes: C0, Both Bits Set; C1, MSE Not INIT Long. Appendix D contains detailed error code information.

NOTE

All data must be removed before attempting to reformat. No data will be saved by this procedure.

Implementing Multiple Sector Slipping. The following procedures should be followed when implementing multiple sector slipping.

1. Set UIB Byte 13 for the number of spares to use.
2. Set UIB Byte 4 for the number of sectors/track minus the value in Byte 13.
3. Initialize the unit using the Initialize Long command (7C).
4. Read and save any data that may be on the track.
5. Issue the Report Sector ID command (7A).
6. Rearrange the table as desired. For example:

Original Table		New Table
		Bad ↓
0 1 2 3		0 1 2 E
4 5 6 7	Assume Sector	3 4 5 6
8 9 A B	3 is bad	7 8 9 A
C D E F		B C D F
		↑
Spares		Spare

7. Issue Format With Sector ID command (7B).
8. Restore data.

Initializing the V/ESDI 4201 Controller

The data field of each sector may be expected by users to be a specific value (i.e. 512 bytes), however, the actual size of each sector must be larger than just the data field. The controller and drive require additional fields for sector identification, PLO sync times, and error detection, and correction information. By using the format specified previously in this chapter, the formula for calculating the drive's minimium bytes/sector is as follows:

Gapl + header + Gap2 + sync + user data + ECC + pad + W-R delay

Gapl, Gap2, and the W-R delay are drive specific parameters; the header, sync, ECC, and pad are controller overhead; and the user data is selected by the user. Refer to the drive manual of the user's system when selecting drive parameters and refer to Figure 3-4, Sample Formatted Sector for the controller overhead (fixed at 22 bytes).

After establishing the fixed controller fields, the formula can be reduced to the following:

Gapl + Gap2 + user data + W-R delay + 22

Selecting the Drive Sector Sizes of the V/ESDI 4201

The following procedure should be used when selecting the drive sector size and building the UIB parameters:

Determine Minimum Bytes/Sector

Determine the minimum bytes/sector required by the drive and controller by setting the variables to the following equation:

Gapl + header + Gap2 + sync + user data + ECC + pad + W-R delay

Gap 1:

This is set by drive parameters and should include the drive's head scatter time (not all drives require this) and the PLO sync time. Place the sum of the values in the UIB Gap 1 byte. Note that the UIB expects a word count and any round odd byte count up to the next even value.

Header:

This field is fixed by the controller and requires 8 bytes.

Gap 2:

This is set by drive parameters and includes the post-header (id field) PLO sync time. Add at least 4 bytes to allow for the controller write splice field. Use this value in the UIB Gap 2 byte. Note that the UIB expects a word count and round odd byte counts up to the next even value.

Sync:

This field is fixed by the controller and requires 2 bytes.

Data:

This field is selected by the user and should be set in the UIB bytes/sector. Note this must be an even byte count.

ECC:

This field is fixed by the controller and requires 4 bytes.

Pad:

This field is fixed by the controller and requires 8 bytes.

W-R:

Some drives require a write gate to read gate transient time. This value is not needed in the UIB but must be allowed for when calculating the drive's minimum bytes/sector. If this value is not required by the drive, still allow for 2 bytes for rotational deviation.

Determine Sectors/Track

After selecting the minimum bytes per sector, determining the sectors per track can be done in several methods. This really depends on the information provided in the drive user's manual. The following reflect the most commonly supplied methods:

Provided Bytes/Sector and Sectors/Track

If the drive manual provides a chart showing bytes/sector and corresponding sectors per track, use that chart. Find the next largest bytes per sector specified in the chart which will accommodate the minimum bytes per sector calculated above. Find the corresponding sectors per track. Set this value in the UIB sectors/track and set the drive switches accordingly.

Note that the bytes/sector selected may not go evenly into the bytes/track. If this is the case, determine how your specific drive handles the remaining bytes.

- If the drive adds the excess bytes to the last sector, i.e., the last sector is larger than the others, no further action is required.

- If the drive puts the excess bytes in a short sector, i.e., the last sector is smaller than the others, determine how "short" is the last sector.
- If the last sector is large enough to accommodate the minimum bytes/sector calculated above, no further action is required.
- If the last sector is too small for the minimum bytes/sector required, but is greater than 120 microseconds, set the UIB sectors/track to the actual -1 and set the UIB attribute flag "runt enable" to a 1.
- If the last sector is smaller than 120 microseconds, try the next smaller sectors/track and repeat the first two steps.

Calculate Sectors/Track

If the drive manual provides an equation to calculate sectors/track, follow the instructions provided and set the drive switches and UIB sectors/track accordingly. Be sure to honor the above "runt sector" note.

The basic formula for determining sectors/track is as follows (be sure to honor the above "runt sector" note):

(sector/track) = (bytes/track) / (bytes/sector)

Optional Drive Set-Ups

Here are s6me additional notes for drive set-up. These are not mandatory requirements, but should be taken into consideration if absolute drive capacity is not of the utmost importance.

- There is no harm in making the physical sector size much larger than the minimum bytes/sector required. Sometimes it is more convenient to select the next even sector size (i.e., bytes/track = 40960, 64 sectors/track =640 bytes/sector, minimum required is 610). By selecting this sector size, no runts are generated.
- It is also recommended that the Gapl and Gap2 values be increased by a few words over drive recommended values. This increases reliability and allows for deviations incurred by mass production.
- It is not necessary nor recommended to account for every byte in each sector

Example: Hitachi DK815-10 Winchester Disk Drive

Step 1:

Determine minimum bytes/sector by setting the variables in the equation:

Gapl + Gap2 + data + W-R delay + 22

Drive Manual Spec		Controller Overhead	
head scatter	16	header	8
PLO sync	11	sync	2
post-id PLO sync	16	ECC	4
W-R transient	25	pad	8
bytes/track	4090		
		total	22

Using the Reduced Formula		
UIB Gap 1	(head scatter + PLO sync)	28
UIB Gap 2	(post header PLO sync + 4)	20
UIB bytes/sector	(specified by user)	512
W-R transient	(rounded up to even byte)	26
Controller Overhead		22
Minimum bytes/sector		608

Step 2:

Determine sectors/track using the equation:

(bytes/track) / (bytes/sector) = sectors/track
960/608 = 67 + 224/608

Because this drive can be configured to add the excess bytes to the last sector (enable "last sector longer"), this yields 66 sectors at 608 bytes and 1 sector at 832 bytes.

Step 3:

Now lets optimize the above configuration and look at additional options. Since we now know that 67 is the maximum number of sectors we can select, let's build a chart of the possible sectors/track and bytes/sector. The sectors can be more evenly divided by using the equation:

(bytes/track) / (sectors/track) = bytes/sector

Remember, this drive adds excess bytes to the last sector. Also, by selecting a sector size greater than the minimum required, we can increase the size of the gaps accordingly (it is not recommended to account for all the bytes in a sector, but increasing each gap by a word or two, if possible, will increase reliability and allow for deviations).

sectors/	bytes/	last	total gaps
track	sector	sector	increase
67	611	634	2
66	620	660	12
65	630	640	22
64	640	640	32

Note that if this drive generated a "runt sector", only the 64 sector option can be used, because the runt is too short (at 67 = 23 bytes, 66 =40, 65 =10 [byte time = 408ns]).

THIS PAGE LEFT BLANK INTENTIONALLY

4 NON-MACSI OPERATION OF THE V/ESDI 4201

Overview

All command and status information, which allows the V/ESDI 4201 to perform disk functions, reside within the 512 bytes of Short I/O space. All disk functions have an extended list of parameters which defines the exact function for the V/ESDI 4201. This list is called the Input/Output Parameter Block (IOPB).

In addition to the IOPB, the V/ESDI 4201 has two 16-bit Read/Write registers for communicating commands and statuses between the V/ESDI 4201 and the host: the Drive Status Register and the Command/Status Register. The Drive Status Register contains two eight bit packets of data which are used to indicate the status of a given drive. The Command/Status Register contains all of the bits used by the host CPU to begin a transaction. It also contains the bits required to indicate the condition of the V/ESDI 4201 to the host CPU.

If the optional SCSI port is used, there is also a SCSI Command/Status Register, a Tape Status Change Register, and seven Tape Drive Status Registers. The format of the Short I/O memory is shown in the following two figures.

The first figure, Figure 4-1, illustrates the format of Short I/O memory space for a V/ESDI 4201 without the addition of a SCSI port. The next figure, Figure 4-2, illustrates how the Short I/O memory space is modified to accommodate the additional registers. To ensure clarity and to avoid possible misunderstandings, the information regarding the SCSI port registers has not been included in this user's guide but is available in a supplemental user's guide (document number UG-0780-000-XXX). Please refer to that document for specific operating parameters of the V/ESDI 4201 SCSI Port.

Further discussions in this user's guide do not include SCSI port considerations unless there is a direct effect on disk operation.

HEX OFFSET	HIGH / BYTE	LOW / BYTE
000	DISK DRIVE 1 STATUS (2 UNIT)	DISK DRIVE 0 STATUS (2 UNIT)
002	COMMAND STATUS REGISTER	
004	COMMAND CODE	COMMAND OPTIONS
006	STATUS CODE	ERROR CODE
008	CYLINDER HIGH (LOG D31-24)	CYLINDER LOW (LOG D23-16)
00A	HEAD (LOG D15-8)	SECTOR (LOG D07-00)
00C	SECTOR COUNT HIGH	SECTOR COUNT LOW
00E	BUFFER ADDR (A31-24)	BUFFER ADDR (A23-16)
010	BUFFER ADDR (A15-08)	BUFFER ADDR (A07-00)
012	BUFFER MEMORY TYPE	BUFFER ADDRESS MODIFIER
014	4 UNIT DRIVE / INT LEVEL	NORMAL INT VECTOR
016	DMA BURST COUNT	ERROR INT VECTOR
018	IOPB PTR (A31-24)	IOPB PTR (A23-16)
01A	IOPB PTR (A15-08)	IOPB PTR (A07-00)
01C	IOPB MEMORY TYPE	IOPB ADDRESS MODIFIER
01E	ABSOLUTE SKEW	ENTRY COUNT
020 • • 1F9	USER AVAILABLE MEMORY (1DA Bytes Available)	
1FA	DISK DRIVE 3 STATUS (4 UNIT)	DISK DRIVE 2 STATUS (4 UNIT)
1FC	DISK DRIVE 1 STATUS (4 UNIT)	DISK DRIVE 0 STATUS (4 UNIT)
1FE	OPTIONAL DISK STATUS CHANGE REGISTER	

Figure 4-1. V/ESDI 4201 Standard Short I/O Memory Format

HEX OFFSET	HIGH / BYTE	LOW / BYTE	
000	DISK DRIVE 1 STATUS (2 UNIT)	DISK DRIVE 0 STATUS (2 UNIT)	
002	DISK COMMAND STATUS REGISTER		
004	COMMAND CODE	COMMAND OPTIONS	DISK IOPB
006	STATUS CODE	ERROR CODE	
008	CYLINDER HIGH (LOG D31-24)	CYLINDER LOW (LOG D23-16)	
00A	HEAD (LOG D15-8)	SECTOR (LOG D07-00)	
00C	SECTOR COUNT HIGH	SECTOR COUNT LOW	
00E	BUFFER ADDR (A31-24)	BUFFER ADDR (A23-16)	
010	BUFFER ADDR (A15-08)	BUFFER ADDR (A07-00)	
012	BUFFER MEMORY TYPE	BUFFER ADDRESS MODIFIER	
014	4 UNIT DRIVE / INT LEVEL	NORMAL INT VECTOR	
016	DMA BURST COUNT	ERROR INT VECTOR	
018	IOPB PTR (A31-24)	IOPB PTR (A23-16)	
01A	IOPB PTR (A15-08)	IOPB PTR (A07-00)	
01C	IOPB MEMORY TYPE	IOPB ADDRESS MODIFIER	
01E	ABSOLUTE SKEW	ENTRY COUNT	
020 • •	USER AVAILABLE MEMORY (0E0 Bytes Available)		
100	TAPE COMMAND STATUS REGISTER		
102	COMMAND CODE	COMMAND OPTIONS	TAPE IOPB
104	STATUS CODE	ERROR CODE	
106	BYTES / RECORD HI WORD	BYTES / RECORD HI WORD	
108	BYTES / RECORD LO WORD	BYTES / RECORD LO WORD	
10A	RECORD COUNT HI	RECORD COUNT LO	
10C	BUFFER ADDR (A31-24)	BUFFER ADDR (A23-16)	
10E	BUFFER ADDR (A15-08)	BUFFER ADDR (A07-00)	
110	BUFFER MEMORY TYPE	BUFFER ADDRESS MODIFIER	
112	INTERRUPT LEVEL	DMA BURST COUNT	
114	NORMAL INT VECTOR	ERROR INT VECTOR	
116	FILEMARK COUNT	RESERVED '0'	
118	RESDL BYTES MSB HI WORD	RESDL BYTES LSB HI WORD	
11A	RESDL BYTES MSB LO WORD	RESDL BYTES LSB LO WORD	
11C	RESDL RECORDS MSB	RESDL RECORDS LSB	
11E	RESERVED '0'	RESERVED '0'	
120	USER AVAILABLE MEMORY (0C0 Bytes Available)		
1E0	TAPE DRIVE 0 STATUS REGISTER		
1E2	TAPE DRIVE 1 STATUS REGISTER		
1E4	TAPE DRIVE 2 STATUS REGISTER		
1E6	TAPE DRIVE 3 STATUS REGISTER		
1E8	TAPE DRIVE 4 STATUS REGISTER		
1EA	TAPE DRIVE 5 STATUS REGISTER		
1EC	TAPE DRIVE 6 STATUS REGISTER		
1EE	RESERVED '0'		
1F0	TAPE STATUS CHANGE REGISTER		
1F2	RESERVED '0' (000 Bytes)		
1FA	DISK DRIVE 3 STATUS (4 UNIT)	DISK DRIVE 2 STATUS (4 UNIT)	
1FC	DISK DRIVE 1 STATUS (4 UNIT)	DISK DRIVE 0 STATUS (4 UNIT)	
1FE	OPTIONAL DISK STATUS CHANGE REGISTER		

Figure 4-2. V/ESDI 4201 w/SCSI Port, Short I/O Memory Format

How to Issue a Command

To initiate a disk function, the host must first build an IOPB and then write a GO command to the Command/Status Register (CSR). A Read Modify Write operation (without releasing the bus) should be performed when setting the GO bit. This ensures that the status of the other bits in the CSR can be maintained. This bit should not be used as a semaphore between multiple host CPUs even if a Read Modify Write operation is performed.

The V/ESDI 4201 does necessary seeking, handles error corrections, and performs the direct memory access (DMA) required to complete the command. The V/ESDI 4201 will then automatically update the IOPB, the CSR, and will freeze the Drive Status Register. At this time, the V/ESDI 4201 will also issue a system interrupt (if enabled to do so).

Interrupts

Interrupts are used to signal the host CPU that the current command has been completed or status has changed (if enabled); therefore, the host does not have to continuously monitor the V/ESDI 4201. The interrupt vectors contain numbers previously defined by the host CPU which identify the type of interrupt that has occurred, such as error on last command or command completed normally. The host can then read the updated status and appropriate information in the IOPB. The interrupt signal is taken off the bus when the interrupt acknowledge cycle is performed in accordance with VME specifications.

The Operation Done bit in the Command/Status Register must be cleared by the host to let the V/ESDI 4201 know that the interrupt has been serviced and that it can begin updating the Drive Status Register. Clearing the Operation Done bit should be a Read Modify Write operation. A sample of this transaction is displayed as a flowchart in Figure 4-3. The user may wish to review the flowchart briefly at this point, and refer back to it after reading the subsequent sections of the user's guide.

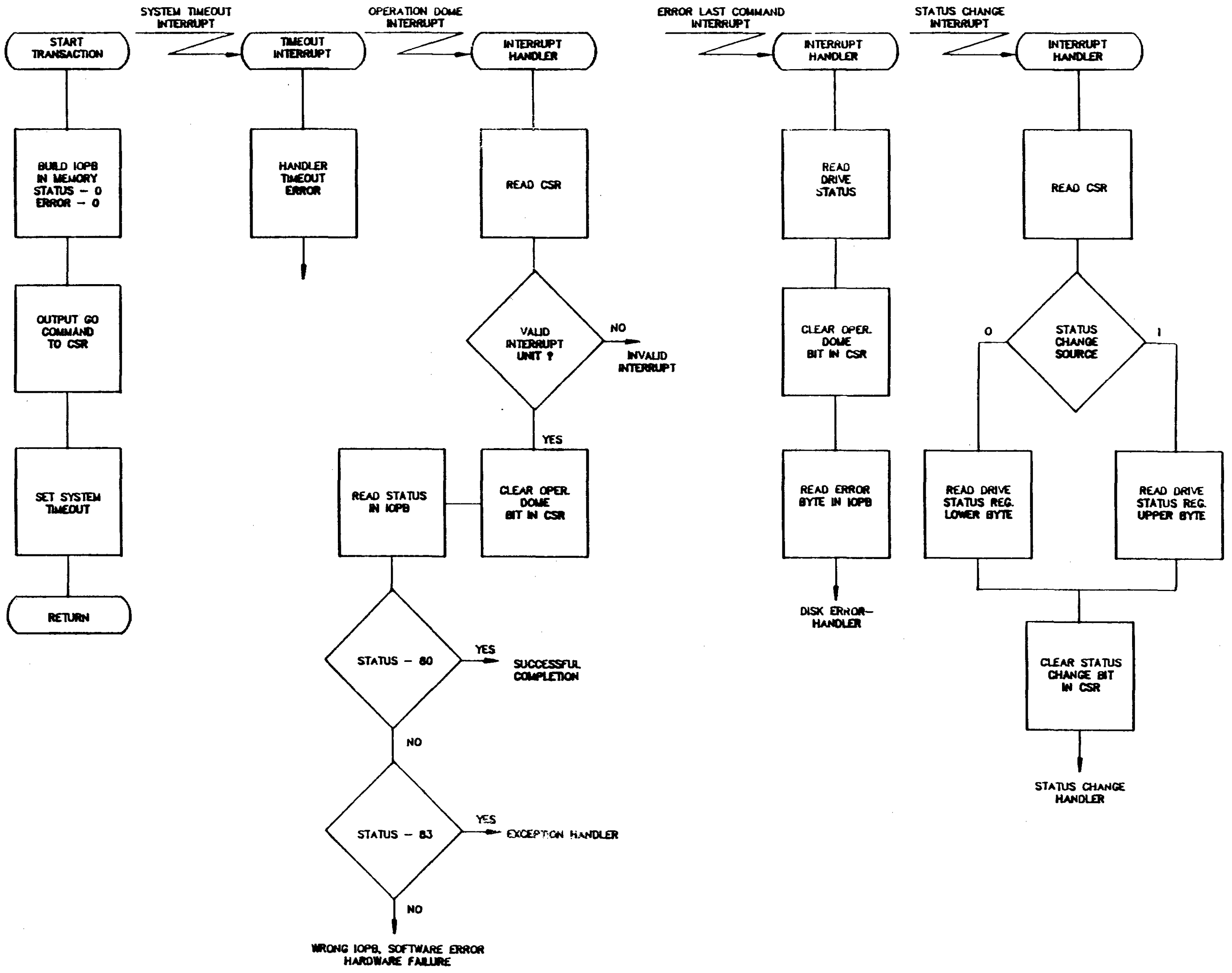


Figure 4-3. Operation Flowchart

IOPB and I/O Registers

The host CPU stores the IOPB and the two I/O registers as a 32-byte block of data in the Short I/O space. The base address of this block of memory is set by the on-board DIP switches (see section 2). The Drive Status Register always resides at the base address. The Command/Status register resides at the base address plus two, and the IOPB resides at the base address plus four. The V/ESDI 4201 will only respond to commands when an IOPB exists at this address. These memory locations can be accessed on either a byte or word basis.

Status Registers

The V/ESDI 4201 uses status registers to communicate information from the V/ESDI 4201 to the host system. These registers are, the Drive Status Registers and the Command/Status Register. The V/ESDI 4201 is powered up in 2 drive mode with the Drive Status Register residing at the base address and the Command/Status Register residing at the base address plus two in V/ESDI 4201 Short I/O memory space. When 4 drive operation is selected, the location of the drive status register changes to the base address plus 1FA (units 3 and 2) and 1FC (units 1 and 0).

Command/Status Register

The Command/Status Register contains all of the control bits used by the host CPU to initiate commands in the V/ESDI 4201. It also contains the bits required to indicate the condition of the V/ESDI 4201 to the host CPU. The CSR is located at the base address plus two in the Short I/O space of the controller.

The format of the Command/Status Register is shown in Figure 4-4. All bits can physically be both written and read by the host, but some of the bits are logically read only or write only. When bit type operations are being performed, a Read Modify Write operation should be used.

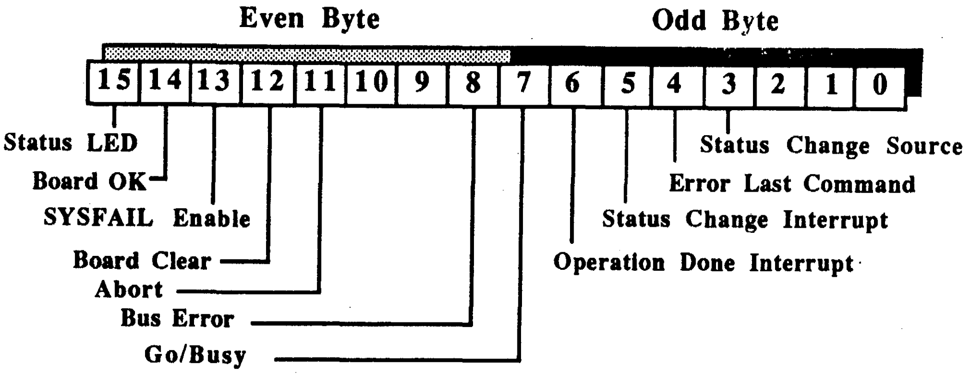


Figure 4-4. Command/Status Register Format

NOTE

The Command/Status Register is accessible by the host through the Short I/O address space. Byte (D8) and word (D16) accesses are both supported, but long word (D32) accesses are not supported. The Status Change and Status Change Source bits can optionally be relocated in the Optional Status Change Register.

A definition of each Command/Status Register bit is provided in the following paragraphs.

Bit 15 Status Led:

If '0', the on-board LED status indicator is red unless the V/ESDI 4201 is accessing the disk, in which case it is green. When '1', the LED is always green provided BOARD OK (bit 14) is '1' (the LED may flicker during a disk access). This bit is controlled by the host and is never changed by the V/ESDI 4201.

Bit 14 Board OK:

If '1', this bit indicates that power-up diagnostics were completed successfully. If '0', this bit indicates that a board-detected fault occurred during the power-up diagnostics. This bit is valid only when the GO/BUSY bit is '0'. This bit is logically a read only bit and should not be changed by the host.

Bit 13 SYSFAIL Enable*:

If '0', and BOK is also '0' (inactive), this bit enables BOK to cause SYSFAIL to be asserted on the VMEbus. If '1', this bit causes the SYSFAIL indication to be disabled. The V/ESDI 4201 never changes this bit.

Bit 12 Board Clear:

If the host sets this bit to '1' and holds it at a '1' for at least one microsecond before setting it to a '0', this causes a hardware reset of the controller. After the reset, the controller will execute its power-up diagnostics (BUSY set). When finished (and if the power-up diagnostics are successful), BUSY is reset and BOK is '1'. If they are not successfully completed, BOK and BUSY are both '0'. There is a 100 microsecond delay before the CSR is valid after a BOARD CLEAR has occurred. This bit is never changed by the V/ESDI 4201. For some of the faster host CPUs, a few NOPs may be necessary between setting and clearing this bit for the V/ESDI 4201 to properly reset.

Bit 11 Abort:

If the host sets this bit to '1', the V/ESDI 4201 will abort any IOPB in progress. This bit is set to '0' by the V/ESDI 4201 upon receipt. If an IOPB was in progress, the Operation Done and Error Last commands will be set to '1' once the abort is completed. If the abort signal is detected during a sector write operation, the V/ESDI 4201 will finish the operation before aborting.

Any disk-oriented command which reads or writes data to the drive may be aborted by setting bit 11 in the command/status register (CSR). Data transfers to/from the drive will not be terminated immediately. Instead, to prevent the loss of data, they will be terminated at the end of the current sector being processed. Then, the IOPB will return a command complete with error (status 82) and an error code of 77 (hex). The rest of the IOPB will indicate at what point the operation was terminated.

Bit 10, 9 Reserved:

These bits must be zero.

Bit 8 Bus Error:

The V/ESDI 4201 sets this bit to '1' if a bus error has occurred. This bit is both read and modified by the host. If an error occurs, the host must clear this bit (set it to '0').

Bit 7 Go/Busy:

The host sets this bit to '1' in order to start the command defined in the resident IOPB. The V/ESDI 4201 will execute the command, and on completion will clear the Go/Busy bit, set the Operation Done bit, update the Error Last Command bit (see bit 4), and interrupt the host if enabled to do so. The host responds by clearing the Operation Done bit.

Bit 6 Operation Done Interrupt:

The V/ESDI 4201 sets this bit to '1' upon completion of a command. Then, immediately after setting this bit, the V/ESDI 4201 will generate a system interrupt if enabled to do so. However, this bit is set regardless of the interrupt enable status. The host must clear this bit after it has responded to the command completion condition. (See note on following page for more information.)

Bit 5 Status Change Interrupt:

The V/ESDI 4201 modifies this bit only when the Status Change Interrupt enable bit (in the UIB) is set. This bit is set to '1', if as a result of a Seek command, any one of the following drive bits changes from an inactive to an active state (see below for more information):

- DRIVE READY
- FAULT
- ON CYLINDER
- SEEK ERROR

This bit is not set and an interrupt is not generated when these lines toggle during a command which has an implied seek. A change in the Write Protect line does not constitute a status change condition. Immediately after setting this bit, the V/ESDI 4201 will generate a system interrupt. The host clears the status change bit to acknowledge that it has responded to the status change condition.

NOTE

Bits five and six are interrupt bits in the Command/Status Register. They are used to determine the type of interrupt and to acknowledge that the interrupt has been serviced. When either one is set, the drive status (two bytes) and controller status (two bytes) are frozen. The status registers will not change until the interrupt bit is cleared. This is done by resetting the appropriate interrupt bit in the Command/Status Register. The controller can stack up to ten interrupts in this manner. These bits are mutually exclusive such that only one interrupt is provided at a time. The above is true even if the Optional Status Change Register is used, except the status change interrupt is cleared in the Optional Status Change Register instead of the CSR.

This bit and the Status Change Source bit will appear in word 1FE if the Optional Status Change Register is selected. In this case, bit 5 and bit 3 of the CSR will not be used and will remain set to 0.

Bit 4 Error Last Command:

The V/ESDI 4201 sets this bit to '1' if an error occurred during the last command. It sets this bit to '0' if an error was not detected.

Bit 3 Status Change Source:

This bit indicates which drive (1 or 0) was responsible for generating the status change interrupt. If '0', drive zero generated the interrupt. If '1', drive one generated the interrupt. If the optional status change register is used, this bit will not be used and will remain zero.

Bit 2, 1, 0 Reserved Bits:

These bits must be zero.

Drive Status Register

NOTE

When four unit operation is selected, the format of the Drive Status Register does not change. However, its location in memory and the number of drives reported is affected as follows:

	MSB	LSB
Base Address + 506 (1FA hex)	Unit 3 Status	Unit 2 Status
Base Address + 508 (1FC hex)	Unit 1 Status	Unit 0 Status

The Drive Status Register contains two identical bytes which represent the status of the drive(s). The status of the drive connected as unit zero is in the lower byte, while the unit one status is displayed in the upper byte.

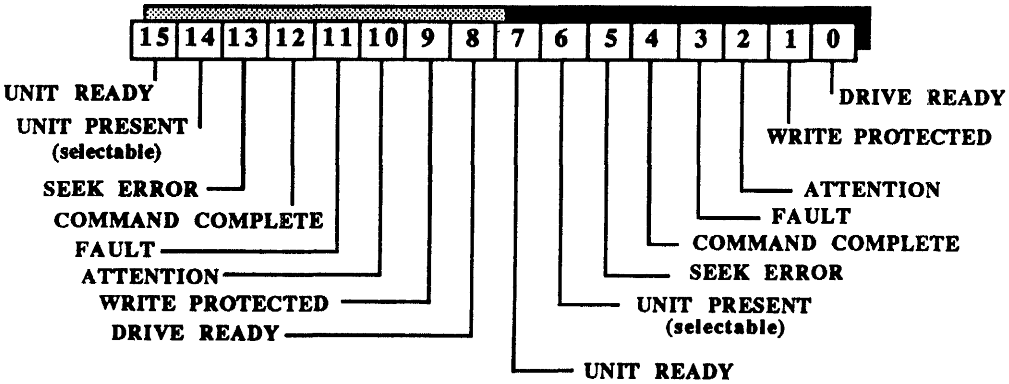


Figure 4-5. Drive Status Register

NOTE

These status bits are continually updated until they are frozen by the occurrence of an interrupting condition. They remain frozen until the interrupting condition is cleared.

The bits of the Drive Status Register are either a direct reflection of the drive status, or they are calculated by the controller based on the drive status indicators. If a '1' is returned, then the status bit is set, if a '0' is returned, then the status bit is not set (for example, if a '0' is returned in the Write Protected field, then the track is not write protected).

Bit 15/7 Unit Ready:

This bit is calculated by the controller. If '1', the drive is ready, on cylinder, and no fault conditions or seek errors have been detected.

Bit 14/6 Unit Present:

This bit is asserted by the controller. If '1', the drive unit has responded to a select sequence.

Bit 13/5 Seek Error:

This signal originates at the drive. If '1', an error occurred while moving the heads during a Seek. The head position is unknown when this occurs. To clear the error, issue a Clear Drive Fault command or a Restore command.

Bit 12/4 Command Complete:

Whenever the drive is executing a command, this signal becomes '0' (false), and once the command is complete, this signal acknowledges it by returning to a '1' (true). In addition, during overlapped Seek operations, this bit is false during the Seek and changes to true once the seek is complete.

Bit 11/3 Fault:

This signal originates at the drive. If '1', the drive has experienced something which caused it to go into a fault condition. All reading and writing is disabled until the fault is cleared. To clear the fault, issue a Clear Drive Fault command or a Restore command

Bit 10/2 Attention:

This signal is issued if the drive wants the controller to request its standard status. This is usually a result of a fault condition.

Bit 9/1 Write Protected:

This signal originates at the drive. If '1', the drive is write protected and must be write enabled before formatting or writing data to disk may proceed.

Bit 8/0 Drive Ready:

This signal is supplied by the drive. If '1', the drive is spun up (up to speed) and is ready to Seek, Read or Write.

Enhancing ESDI Disk Performance

The V/ESDI 4201 incorporates several features which provide the system designer with many alternatives for enhancing overall performance of the disk subsystem. These features include IOPB linking, overlapped seeks, scatter/gather, and others. All of these operations are discussed in the context of providing the optimum performance for specific disk-type transactions. Any questions you may have can be answered by the Interphase applications engineers at (214) 350-9000.

Linked IOPBs

Linking IOPBs increases performance when noncontiguous blocks of data must be transferred to/from system memory. The higher level of performance is achieved by reducing driver overheads and by eliminating the need for the host CPU to service multiple interrupts. In addition, even greater performance is achieved if the linked IOPBs are stored in the V/ESDI 4201 Short I/O space; then, multiple bus requests are eliminated as well.

Linked IOPBs are enabled by setting bit five of the IOPB command options byte (word zero, lower byte). Only one interrupt (if enabled) is generated at the completion of the link. Note that interrupts and vectors may be enabled in all of the IOPBs in the link if it is so desired. If the link is terminated due to an error, the current IOPB (the one being processed) is used to generate the interrupt information. Subsequent IOPBs will not be processed once the link is broken.

Instead of using linked IOPBs to transfer noncontiguous blocks of data to/from system memory to contiguous blocks on the disk, the scatter/gather command should be used to remove rotational latencies.

Scatter/Gather Operation

There are six V/ESDI 4201 commands used for Scatter/Gather operations. Commands A1 and A2 are used for normal 16-bit wide scatter and gather operations. Commands A3 and A4 are used for normal 32-bit wide scatter and gather operations, and commands A5 and A6 are used for 32-bit wide. The scatter commands and gather commands allow the user to place contiguous disk data in noncontiguous areas of system memory (scatter operation) or to place noncontiguous areas of system memory in contiguous areas of the disk (gather operation).

The purpose of the Scatter/Gather commands is to unburden the host of having to process several transactions when noncontiguous blocks of data are being transferred. The commands allow multiple blocks of data to be transferred using only one command, rather than one command per block. So, whenever the Scatter/Gather commands are used, system performance is improved because both the number of VMEbus interrupts and the number of bus transactions associated with common disk activity are minimized.

The disk control aspects of the Scatter command are identical to the Read Sector(s) command (81). Similarly, the disk control aspects of the Gather command are identical to the Write Sector(s) command (82). Zero latency and virtual buffering are operational on Scatter/Gather commands.

Scatter/Gather Lists

The Scatter/Gather commands use a list of elements to control the scatter and gather operations. Each element in the list is used to specify the byte count, address, memory type, and address modifier of a block of data stored in system memory. The host uses the elements to build a separate list that specifies a number of noncontiguous blocks, and the V/ESDI 4201 then uses the host-built list to scatter or gather the data.

The beginning of the V/ESDI 4201 Scatter/Gather list is pointed to by bytes 0A-0F in the IOPB. The total number of elements in the list is specified by byte 1B and the remaining IOPB parameters are identical to corresponding parameters in the Read and Write commands. The lists can reside anywhere in system memory, but for optimum performance, they should be stored in V/ESDI 4201 Short I/O space. The V/ESDI 4201 Scatter/Gather list can take one of two formats depending on the Scatter/Gather command code used. Each element in a list is an eight-byte entry. When either command codes A1 and A2 (16-bit transfers) or A3 and A4 (32-bit transfers) are used, the Scatter/Gather list has the following format:

Table 4-1. A1-A4 Command Codes

WORD	UPPER BYTE	LOWER BYTE
0	Element 1 Byte Count (High)	Element 1 Byte Count (Low)
1	Element 1 Address (A31-A24)	Element 1 Address (A23-A16)
2	Element 1 Address (A15-A08)	Element 1 Address (A07-A00)
3	Element 1 Memory Type	Element 1 Address Modifier
4	Element 2 Byte Count (High)	Element 2 Byte Count (Low)
5	Element 2 Address (A31-A24)	Element 2 Address (A23-A16)
6	Element 2 Address (A15-A08)	Element 2 Address (A07-A00)
7	Element 2 Memory Type	Element 2 Address Modifier
•		
•		
•		
w	Element n Byte Count (High)	Element n Byte Count (Low)
x	Element n Address (A31-A24)	Element n Address (A23-A16)
y	Element n Address (A15-A08)	Element n Address (A07-A00)
z	Element n Memory Type	Element n Address Modifier

Word zero must contain the number of bytes in the data memory block. This parameter must be an even multiple of the sector size. Words one and two contain the starting address of the VMEbus data memory block. The address modifier is designated in bits 0-5 of Word three, and the memory type is defined in bits 8-9. All other bits in Word three are reserved and should be set to '0'.

When command codes A5 and A6 (32-bit transfers) are used, the Scatter/Gather list has the following format:

Table 4-2. A5-A6 Command Codes

WORD	UPPER BYTE	LOWER BYTE
0	Not Used	Not Used
1	VMEbus Memory Type	VMEbus Address Modifier
2	Element 1 Address (A31-A24)	Element 1 Address (A23-A16)
3	Element 1 Address (A15-A08)	Element 1 Address (A07-A00)
4	Element 2 Address (A31-A24)	Element 2 Address (A23-A16)
5	Element 2 Address (A15-A08)	Element 2 Address (A07-A00)
•		
•		
w	Element n Address (A31-A24)	Element n Address (A23-A16)
x	Element n Address (A15-A08)	Element n Address (A07-A00)
y	Element 1 Byte Count (High)	Element 1 Byte Count (Low)
z	Element 2 Byte Count (High)	Element 2 Byte Count (Low)
•		
•		
n	Element n Byte Count (High)	Element n Byte Count (Low)

Scatter/Gather Guidelines

The following guidelines must be observed when using any of the Scatter or Gather commands.

- The list size must be less than or equal to the number of bytes per sector.
- All elements within a list must contain the same VMEbus memory type and address modifier code.
- The byte count must be a multiple of the bytes/sector parameter.

Scatter/Gather Completion

After a Scatter/Gather command has been completed, the V/ESDI 4201 sets IOPB bytes 0A-0F back to the beginning of the list. Then, IOPB bytes eight and nine are updated to include the number of sectors that have not yet been processed. This number will be zero for an error-free transaction. In the event of an error, it is recommended that the entire list be retried since it is sometimes difficult to ascertain exactly which entries have not been transferred.

Overlapped Seeks

The V/ESDI 4201 supports overlapped seeks in addition to the implied seeks embedded in many of the commands. Overlapped seeks are used to minimize seek latency when the V/ESDI 4201 is operating more than one drive. The overlapped seeks allow the host to start a seek on one drive and then complete a data transfer on another drive while the first drive is still seeking.

For example, if the host has a transaction to conduct with the first drive that requires a seek of three cylinders and a transaction with the second drive that requires a seek of twenty cylinders, it can issue a Seek command to the second drive and a Read or Write command with an implied seek to the first drive. After the transaction with the first drive has been completed, the host can either wait for a status change interrupt from the second drive or immediately issue a Read or Write command and ignore the status change interrupt (status change interrupts can also be disabled).

In some systems, the host can issue explicit seeks on both drives and then service whichever drive returns the first status change interrupt. This method frees the host from the sometimes complicated calculations involved in deciding which drive will be able to respond first.

Status Change Interrupts

In multiple drive systems, the Status Change Interrupts are used with overlapped seek operations to improve system performance. The Status Change Interrupt requires the following sequence of events:

- A Seek command is issued.
- An Operation Done interrupt is posted and acknowledged. The host is now free to issue another command.
- The drive completes the Seek and sets its On Cylinder bit active.
- The V/ESDI 4201 sets the status change bit in the CSR (or Optional Status Change register) and posts an interrupt.
- The host performs an interrupt acknowledge cycle which clears the hardware interrupt and fetches a vector to the host's status change interrupt routine.
- The host clears the status change bit.

Internally, the V/ESDI 4201 can have more than one interrupt ready to post at the same time. The following two controls apply:

- First, interrupts are internally queued. The V/ESDI 4201 will never post both an Operation Done and a Status Change condition at the same time. The next pending interrupt is posted only after the present interrupt has been responded to by clearing either the Operation Done or Status Change bit in the CSR or Optional Status Change register.
- And second, a status change condition can be posted even if a command is in progress. Therefore, it is possible for the Go/Busy and the Status Change bits to be active at the same time.

Due to the asynchronous nature of the status change interrupt, its use with some operating systems can prove to be somewhat arduous if not properly handled.

INTERPHASE recommends using the Optional Status Change Register for returning status change interrupt information. This is selected by setting bit seven in the Status Change Interrupt Level (byte 10 hex) of the UIB. The use of this option separates the Status Change Bits (bits three and five in the CSR) from the Operation Done and Go/Busy bits. It is important to note that bits three and five will not be updated in the CSR when this option is used. They are updated in the Optional Status Change Register instead. The bit positions (three and five) remain the same. Other than a different address for the Status Change Register, the details of the status change mechanism remain the same.

If for some reason the above technique is not acceptable, there are other methods for handling status change interrupts. One way to prevent too many interrupts is to disable the interrupt that is normally generated by the Seek command and let the status change interrupt be the one that wakes up a sleeping seek process. However, unless another command can be given while the seek process is asleep, the status change mechanism is no more efficient than the normal implied seek commands.

When status change interrupts are used and the status change bits remain in the CSR, it is possible for the V/ESDI 4201 to be updating the CSR at the same time the host is issuing a Go to the CSR. There is a small possibility that the Go will be overwritten by the V/ESDI 4201 and be lost.

One possible method for avoiding this situation is to set a timer when a command is issued and then retry the command if a timeout occurs. And since setting a command timeout is good design practice, the probability of the conflict is small.

A second way of handling the conflict is to issue the command and then read the CSR to make sure the Go/Busy bit remained set. If it did not, then the host may simply issue another Go.

THIS PAGE LEFT BLANK INTENTIONALLY

5 INPUT/OUTPUT PARAMETER BLOCK (IOPB)

IOPB Format

The Input/Output Parameter Block (IOPB) allows the host CPU to communicate with the V/ESDI 4201 and issue the necessary commands. Although some commands do not need certain parameters, the length of the IOPB is always 14 words. The IOPB starts at the base address plus four in the Short I/O space of the controller. The format of the IOPB is shown in Table 5-1 below.

Table 5-1. IOPB Format

IOPB Format			
Word # (DEC)	**Offset (HEX)**	**High Byte**	**Low Byte**
0	004	Command Code	Commands Options
1	006	Status Code	Error Code
2	008	Cyl. High (Logical D24-31)	Cyl. Low (Log. D16-D23)
3	00A	Head # (Logical D08-D15)	Sector # (Logical D00-D07)
4	00C	Sector Count High	Sector Count Low
5	00E	Buffer Address (A24-A31)	Buffer Address (A16-A23)
6	010	Buffer Address (A08-A31)	Buffer Address (A00-A07)
7	012	Memory Type Address	Modifier Code
8	014	Opt. Drive #/Int.Level (1-7)	Normal Complete Int. Vec.
9	016	DMA Transfer Count	Error Interrupt Vector
10	018	IOPB Pointer (A24-A31)	IOPB Pointer (A16-A07)
11	01A	IOPB Pointer (A08-A15)	IOPB Pointer (A00-A07)
12	01C	IOPB Memory Type	IOPB Address Mod. Code
13	01E	Absolute Skew	Entry Count

The IOPB is discussed in detail in the following subsection of the user's guide.

- Command Code - Word 0, upper byte
- Command Options - Word 0, lower byte
- Status Codes - Word 1, upper byte
- Error Codes for - Word 1, lower byte
- Commands Completed with Exception Error Codes for - Word 1, lower byte
- Commands Completed with Error IOPB Parameters - Words 2-13

The individual command codes and error codes (for commands completed with error) are discussed in Appendices C, and D respectively.

Command Codes: IOPB Word Zero, Upper Byte

The upper byte of word zero of the IOPB specifies commands such as Read or Write data from the disk. The command options are contained in the lower byte of word zero. These options allow the user to specify such parameters as volume and unit number, as well as operating modes such as logical or physical disk addressing modes.

The upper byte of the first word in the IOPB (word zero) is the command information that must be provided to the V/ESDI 4201 by the host. The following table lists all of the available commands in ascending hexadecimal code order. Each command is defined in Appendix C.

Table 5-2. V/ESDI 4201 Disk Control Commands

V/ESDI 4201 Disk Control Commands

Code	Command	Code	Command
70	DIAGNOSTICS	8A	SEEK
71	READ LONG	8B	REFORMAT
72	WRITE LONG	8C	FORMAT TRACK WITH DATA
74	READ HEADER	90	MAP SECTOR
75	READ ESDI FLAW MAP	91	READ SECTORS SEQUENTIAL
76	READ CDC ESDI FLAW MAP	92	WRITE SECTORS SEQUENTIAL
77	REPORT CONFIGURATION	93	VERIFY SECTORS SEQUENTIAL
78	WRITE SECTOR BUFFER	94	READ NONCACHED
79	READ SECTOR BUFFER	95	READ SEQ., DISABLE ADDR. BUMP
7A	REPORT SECTOR ID	96	WRITE SEQ., DISS ADDR.BUMP
7B	FORMAT WITH SECTOR ID	97	CLEAR DRIVE DEFAULT
7C	INITIALIZE LONG	99	VERIFY TRACK
7D	REPORT CONGL LONG	9A	TRACK ID
7E	READ FLAW MAP HEADER	9B	FETCH AND EXECUTE IOPB
7F	GOTO MACSI	9C	VERIFY TRACK SEQUENTIAL
80	WRITE AFTER CACHE	9E	EXTENDED DIAGNOSTICS
81	READ SECTOR(S)	A0	ESDI PASS-THROUGH
82	WRITE SECTOR(S)	A1	READ AND SCATTER
83	VERIFY SECTOR(S)	A2	GATHER AND WRITE
84	FORMAT TRACK	A3	READ AND SCATTER 32
85	MAP TRACK	A4	GATHER AND WRITE 32
86	HANDSHAKE	A5	READ AND SCATTER 32 (LIST 2)
87	INITIALIZE	A6	GATHER AND WRITE 32 (LIST 2)
89	RESTORE (RETURN TO 0)		

Command Options: IOPB Word Zero, Lower Byte

The IOPB command options are contained in the lower byte of Word zero of the IOPB and are used to give the V/ESDI 4201 instructions for executing commands. The format of the command option byte is shown in the IOPB Format Table, and is followed by a short description of each option.

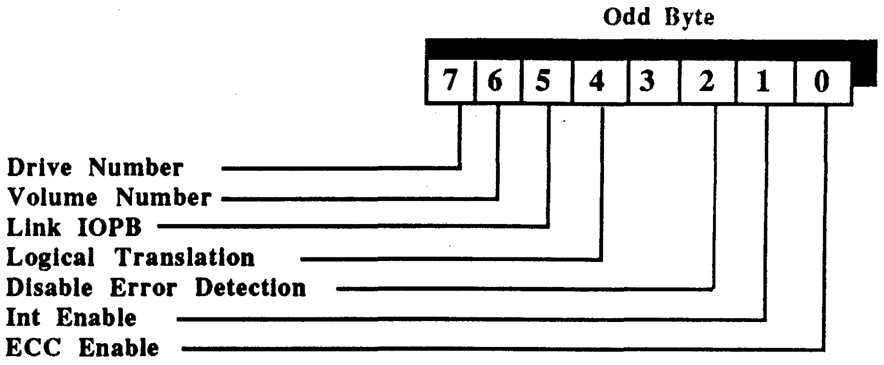


Figure 5-1. Command Options Byte Format

Bit 7 Drive Number:

This bit specifies the drive (zero or one) on which the command is to be performed. If bit six of UIB byte F (see initialization section) is set to '1' indicating four unit operation, this bit is no longer used. For four unit operation, the unit number is specified in the upper byte of IOPB word eight, bits 12 and 13.

Bit 6 Volume Number:

This bit specifies upon which volume (zero or one) of the specified drive the command is to be performed. This bit is only necessary when in logical translation mode (when bit four is '1'). Volume boundaries are defined in the UIB.

Bit 5 Link IOPB:

Setting this bit to '1', enables IOPBs to be linked. The process for linking IOPBs starts by setting this bit in the resident IOPB (i.e., in the first IOPB) in the case of a Fetch and Execute IOPB command. The host CPU decides where the linked IOPBs will reside and must provide the address of the next IOPB in the address pointer of the base IOPB (Words 10 and 11). Each linked IOPB in turn has its link bit set, and points to the next IOPB in the series. The status of each linked IOPB is displayed in its status word (Word one), and the host can track the status of each IOPB individually. Thus, if an error occurs and the link is broken, the host can determine which IOPB was responsible and read its status and error code. The link bit in the final IOPB must be zero. If this bit is set at '0', it indicates that no linking is to be done and the IOPB address pointers are ignored.

Only one interrupt per set of linked IOPBs is generated. This interrupt occurs upon completion of the link, whether the completion is normal, with error, or with exception. The data the V/ESDI 4201 needs for an interrupt (interrupt enable bit and interrupt vector) are obtained from the last completed IOPB.

Bit 4 Logical Translation:

If this bit is set at '1', the logical sector number is first translated to a physical cylinder, head, and sector number before execution. Logical translation can only be used with the following commands: Read Sector, Write Sector, Read Long, Write Long, Read Header, Seek, Read Noncached, Diagnostics, and Verify Sector. During any other command, such as Format or Map Track, this bit is ignored and physical addressing must be used. If this bit is '0', physical addressing must be used for all commands.

Bit 3 Reserved:

This bit is reserved and must be set to '0'. BIT 2 DISABLE ERROR DETECTION: If this bit is set at '1', it disables error detection on the data fields. If '0', the data is checked for errors on every disk read operation.

Bit 1 Int Enable:

If this bit is set at '1', it enables the system interrupt to be generated upon command completion. If '0', no interrupt is generated and the host must monitor the Operation Done bit in the CSR to detect completion.

Bit 0 ECC Enable:

If this bit is set at '1', the error correction algorithm is applied to erroneous data fields after all retries and/or reseeks have been exhausted. If '0', no error correction is attempted.

Command Status Codes: IOPB Word One, Upper Byte

Word one of the IOPB contains the status and error codes. The status code resides in the upper byte and is used to reflect to the host, the execution status of the current IOPB. The status code also provides the status of individual IOPBs when a series of IOPBs are linked. If only one IOPB is in system memory, the status code will contain the same command completion information as the CSR. The IC ?B status codes are updated prior to updating the CSR.

Table 5-3. Command Status Codes

Command Status Codes	
Code (Hex)	**Status**
80	Command Completed With No Error
81	Command In Progress
82	Command Completed With Error
83	Command Completed With Exception

80 Command Completed With No Error:

This indicates that the command has been completed without any errors or exceptions. The Command/Status Register will display Operation Done as '1', Error Last Command as '0', and Busy as '0'.

81 Command In Progress:

This code indicates that the IOPB is currently being executed. This code is displayed after the GO bit is '1' in the Command/Status Register.

82 Command Completed With Error:

In the event of an error, the appropriate code for the error is contained in the lower byte of Word one. In the Command/Status Register Operation Done and Error Last Command are '1', and Busy is '0'. Errors do not occur in normal operation, so the user should investigate the error when this code is displayed.

83 Command Completed With Exception:

This code indicates that the command was completed successfully, but that some method of error recovery was required. The appropriate error information will be contained in the lower byte of Word one. The user may want to log the exception to indicate that system troubleshooting may be required.

Error Codes: IOPB Word One, Lower Byte

If a status code of 80 or 81 is returned in the upper byte of word one, then this byte will be all zeros. If a status code of 82 is returned, this byte will contain the error code. The error code is used to indicate to the host which error occurred while executing the IOPB. Appendix D contains a complete list of error codes for commands completed with error and a brief description of each one.

If a status code of 83 is returned, then this byte will indicate that the command was completed with exception. The exception codes are shown below.

Error Codes For Commands Completed With Exception

Bit 7

If this bit is set at '1', error correction was applied to the data field. If it is set at '0', error correction was either not required or was disabled.

Bit 6

If this bit is set at '1', it indicates that the data transferred into system memory may be erroneous. This bit is used in conjunction with the Move Bad Data bit in the UIB attribute flags (byte E, bit one), which allows the transfer of erroneous data into memory, and the command option Disable Error Detection (bit two of the IOPB command options).

Bit 5

Bit five is a reserved bit and must be set at '0'.

Bit 4

If this bit is set at '1', a Restore and Reseek was performed. The ability to perform a Restore as part of the error recovery is enabled in the UIB.

Bits 3-0

These bits are used to indicate the number of rotational retries that was attempted. The maximum number that can be displayed is 15. If 15 or more retries were attempted, these bits will all be '1'. The number of rotational retries to attempt is specified in the UIB.

Error Codes For Commands Completed With Error

Errors do not occur during normal operation and the source of the error should be determined by the user. Codes that are not used for a specific error have been omitted (i.e., these codes should never be posted). For a complete description of error codes for commands completed with error, refer to Appendix D.

Table 5-4. Error Codes

Error Codes			
Error Code	**Description**	**Error Code**	**Description**
10	Disk not Ready	42	GAP Specification Error
12	Seek Error	4B	Seek Error
13	ECC Error-Data Field	50	Sectors/Track Error
14	Invalid Command Code	51	Bytes Sector Spec Error
15	Illegal Fetch and Execute	52	Interleave Spec Factor
16	Invalid Sector in Command	53	Invalid Head Address
17	Illegal Memory Type	54	Invalid Cylinder Address
18	Bus Timeout	55	ESDI Command Complete Timeout
19	Header Checksum Error	56	Zero Sector Count
1A	Disk Write Protected	5D	Invalid DMA Transfer Count
1B	Unit Not Selected	60	IOPB Failed
1C	Seek Error Timeout	61	DMA Failed
1D	Fault Timeout	62	Illegal VME Address
1E	Drive Faulted	6A	Unrecognized Header Field
1F	Ready Timeout	6B	Mapped Header Error
20	End of Medium	6E	Spare Sector Spec. Error
21	Translation Fault	6F	No Spare Sector Enabled
22	Invalid Header Pad	77	Command Aborted
23	Uncorrectable Error	78	ACFAIL Detected
24	Translation Error, Cylinder	80	XFER Assertion Timeout
25	Translation Error, Head	81	XFER Release Timeout
26	Translation Error, Sector	82	Status Assertion Timeout
27	Data Overrun	83	Status Release Timeout
28	No Index Pulse on Format	A0	S/G List Too Large
29	Sector Not Found	A1	Illegal Element Byte Count
2A	ID Field Error, Wrong Head	AB	Illegal Element Size
2B	Invalid Sync in Data Field	AC	Illegal List Byte Count
2C	No Valid Header Found	AD	Illegal IOPB Sector Byte Count
2D	Seek Timeout Error	AE	Illegal Element Count
2F	Not on Cylinder	C0	Both Bits Set
30	RTZ Timeout	C1	MSE Without Initialize Long
31	Invalid Sync in Header	F0	Mapped Header
3E	UIB Skew Factor	F1	Sector Not Flagged
3F	No Heads Specified	FC	No Write List
40	Unit Not Initialized	FD	No Write Buffers
FF	Command Not Implemented	FE	Out of Buffers
41	Not Used		

IOPB Words 2-13

The first two words of the IOPB (Words zero and one) contain the command, status, and error codes as well as the user selectable command options. These words have been discussed in detail in the preceding sections of the user's guide.

The remaining words in the IOPB provide additional information to the V/ESDI 4201 to execute the desired command. These words specify parameters such as spiral skew, head and cylinder number, memory type, and buffer address. The head number, cylinder number and sector count words are automatically updated by the V/ESDI 4201 upon completion of the command. This feature is most often used as a diagnostic aid in cases of partially successful completion of a transaction. Table 5-5 shows the format of the remaining words, and the following pages define the operating parameters specified by those words.

Table 5-5. IOPB Words 2-13

IOPB Words 2-13			
Address			**Word No.**
Offset	**Upper Byte**	**Lower Byte**	**(DEC)**
008	Cylinder High (Log. D24-D31)	Cylinder Low (Log. D16-D23	2
00A	Head # (Logical D08-D15)	Sector # (D00-D07)	3
00C	Sector Count High	Sector Count Low	4
00E	Buffer Address (A24-A31)	Buffer Address (A16-A23)	5
010	Buffer Address (A08-A15)	Buffer Address (A00-A07)	6
012	Memory Type	Address Modifier Code	7
014	Opt. Drive #/Int. Level (1-7)	Normal Complete Int. Vector	8
016	DMA Transfer Count	Error Interrupt Vector	9
018	IOPB Pointer (A24-A31)	IOPB Pointer (A16-A23)	10
01A	IOPB Pointer (A08-A15)	IOPB Pointer A00-A07)	11
01C	IOPB Memory Type	IOPB Address Modifier Code	12
01E	Absolute Skew	Entry Count	13

Cylinder Address (Word 2)

This word contains the cylinder address. When the V/ESDI 4201 is operating in physical mode, Word two is the actual cylinder address where the read/write heads are to be moved. In logical translation mode, Word two is the upper word of the two-word logical address. If an error occurs during a Read or Write, the cylinder address will point either to the physical or logical error location.

Head and Sector Numbers (Word 3)

The head and sector numbers are specified by Word three. In physical mode, the head number is in the upper byte of the word, and the sector number is in the lower byte. In logical translation mode, Word three is the lower word of the address. If an error occurs, Word three is used to point either to the physical head and sector or to the logical sector where the error occurred.

Sector Count (Word 4)

The sector count, or number of sectors to be transferred, is specified in Word four.

Buffer Address (Words 5 and 6)

These words contain the buffer address, or memory address where the data is to be transferred to or from. If an error occurs, the buffer address will point to the start of the sector buffer in system memory where the bus transfer was when the error occurred.

Memory Type/Address Modifier Codes (Word 7)

This word contains the memory type and address modifier codes. The upper byte contains the memory type code as detailed below in Table 5-6.

Table 5-6. IOPB (Word 7) Memory Type Codes

IOPB (Word 7) Memory Type Codes

Code	Memory Type
01	16-Bit Internal Memory (not valid for disk data transfers)
02	16-Bit (Word-Wide) Data Transfers
03	32-Bit (Long Word) Data Transfers

When using memory type 03 (long word), if the buffer address is not aligned on a VMEbus-specified long word boundary, the V/ESDI 4201 will transfer that block of data using 16-bit transfers. If this happens, the V/ESDI 4201 will change the memory type to an 02 before indicating command complete.

If bit two of the memory type is set to '1' (codes four through seven), this disables the incrementing of the VME address counters during data transfers. This is useful when transferring data to a slave device whose data buffer is a single address. This bit is only interrogated when reading/writing data sequentially (command codes 91, 92, and 9A). All three memory types (D16, and D32) are supported.

The lower byte contains the address modifier codes. When the V/ESDI 4201 is transferring data to or from system memory, 6-bit address modifiers are used to indicate the type of memory access to be made for a particular address. These modifiers range from 0 to 3F. Refer to the VMEbus specification for additional address modifier information.

Optional Drive Number & Interrupt Level/Vector (Word 8)

The V/ESDI 4201 has seven levels of interrupts which are software programmable on a per command basis. Each IOPB must specify an interrupt level, and can contain two interrupt vectors, one for normal command completion and one for abnormal command completion (see Word nine, lower byte). This word contains the interrupt level in the upper byte (bits 8-11) and the interrupt vector for normal command completion and command completed with exception in the lower byte. The interrupts can be stacked up to ten at a time. If two interrupts occur at the same time, status is presented for one and the interrupt is generated. Then, after the first interrupt is serviced and cleared, the second interrupt is generated.

If bit 6 of UIB byte F is set to '1' to indicate four unit operation, then bits 12 and 13 of the upper byte of word eight are used to specify the drive number. For two unit operation (bit six of UIB byte F is '0'), bits 12 and 13 are reserved and must be set to '0'. Bits 11, 14 and 15 of the upper byte of IOPB Word 8 are reserved and must be set to '0' at all times.

DMA Transfer Count and Error Interrupt Vector (Word 9)

The DMA transfer count is contained in the upper byte of Word nine. The transfer count is used to specify how many transfers of data the V/ESDI 4201 is to perform before releasing the VMEbus and re-requesting it. After the VMEbus is released, it will be re-requested within 300 nanoseconds. The DMA transfer count allows higher priority devices to request and be granted control of the bus. In systems not using the priority arbitration option, if the transfer count is '0', the bus is released only after the entire sector of data is transferred. It should be set to '0' in systems using priority arbitration, since the arbitrator will assert Bus Clear when a higher priority device requests the bus. In this event, the V/ESDI 4201 will release the bus within two transfers of the assertion of the Bus Clear signal.

Changes in the transfer count between transactions should be minimized because they require firmware overhead to recalculate where the transfers will occur. The minimum transfer count is two (2).

The lower byte of Word nine contains the error interrupt vector for abnormal command completion.

Address Pointers (Words 10 and 11)

These two words contain the address pointer to the linked IOPB when running in linked mode. Word 10 contains pointer addresses A24-A31 and A16-A23, in the MSB and LSB respectively. Word 11 contains pointer addresses A08-A15 and A00-A07, in the MSB and LSB respectively.

Memory Type and Address Modifier Codes (Word 12)

The linked IOPB memory type codes and address modifier codes are contained in the upper and lower bytes of Word 12, respectively. The memory type codes are listed below in Table 5-7.

Table 5-7. Linked IOPB (Word 12) Memory Type Codes

Linked IOPB (Word 12) Memory Type Codes

Code	Memory Type
01	16-Bit Internal Memory
02	16-Bit (Word-Wide) Memory
03	32-Bit (Long Word) Memory

The 16-bit internal memory (01) is for the IOPB that already resides in V/ESDI 4201 Short I/O space. For linked IOPB operations, memory type 01 indicates that the next linked IOPB has been transferred into the controller's Short I/O space and is no longer in external system memory.

When using memory type 01, the IOPB pointer points to the first word of the IOPB. For example, if the Short I/O space is set at address 8600 (see Section 2) and the IOPB is loaded 30 bytes from the start of the Short I/O, the IOPB pointer for memory type 01 is 8630.

The purpose of using memory type 01 is to increase speed by reducing the number of bus accesses needed. When this type of memory is specified, external IOPBs do not have to be fetched before they can be executed; thus, the time required for two bus accesses and for transferring the IOPB data is saved.

When external IOPBs are fetched (memory type 02 or 03), they are not moved to the controller's Short I/O space which is reserved for the resident IOPB.

When using memory type 03, if the linked IOPB address is not aligned on a VMEbus-specified long word boundary, the V/ESDI 4201 will transfer that block of data using two 16-bit transfers. Note that after completion, the IOPB memory type will be changed to an 02.

The linked IOPB address modifier code is in the lower byte of Word 12. The address modifier codes are defined by the VMEbus specification. This byte is a "Don't Care" if the memory type is set to 01.

Absolute Skew and Entry Count (Word 13)

The upper byte of this word is used when a skew factor is required other than the skew value specified in the UIB. The upper byte of Word 13 specifies this skew factor. If this byte is nonzero during a Format, the value of this byte is used as the skew factor for each track. If this byte is zero, the skew factor specified in the UIB for this drive unit will be used.

There is an important difference between the skew value in the UIB and the absolute skew specified here in the IOPB. The skew in the UIB is used with the cylinder number and the head number to calculate the number of sectors between the index pulse and the first logical sector (sector zero) on the track. The skew value here in the IOPB (if nonzero) is taken as an absolute value and fixes the first logical sector regardless of the location of the track on the drive.

The lower byte of Word 13 is used with the Scatter/Gather commands to specify the number of entries in the Scatter/Gather list.

6 MULTIPLE ACTIVE COMMAND SOFTWARE INTERFACE (MACSI)

MACSI Overview

The host processor communicates with an Interphase MACSI controller through a section of dual ported memory (typically 512 bytes) that is mapped into the Short I/O space of the VMEbus. All commands and responses pass through this dual ported memory.

The MACSI System Interface allows the controller to have multiple commands active simultaneously. The controller accepts commands from the host and queues them internally. The controller then acts on each command as soon as possible within the confines of the peripheral interface. As commands are completed, the host is notified of both the completion of the command and the status of the completion.

Command Queues

Each command to the controller is fully specified using a software structure that is generated by the host, called an Input/Output Parameter Block (IOPB). The host submits commands to the controller by making an entry into a circular queue called a Command Queue. Each entry in the Command Queue is called a Command Queue Entry (CQE). Each CQE is a 12-byte block that contains various command specific control information and a pointer to the actual IOPB.

Work Queues

To issue a command, the host first builds the IOPB in an area of the Short I/O space specifically reserved for use by the host. It then puts a CQE into the next available slot in the Command Queue and sets the "GO" bit. On seeing the "GO" bit, set the controller moves the control information (from the CQE) and the IOPB for that particular command into another internal structure called a Work Queue. After the CQE and the IOPB have been placed in a Work Queue, the slot in the Command Queue that was filled by the command becomes available for reuse by the host. The controller then performs commands from the Work Queue at the first opportunity.

Command Response Block

When a command is complete, the control information and the IOPB is written by the controller into the Command Response Block, (also in Short I/O space) and an interrupt to the host is generated. The Command Response Block provides command completion status and other information that enable the host to identify which command has completed. The host acknowledges the interrupt by writing a word to the Command Response Block which then releases it for further use.

Performance Issues

The number of CQEs that can be placed in the Command Queue is under programmatic control at initialization time. Eight or less is usually sufficient because a CQE occupies a slot in the Command Queue for such a very short time (until it is moved into a Work Queue). As a result, the host can virtually always issue a command to the Command Queue. Thus, the host is not constrained by any timing issues of the controller's Command Queue.

In the unlikely case that the Command Queue is full when the host tries to enter a command, a MACSI controller provides for efficient operation by optionally interrupting the host when a slot becomes available in the Command Queue.

The Work Queue concept is integral to the way that MACSI allows multiple commands to be active simultaneously. Information in the IOPB determines the Work Queue into which a particular command is placed. The various Work Queues can be prioritized so that commands in one queue take precedence over commands in another queue. Commands for disk operations may optionally be sorted into the Work Queue as they are received from the host.

MACSI accommodates multiple Work Queues. Each Work Queue is dedicated to a particular unit. The characteristics of these Queues, as well as other controller operating parameters are programmable and must be initialized before use.

MACSI offers the system designer maximum flexibility while still offering superior performance. Command response time as well as command parsing time are overlapped with peripheral activity. And the Command Queue has been designed to minimize Command Latency. In addition, multiple Work Queues provide all of the flexibility that is required when implementing today's newest class of high performance systems.

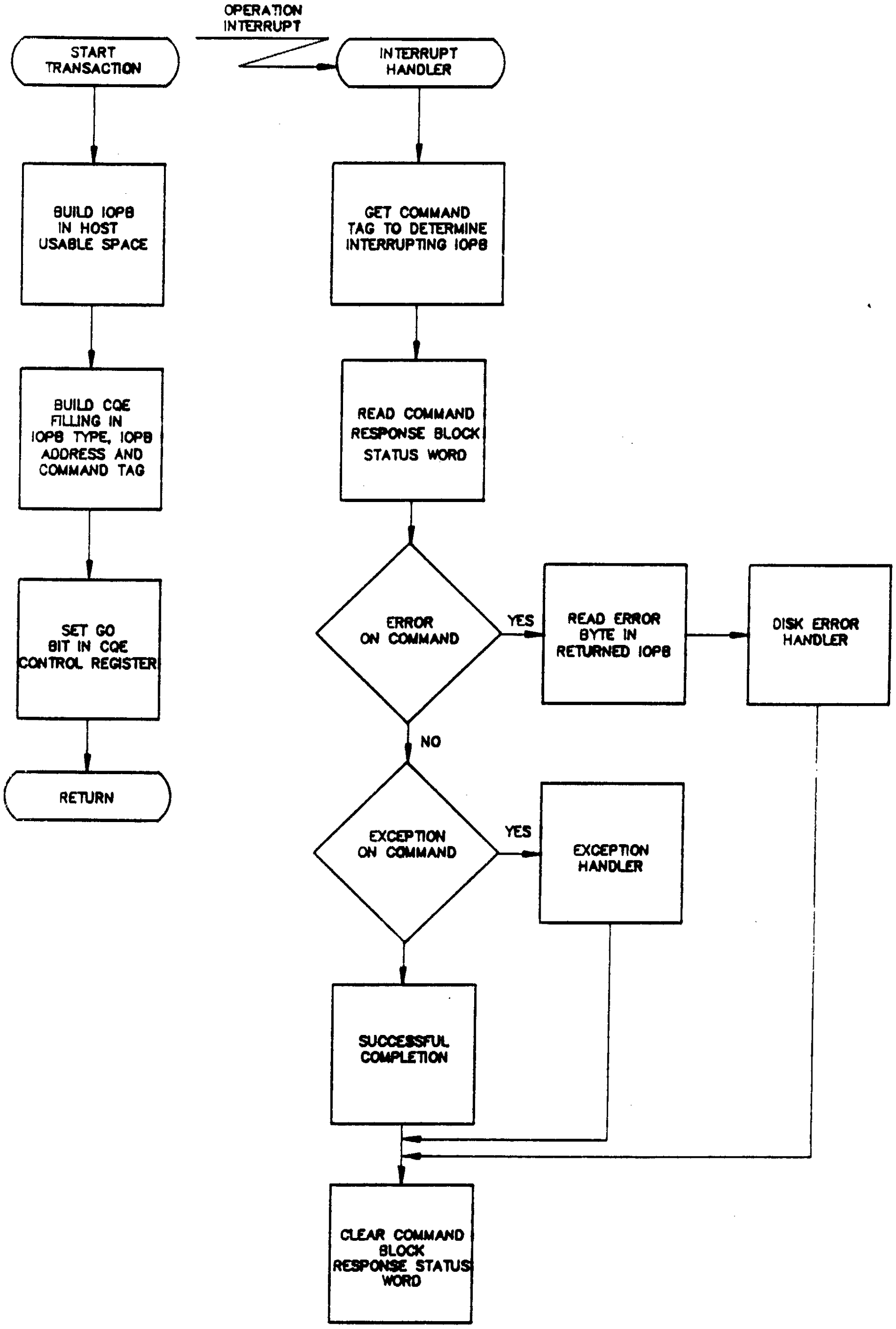

OPERATION INTERRUPT
START TRANSACTION
INTERRUPT HANDLER
BUILD IOPB IN HOST USABLE SPACE
GET COMMAND TAG TO DETERMINE INTERRUPTING IOPB
BUILD CQE FILLING IN IOPB TYPE, IOPB ADDRESS AND COMMAND TAG
READ COMMAND RESPONSE BLOCK STATUS WORD
SET GO BIT IN CQE CONTROL REGISTER
ERROR ON COMMAND
YES
READ ERROR BYTE IN RETURNED IOPB
DISK ERROR HANDLER
NO
RETURN
EXCEPTION ON COMMAND
YES
EXCEPTION HANDLER
SUCCESSFUL COMPLETION
CLEAR COMMAND BLOCK RESPONSE STATUS WORD

Figure 6-1. MACSI Control Structures Block Diagram

GoTo MACSI Mode Command

The GoTo MACSI Mode Command is used to convert from Non-MACSI operation to MACSI operation.

The GoTo MACSI Mode (7F) command is placed in the command field of the IOPB the same way as any other non-MACSI mode. What is different is the block of data used by the command, and the way the command is completed.

The entire VMEbus Short I/O space is changed during this command. When the V/ESDI 4201 executes the GoTo MACSI Mode, it first checks the command for proper format and parity. If an error is detected, command execution is terminated in non-MACSI mode.

Within 200 micro-seconds after the Go bit is set, the V/ESDI 4201 sets the CNA bit in the MSR, and clears the CRBVA bit in the CRB. The V/ESDI 4201 then clears all of the Short I/O space, before clearing CNA, and setting CRBVA.

The V/ESDI 4201 obtains the buffer address, IOPB type, and modifier from the IOPB. The buffer address points to a Controller Initialization Block (CIB) supplied by the VMEbus driver. The CIB is 16 words long, and contains parameters that the V/ESDI will use. The CIB is shown in the following figure, and the words are described in the list after the figure:

	Disk MACSI CIB Block															
Hex Offset	Even Byte								Odd Byte							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000	Number of CQEs															
002	CRB Offset															
004	FLAGS															
006	Number of Commands to Q															
008	RESERVED MUST BE 0															
00A	RESERVED MUST BE 0															
00C	Back to Back Write Count															
00E	Round-Robin Count															
00E-1E	RESERVED MUST BE 0															

Figure 6-2. MACSI CIB Block

Number of Command Queue Entries

Defines the number of Command Queue Entries. A maximum of eight are allowed.

Command Response Block Offset

Contains a relative address for the Command Response Block.

MACSI Flags

Contains the flags used in the MACSI mode. The following flags are used:

- Bit 0 - Not used
- Bit 1 - Enable overlap seeks
- Bit 2 - Enable automatic drive fault clearing

Number of Command Queue Entries

Defines the total number of commands that can be queued. A maximum of 64 can be used.

Back to Back Write Count

Back to back write count, defines the maximum of locations in the work queue that the V/ESDI 4201 is to use when doing grouped write operations. No interrupts are sent to the VMEbus host until the last grouped write operation is completed.

Round-Robin Count

Round Robin Count, causes the controller to proceed on a work queue, until a cylinder switch occurs, or the Round Robin Count goes to 0. A Round Robin Count of 0 will force an automatic work queue switch to occur on every command completion.

GoTo MACSI Mode Error Conditions

Several types of error conditions can occur as a result of the GoTo MACSI Mode Command. These error conditions are discussed briefly in the following list. The various registers and conditions are discussed in the previous sections of this chapter.

- E0H - Already in MACSI Mode: The GoTo MACSI Mode command is issued, and the V/ESDI 4201 is already in the MACSI Mode.
- E1H - Bad CIB address: The address given for the V/ESDI 4201 Unit Initialization Block is invalid. Address 0 is invalid.
- E3H - Number of CQEs is incorrect: The number of CQEs must be greater than 0 and less than 9.
- E9H - Number of Commands to queue is invalid: The number of commands to queue must be greater than 0 and less than 65.

The host processor communicates with the 4201 through 512 bytes of dual ported memory that is mapped into the Short I/O space of the VMEbus. All commands and responses pass through this dual ported memory. The new system level interface of the 4201 is implemented with the Interphase Multiple Active Command Software Interface (MACSI).

This document describes the Multiple Active Command Software Interface (MACSI) of the 4201. The MACSI System Interface allows the host to not only have multiple commands queued but also for the controller to have multiple commands active simultaneously. The controller accepts commands from the host and queues them internally. The controller then acts on each command as soon as possible within the confines of the disk interface. As commands are completed, the host is notified of both the completion of the command and the status of the completion.

Each command to the controller is fully specified using a software structure that is generated by the host, called an Input/Output Parameter Block (IOPB). The host submits commands to the controller by making an entry into a circular queue called a Command Queue. Each entry in the Command Queue is called a Command Queue Entry (CQE). Each CQE is a 12-byte block containing a pointer to the IOPB and various control information.

To issue a command, the host first builds the IOPB in an area of the Short I/O space specifically reserved for use by the host. It then puts a CQE into the next available slot in the Command Queue. The controller reads the Command Queue and places the control information from the CQE and the IOPB for that particular command into one of another set of internal structures called Work Queues. After the CQE and the IOPB have been placed in a Work Queue, the slot in the Command Queue that was filled by the command becomes available for reuse by the host. The controller then automatically performs commands from the Work Queue at the first opportunity.

When a command is complete, the control information and the IOPB is written by the controller into the Command Response Block, (also in Short I/O space) and an interrupt to the host is generated. The Command Response Block provides completion status and other information that enable the host to identify which command has completed. The host acknowledges the interrupt by writing a word to the Command Response Block which then releases it for further use.

The Command Queue has slots for a number of CQEs. In addition, a CQE occupies a slot in the Command Queue for a very short time (until it is moved into a Work Queue). As a result, the host will virtually always have slots available in the Command Queue for issuing commands. Thus, the host need not be concerned with any of the intimate timing issues of the controller's Command Queue. Even in the unlikely case that the Command Queue is full when the host tries to enter a command, the controller provides efficient operation by optionally interrupting the host when a slot becomes available in the Command Queue.

The concept of Work Queues is integral to the way that MACSI allows multiple commands to be active simultaneously. Information in the IOPB determines the Work Queue into which a particular command is placed. The various Work Queues can be prioritized so that commands in one queue take precedence over commands in another queue. Commands for disk operations may optionally be sorted into the Work Queue as they are received from the host.

The 4201 accommodates up to 11 Work Queues. Each Work Queue is dedicated to a particular unit. Four are reserved for the disk and seven for the optional SCSI adapter. The characteristics of these Queues, as well as other controller operating parameters are programmable and must be initialized before use. In order to accommodate this initialization, the controller, on power-up, is in 2-unit Non-MACSI mode. Short I/O space takes exactly the same form as described earlier.

System Interface Detailed Description

The host communicates with the controller through 512 bytes of dual ported memory located on the controller. This memory is mapped into the Short I/O space of the VMEbus. Every location can physically be both written to and read from by the host at any time, but the protocol of the MACSI System Interface puts some restrictions upon when certain areas should be accessed. Also, some areas are logically write only or read only. MACSI partitions this RAM into five major sections:

- Master Control/Status Block (MCSB)
- Command Queue (CQ)
- Host Usable Space (HUS)
- Command Response Block (CRB)
- Device Status Block (DSB)

The Master Control/Status Block (MCSB) is used to pass and receive information relative to the overall operation of the controller. The Master Control/Status Block is 16 bytes long.

The Command Queue (CQ) consists of a programmable number of slots for Command Queue Entries (CQE). Each CQE includes all of the information that is needed for the host to find, execute, and respond to the commands contained in an IOPB. The Command Queue is circular, and it is up to the host to keep track of the next Command Queue Entry that it can use. Because the queue is circular, the controller infers chronological ordering of commands. Each Command Queue Entry is "busy" only until the controller can transfer the command to a Work Queue and then free its slot in the CQ. This mechanism allows a large number of commands to be queued up without needing an excessive number of slots in the Command Queue itself. The number of slots in the Command Queue is programmed via the GOTO MACSI MODE command. The actual size of the Command Queue equals the number of slots times 12 bytes. In the 4201, the Command Queue can have a maximum of 8 slots.

The Host Usable Space (HUS) is free-form memory space accessible to both the host and the controller. It is typically used for IOPBs. However, for multiprocessing applications, this is a convenient place for semaphores between CPUs. The amount of Host Usable Space available is determined by the number of Command Queue slots defined when the CQ is initialized and by the length of the Command Response Block defined. The Master Control/Status Block, and the Device Status Block always occupy a total of 40 bytes.

The Command Response Block (CRB) is used by the controller to post completion status in two situations: either as a response to a command completion or a queue slot available condition. If the posting is the result of an IOPB completion, the IOPB itself and related status information are returned to the CRB. The offset of the Command Response Block is defined at initialize time. The length of the Command Response Block can be determined by subtracting the Command Response Block offset from the offset of the Device Status Block (0x1E8H). However, the length of the Command Response Block must be equal to the largest IOPB defined plus 12 bytes. A recommended offset is 1A0H.

The Device Status Block (DSB) is used by the controller to post various unit statuses.

Offset	
000H (000) 00FH (015)	Master Control/Status Block (MCSB)
010H (016) 06FH (111)	Command Queue
070H (112) 1A5H (421)	Host Usable Space (HUS)
1A0H (422) 1DFH (488)	Command Response Block (CRB)
1E0H (488) 1FFH (511)	Device Status Block (DSB)

Figure 6-3. Example 4201 MACSI System Interface Memory Map

This memory map is for a controller that has been initialized with an 8-slot Command Queue IOPB length of 32 bytes and a CRB offset of IA0H.

Master Control/Status Block (MCSB)

The Master Control/Status Block (MCSB) consists of the Master Status Register (MSR), the Master Control Register (MCR), the Interrupt on Queue Available Register (IQAR), and the Queue Head Pointer (QHP).

Even								Odd							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Master Status Register															
Master Control Register															
Queue Head Pointer															
Interrupt On Queue Available Register															
Reserved															
Reserved															
Reserved															
Reserved															

Figure 6-4. Master Control/Status Block (MCSB)

Master Status Register (MSR)

The controller uses this register to report board level status. From the host point of reference, this is a READ ONLY register. However, the contents of this register are not valid for 100 microseconds following the GOTO MACSI MODE command. The bits are defined as follows:

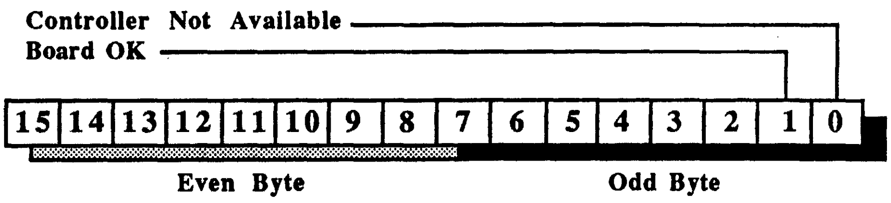


Figure 6-5. Master Status Register (MSR)

Bit 0 (CNA) Controller Not Available:

The controller sets this bit to a '1' to indicate that it is Not Available to receive a command. This bit will only be set during a GOTO MACSI MODE command. The controller will reset this bit to a '0' when the GOTO MACSI MODE command is complete.

Bit 1 (BOK) Board O.K.:

The controller sets this bit to a '1'when the GOTO MACSI MODE command has completed successfully. A '0' indicates that the controller detected a failure during operation.

Bits 2-15 (RSRV) Reserved:

RESERVED and are set to '0' by the controller.

Master Control Register (MCR)

All bits in this register are both set and reset by the host. From the controller's point of reference, this is a READ ONLY register. The controller will never set any of these bits. The bits are defined as follows:

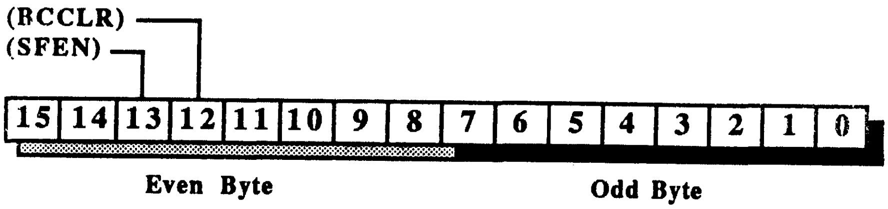


Figure 6-6. Master Control Register (MCR)

Bits 0-11:

RESERVED and should be set to '0' by the host.

Bit 12 (BDCLR):

Setting Bit 12 of the Master Status Register will cause a hardware reset to the controller.

Bit 13 (SFEN):

If '0', and BOK is also '0' (inactive), this bit enables BOK to cause SYSFAIL to be asserted on the VMEbus. If '1', this bit causes the SYSFAIL indication to be disabled. The controller never changes this bit.

Bit 14-15:

RESERVED and is set to '0' by the controller.

Interrupt on Queue Available Register (IQAR)

The Command Queue is a circular queue with slots for numerous Command Queue Entries (CQE). Each CQE occupies a slot in the Command Queue for a short period of time (until it is moved into a Work Queue), thus the host will virtually always have slots available in the Command Queue for issuing commands. In the unlikely event that the Command Queue is full when the host attempts to enter a command, the host must wait until the controller transfers a command from the Command Queue to an internal Work Queue before it can enter the next command.

The host determines that the Command Queue is full by looking at the Go bit in the next available Command Queue Entry. The Command Queue is full if the Go bit of the next available Command Queue Entry is '1'. If the Command Queue becomes full, the host could simply poll the Go bit, waiting until the next Command Queue Entry becomes available. But the controller, through the Interrupt on Queue Available Register, provides for efficient operation by optionally interrupting the host when a slot becomes available in the Command Queue.

The Interrupt on Queue Available is enabled by setting the Interrupt on Command Queue Available bit in the Interrupt on Queue Available Register. The interrupt will occur as soon as the controller detects at least one empty slot in the Command Queue. The host should wait until encountering the Queue Full condition before setting the IQEA bit. Once the IQEA bit is set, the controller generates an interrupt as soon as the necessary queue conditions are satisfied. Once it generates the interrupt, the controller resets the IQEA bit.

The level and vector for the IQEA interrupt are supplied by the host in the Interrupt on Queue Available Register (IQAR). When the necessary queue conditions are satisfied, the controller clears the IQEA bit and generates a Command Complete Interrupt with the Queue Entry Available (CQA) bit set in the Command Response Status Word (CRSW) of the Command Response Block (CRB). Even though the controller provides for efficient operation by providing this mechanism for interrupting the host when a slot (or slots) become available in the Command Queue, it is preferable for the number of slots in the Command Queue to be set during initialization so that the full condition occurs infrequently.

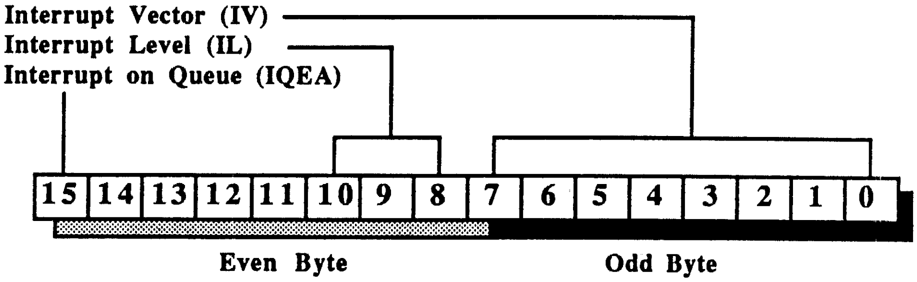


Figure 6-7. Interrupt on Queue Available Register (IQAR)

Bits 0-7 (IV) Interrupt Vector for the Interrupt on Queue Available:

The controller uses this byte as the interrupt vector when issuing an Interrupt on Queue Available interrupt. This byte is set by the host and is not modified by the controller. The host must not modify this byte after setting the Interrupt on Queue Entry Available bit.

Bits 8-10 (IL) Interrupt Level for the Interrupt on Queue Available:

These three bits determine the interrupt level that the controller will use when issuing an Interrupt on Queue Available interrupt. These bits are set by the host and are not modified by the controller. The host must not modify these bits after setting the Interrupt on Queue Entry Available bit.

Values of zero through seven are allowed. An interrupt level of zero is allowed only when the IQEA bit is reset.

Bits 11-14 (RSRV) Reserved:

These bits must be set to '0' by the host.

Bit 15 (IQEA) Interrupt on Queue Entry Available:

This bit is set by the host to request an Interrupt on Queue Entry Available. The interrupt is generated either when the queue has at least one slot available. The controller resets this bit prior to generating the Interrupt on Queue Available interrupt. After the host sets this bit, requesting an interrupt, it cannot change any of the other bits in the Interrupt on Queue Available Register.

Queue Head Pointer

The Command Queue is a circular queue and the controller requires that the host use the slots in the Command Queue in order. The Queue Head Pointer provides a convenient method for the host to control the ordering of and the access to the Command Queue. The Queue Head Pointer register provides a place for the host to store the address of the next available slot in the Command Queue.

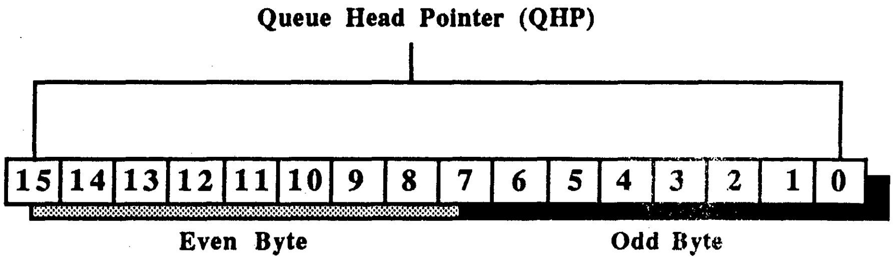


Figure 6- 8. Queue Head Pointer/Queue Head Pointer In Use

Bits 0-15 (Address) Queue Head Pointer:

This is the address of the next slot in the Command Queue. Since the controller transfers commands out of the Command Queue in circular order as soon as it sees the Go bit set, this address is not actually used by the controller.

The MACSI interface relies on the chronological order of the Command Queue. The host must some how ensure that the slots in the Command Queue are used in chronological order.

Command Queue

The Command Queue (CQ) consists of a programmable number of slots for Command Queue Entries (CQE). The number of slots for CQEs in the Command Queue is set by the host when it initializes the controller. The Command Queue is circular, but it is up to the host to keep track of the next Command Queue Entry that it can use. Because the queue is circular, the controller can infer chronological ordering of commands. The actual size of the Command Queue equals the number of slots times 12 bytes. The controller Command Queue can have a maximum of 8 slots.

Command Queue Entry (CQE)

Each Command Queue Entry (CQE) includes all of the information that is needed for the host to find, execute, and respond to the commands contained in an IOPB. Each slot in the Command Queue is "busy" only until the controller can transfer the command to a Work Queue and then free the slot in the Command Queue. This mechanism allows a large number of commands to be queued up without needing an excessive number of slots in the Command Queue itself. Each Command Queue Entry consists of a Queue Entry Control Register, an IOPB Address, a Command Tag, and some reserved bytes. Each entry is two bytes long, except the Command Tag which is four bytes long; thus, there are 12 bytes in each CQE.

<table>
<tr><th colspan="8">Even Byte</th><th colspan="8">Odd byte</th></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">Queue Entry Control Register</td></tr>
<tr><td colspan="16">IOPB Address</td></tr>
<tr><td colspan="16">Command Tag</td></tr>
<tr><td colspan="16">Command Tag (4 words long)</td></tr>
<tr><td colspan="8">IOPB Length</td><td colspan="8"></td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

Figure 6-9. Command Queue Entry

Queue Entry Control Register (QECR)

The Queue Entry Control Register (QECR) is used to kick off command execution, to sort disk commands into a work queue, and to flag high priority commands. The Queue Entry Control Register is two bytes long.

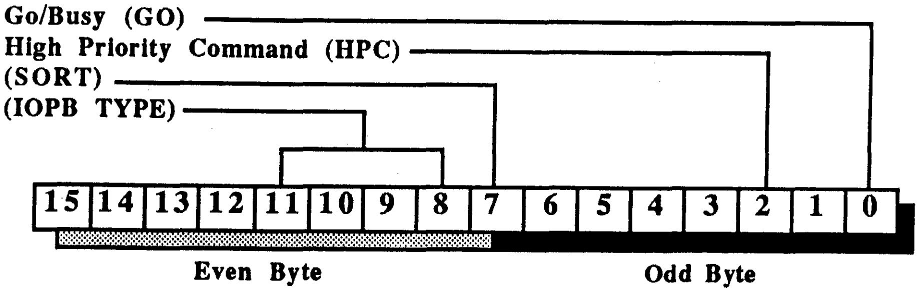


Figure 6-10. Queue Entry Control Register (QECR)

Bit 0 (GO) Go/Busy:

The Go/Busy bit is set by the host to initiate action on a CQE. The host must assemble the IOPB in the Host Usable Space (HUS) and assemble the entire CQE in the Command Queue (CQ) before it sets this bit. The controller moves the CQE and the IOPB into internal memory as soon as it sees the Go/Busy bit set. Once the IOPB and CQE are in internal memory, the controller will reset the Go/Busy bit to free the slot in the Command Queue.

Bit 1:

RESERVED and not used by the controller.

Bit 2 (HPC) High Priority Command:

The High Priority Command (HPC) bit flags a command so that the controller places the command at the top of its Work Queue. If there are already other commands in the work queue with the HPC bit set, the new command is queued up directly behind the other High Priority Commands. (Thus, there is a FIFO-type ordering of High Priority Commands.)

Bits 3-6 (RSRV) Reserved:

These five bits are reserved and must be set to '0' by the host.

Bit 7 (SORT):

Setting this bit will cause the controller to sort the incoming disk command into its internal work queue.

Bits 8-11 (IOPB TYPE):

These four bits describe the IOPB type. IOPB type 1 is for DISK commands. IOPB type 2 is for SCSI commands.

Bits 12-15 (RSRV) Reserved:

These four bits are reserved and must be set to '0' by the host.

IOPB Address

The IOPB Address is a 2-byte pointer to the IOPB associated with the Command Queue Entry. It is the offset of the IOPB from the first location of the dual ported memory. The IOPB must reside in the Short I/O space. Since the controller transfers both the CQE and the IOPB into internal memory as soon as it sees the Go/Busy bit set, this section of Short I/O is only "tied up" for a very short time.

Command Tag

The Command Tag is a 4-byte value that the host can use to keep track of individual commands. The controller does not use the Command Tag, nor does it modify it. It simply returns the Command Tag as part of the Command Response. Thus, in a typical implementation, the host would use a unique value Command Tag for each command so that it can always differentiate one command from another. The format of the Command Tag is determined solely by the host.

IOPB Length

IOPB Length byte specifies the length of the IOPB in long words. The maximum IOPB length is 32 bytes. A recommended value is 8.

Host Usable Space (HUS)

The Host Usable Space (HUS) is free form memory space accessible to both the host and the controller. No partitioning of the Host Usable Space (HUS) is implied or required by the controller MACSI interface. The manner in which it is used is totally under the control of the host. Typically, the Host Usable Space in the controller is used to pass the IOPB portion of a command. And in some multiprocessing applications, the HUS is a handy place to post semaphores between CPUs.

The amount of Host Usable Space available is determined by two factors: the number of Command Queue slots defined when the Command Queue (CQ) is initialized, and by the length of the Command Response Block that is defined. For example, if the Command Queue is initialized with eight slots (each CQE slot is 12 bytes long) and a Command Response Block of 56 bytes is defined, there will be 300 bytes of Host Usable Space available. The Master Control/Status Block, and the Device Status Block always occupy a total of 48 bytes.

Command Response Block (CRB)

The Command Response Block (CRB) is used by the controller to post completion status and to post data that results from a command completion or from a queue slot available condition. The Command Response Block is made up of the Command Response Status Word (CRSW), the Command Tag, and the Returned IOPB. If the posting is the result of an IOPB completion, the IOPB itself and related status information are returned here.

The offset of the Command Response Block is defined at initialize time. The length of the Command Response Block can be determined by subtracting the Command Response Block offset from the offset of the Device Status Block (1E0H). However, the length of the Command Response Block must be equal to the largest IOPB defined plus 12 bytes. A recommended minimum value is 44 bytes

Even								Odd							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Command Response Status Word															
IOPB Type															
Command Tag															
Command Tag (4 Words Long)															
IOPB Length															
RESERVED															
RETURN IOPB															

Figure 6-11. Command Response Block (CRB)

Command Response Status Word (CRSW)

The Command Response Status Word (CRSW) is the first word in the Command Response Block (CRB). It describes the nature of the command response. It also contains a handshake bit, the Command Response Block Valid/Clear Interrupt (CRBV) bit. (The CRBV synchronizes the command interaction of the controller and the host.) The bits of the Command Response Status Word (CRSW) are defined as follows:

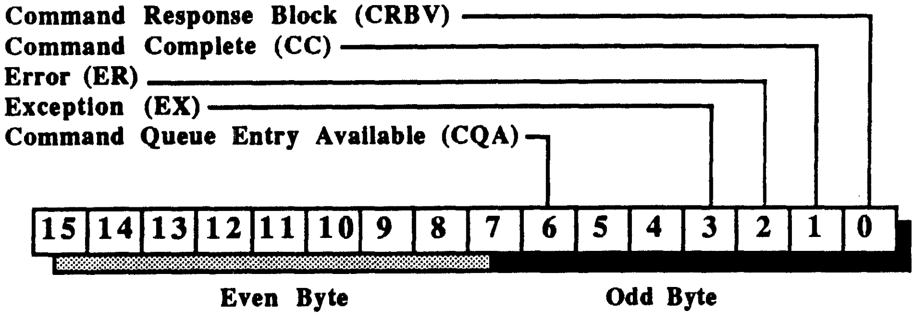


Figure 6-12. Command Response Status Word (CRSW)

Bit 0 (CRBV) Command Response Block Valid/Clear Interrupt:

The Command Response Block Valid/Clear Interrupt (CRBV) bit is set by the controller after it loads the returned IOPB and the Command Tag into the Command Response Block (CRB).

The Command Response Block Valid/Clear Interrupt (CRBV) bit is also an Interrupt Pending bit because the controller sets it immediately prior to issuing a Command Complete interrupt to the host. The controller keeps the CRBV stable while this bit is set.

After it is finished using the information in the CRB, the host clears the interrupt by clearing the Command Response Block Valid/Clear Interrupt (CRBV) bit. Once this bit is reset, the controller can use the Command Response Block to present the next command response.

Bit 1 (CC) Command Complete:

The Command Complete (CC) bit is set by the controller when the Command Response Block (CRB) is being used to post the response to a Command Completion, as opposed to a Queue Entry Available condition or a Start Queued Mode operation. The Command Complete (CC) bit is set even when the command is completed with error or exception. (See Command Response Status Word (CRSW) bits 2 and 3.)

Bit 2 (ER) Error:

The Error (ER) bit is set by the controller when the Command Response Block (CRB) is being used to post a Command Completion with an Error. The host can determine the exact nature of the error by examining the returned IOPB. The error bit is valid only when Command Complete is active.

Bit 3 (EX) Exception:

The Exception (EX) bit is set by the controller when the Command Response Block (CRB) is being used to post a Command Completion with an Exception. The host can determine the exact nature of the exception by examining the returned IOPB. The exception bit is valid only when command complete is active.

Bits 4-5 Reserved:

These bits are reserved and are set to '0' by the controller.

Bit 6 (CQA) Command Queue Entry Available:

The Command Queue Entry Available (CQA) bit is set by the controller when the Command Response Block (CRB) is presented in response to a queue entry available condition.

The Command Queue Entry Available (CQA) bit is mutually exclusive with the Command Complete (CC) bit.

Bits 7-15 Reserved:

These bit is reserved and is set to '0' by the controller.

IOPB TYPE

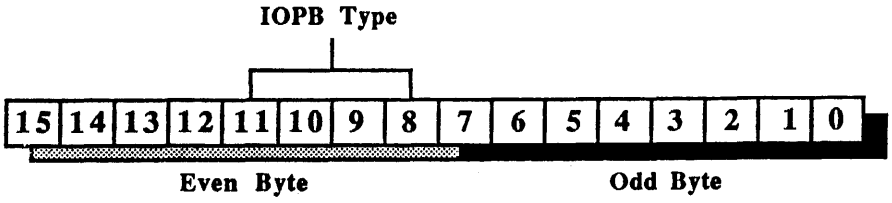


Figure 6-13. IOPB Type

Bits 0-7 Reserved:

RESERVED and are set to '0' by the controller.

Bits 8-11 (IOPB TYPE):

These 4 bits describe the IOPB type. IOPB type 1 is for DISK commands. IOPB type 2 is for SCSI commands.

Bits 12-15 Reserved:

RESERVED and are set to '0' by the controller.

Command Tag

This Command Tag is the same 4-byte value that was provided in the Command Queue Entry (CQE) when this command was originally issued to the controller. The controller does not use the Command Tag, nor does it modify it. It simply returns the Command Tag as part of the Command Response. The Command Tag is used by the host to determine which command the controller is responding to.

IOPB Length

The upper byte specifies the length (in long words) of the returned IOPB. The IOPB Length word is returned from the Command Queue Entry (CQE) exactly as it was originally entered by the host. It is not modified by the controller.

Returned IOPB

The Returned IOPB field of the Command Response Block (CRB) is usually an image of the IOPB modified to reflect command completion status.

This returned IOPB area is undefined for a response to a Queue Entry Available condition or for any other command response where the original command did not require an IOPB.

Device Status Block (DSB)

The controller uses the Device Status Block to post various unit statuses.

Table 6-1. Device Status Block

Hex Offset	Even								Odd							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1E0	Tape Drive 0 Status Register															
1E2	Tape Drive 1 Status Register															
1E4	Tape Drive 2 Status Register															
1E6	Tape Drive 3 Status Register															
1E8	Tape Drive 4 Status Register															
1EA	Tape Drive 5 Status Register															
1EC	Tape Drive 6 Status Register															
1EE	RESERVED															
1F0	RESERVED (010 bytes)															
1FA	Disk Drive 3 Status (4 unit)								URDY	PRES	ER	CC	FLT	ATT	WP	DRDY
1FC	Disk Drive 1 Status (4 unit)								Disk Drive 0 Status (4 unit)							
1FE																SCSI

Tape Status Register

There are two types of registers used for monitoring the status of the tape drives: 7 Tape Status Registers, one for each possible tape drive, and one Status Change Register that is used to identify the tape unit that has undergone a status change. Tape Status Registers are maintained by the V/ESDI 4201 and can be read by the host at any time. If more than one drive has a status change, the tape identifiers are written to the Status Change Register in FIFO order after each interrupt is serviced.

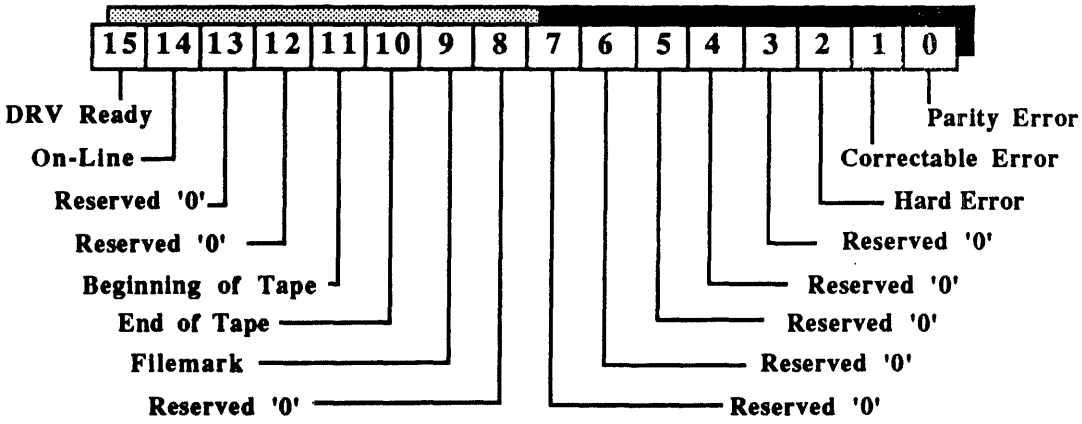


Figure 6-14. Format of the Tape Status Register

Bit 15 Drive Ready:

Unit is loaded and ready.

Bit 14 On-Line:

Drive is on-line

Bit 13 Reserved:

This bit must be '0'.

Bit 12 reserved:

This bit must be '0'.

Bit 11 BOT:

Beginning of tape (Load Point) reported on the tape.

Bit 10 EOT:

End of tape was reported by the drive.

Bit 9 Filemark:

A filemark was detected on the tape.

Bit 8 Reserved:

This bit must be '0'.

Bit 7 Reserved:

This bit must be ‘0’.

Bit 6 Reserved:

This bit must be ‘0’.

Bit 5 Reserved:

This bit must be ‘0’.

Bit 4 Reserved:

This bit must be ‘0’.

Bit 3 Reserved:

This bit must be ‘0’.
Bit 2 Hard Error:

A hard error (uncorrectable) was reported by the formatter.

Bit 1 Correctable Error:

A data error was corrected by the formatter.

Bit 0 Parity Error:

Parity error was detected by V/ESDI 4201 on a data read from the tape.

The Disk Drive Status Register is identical in operation, location, and format in the Non-MACSI Mode discussed in Chapter 4.

Board Status Bits Register

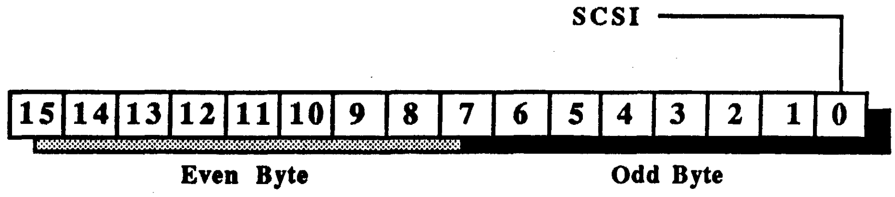


Figure 6-15. Board Status Bits Register

Bit 0: (SCSI) SCSI Card Present

The controller sets this bit to a '1' to indicate that the optional SCSI daughter card is attached.

Bit 1-15 Reserved:

RESERVED and must be set to '0' by the host.

VMEbus Interrupts

At the completion of a command (either successful or terminated with an error or with an exception), the controller notifies the host by generating a Command Complete Interrupt on the VMEbus. The controller can respond to the VMEbus Interrupt Acknowledge Cycle with different Interrupt Vectors based on the cause of the interrupt. But for those VMEbus systems that only allow one interrupt vector per device, the host can still determine the source of the interrupt by checking the status bits in the Command Response Status Word found in the Command Response Block.

The hardware driving the VMEbus interrupt line is cleared at the completion of the VMEbus Interrupt Acknowledge Cycle. However, a Clear Interrupt operation must also be executed by the host to notify the controller that the interrupt has been properly serviced by the host and that the controller may now post its next interrupt. The host does this by clearing the CRBV bit in the Command Response Block. (See the section on the Command Response Status Word (CRSW) of the Command Response Block.)

NOTE

Any information that the host needs from the Command Response Block must be accessed before it executes this Clear Interrupt operation.

I/O Parameter Block and Commands

The MACSI software interface of the controller can handle two distinct types of IOPBs: DISK IOPBs and SCSI IOPBs. SCSI IOPBs are commands that are sent to the attached SCSI devices. Disk IOPBs are commands that are sent to the attached DISK devices.

Disk IOPBs

The format of a DISK IOPB is identical to the format used during 4 unit mode operation. The DISK IOPB type is 1.

SCSI IOPBs

The host uses SCSI IOPBs to send commands to specific SCSI peripherals attached to the controller. The controller uses these commands to supervise SCSI activity and to transfer data to or from the VMEbus. The controller translates the incomming TAPE command into a SCSI command and sends it to the device for which it is intended.

MACSI Mode Error Codes

All error codes are in hexadecimal format.

E0 - Already in MACSI mode:

The MACSI MODE command has been issued to the controller when the controller is already in MACSI mode.

E1 - Bad CIB address:

The address given for the controller initialzation block is invalid. This address cannot be 0.

E3 - Number of CQEs is bad:

The number of CQEs must be greater than 0 and less than 9.

E5 - No SCSI daughter Card is attached:

The desired command can only be run when a SCSI daughter card is attached.

E7 - MACSI mode is required:

The desired command can only be run in MACSI mode.

E8 - IOPB size is invalid:

The IOPB size entered in the CQE must be larger than 0.

E9 - Number of Commands to Q is invalid:

The number of commands to Q given in the CIB must be greater than 0 and less than 65.

THIS PAGE LEFT BLANK INTENTIONALLY

APPENDIX A SPECIFICATIONS OF THE V/ESDI 4201

VMEbus SPECIFICATONS	
DTB Master	A32,D16,D32 DMA Transfers
DTB Slave	A16,D8,D16 (Commands & Status)
Requester	Any of R0- - R3 (Static)
Interrupter	Any of I1- - I7 (Dynamic)
Peripheral Data Rate	Up to 24 Mbits/sec
ENVIRONMENTAL SPECIFICATIONS	
Operating Temperature	32-131 deg. F. (0-55 deg. C)
Maximum Relative Humidity	10-90% noncondensing
ELECTRICAL SPECIFICATIONS	
Power	4.5 A nax @ +5V DC (S5%) 0.5 A max @ -12V DC (S5%)
MECHANICAL SPECIFICATIONS	
Lenght	9.20" (233 mm)
Width	6.30 (160 mm)
Thickness	0.77 (19.6 mm)
Weight	1.01 lb. (0.45 kg)

The V/ESDI 4201 uses industry standard "A" cable (daisy chain), and "B" cable (radial) to interface with the drives. Both the "A" and "B" cables may be up to three meters long. The V/ESDI 4201 is compatible with all known ESDI drives, some manufacturers of which are listed below.

HITACHI	MAXTOR	MICROPOLIS	SIEMENS
FUJITSU	CDC	NEC	

THIS PAGE LEFT BLANK INTENTIONALLY

APPENDIX B
CABLE INTERFACES

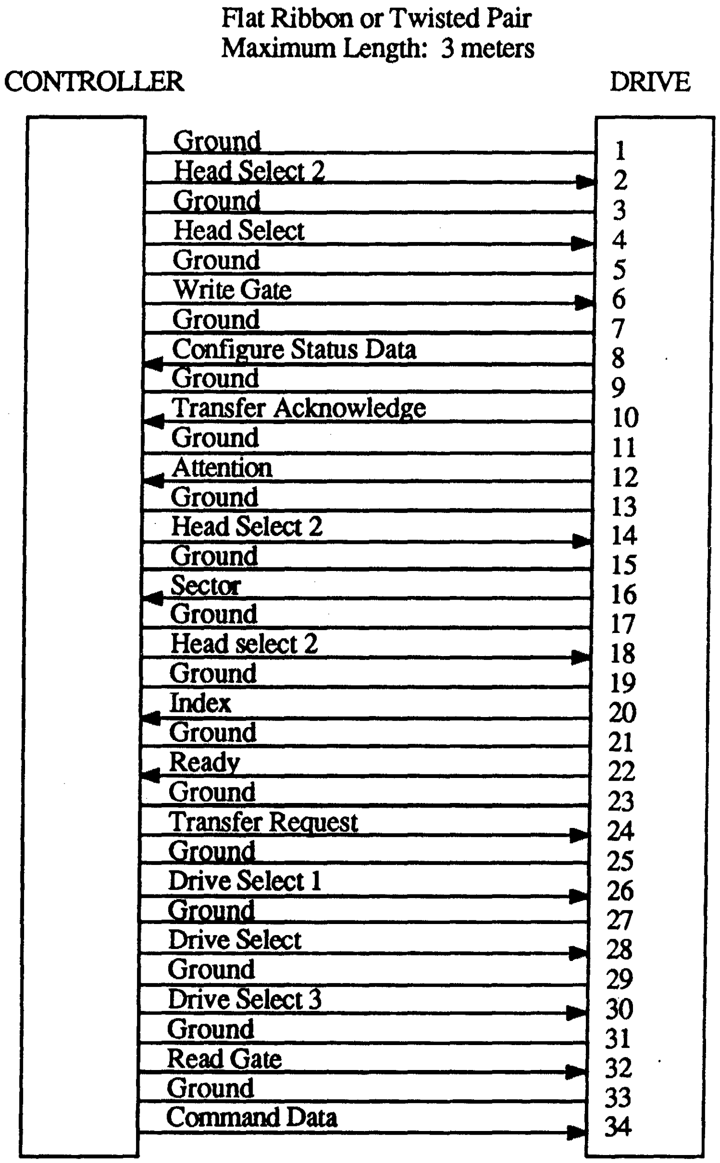
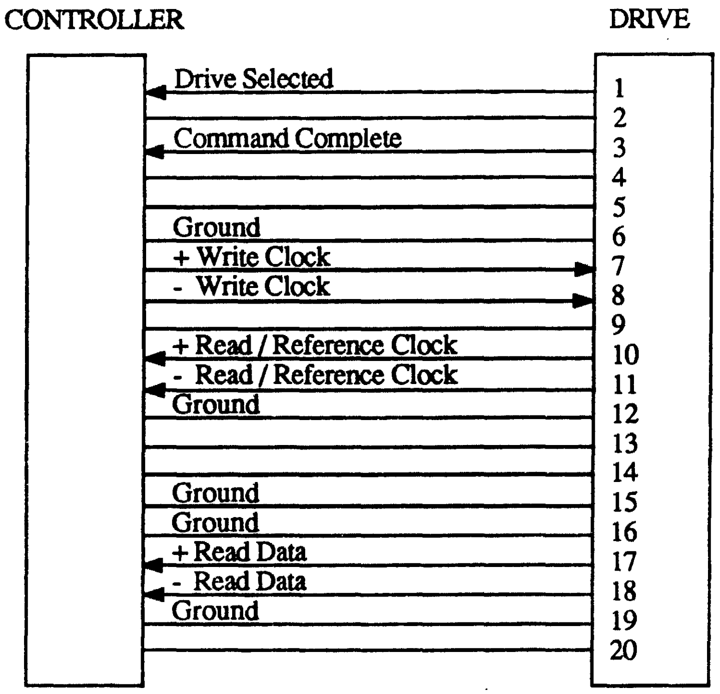


Figure B-1. "A" Cable Interface



Flat Ribbon or Twisted Pair
Maximum Length: 3 meters

Figure B-2. "B" Cable Interface

APPENDIX C COMMAND CODES

CODE	COMMAND	CODE	COMMAND
70	DIAGNOSTICS	89	RESTORE (RETURN TO 0)
71	READ LONG	8B	REFORMAT
72	WRITE LONG	8C	FORMAT TRACK WITH DATA
74	READ HEADER	90	MAP SECTOR
75	READ ESDI FLAW MAP	91	READ SECTORS SEQUENTIAL
76	READ CDC ESDI FLAW MAP	92	WRITE SECTORS SEQUENTIAL
77	REPORT CONFIGURATION	93	VERIFY SECTORS SEQUENTIAL
78	WRITE SECTOR BUFFER	94	READ NONCACHED
79	READ SECTOR BUFFER	95	READ SEQ., DISABLE ADDR. BUMP
7A	REPORT SECTOR ID	96	WRITE SEQ., DISS ADDR. BUMP
7B	FORMAT WITH SECTOR ID	97	CLEAR DRIVE FAULT
7C	INITIALIZE LONG	99	VERIFY TRACK
7D	REPORT CONFIG. LONG	9A	TRACK ID
7E	READ FLAW MAP HEADER	9B	FETCH AND EXECUTE IOPB
7F	GOTO MACSI	9C	VERIFY TRACK SEQUENTIAL
80	WRITE AFTER CACHE	9E	EXTENDED DIAGNOSTICS
81	READ SECTOR(S)	A0	ESDI PASS-THROUGH
82	WRITE SECTOR(S)	A1	READ AND SCATTER
83	VERIFY SECTOR(S)	A2	GATHER AND WRITE
84	FORMAT TRACK	A3	READ AND SCATTER 32
85	MAP TRACK	A4	GATHER AND WRITE 32
86	HANDSHAKE	A5	READ AND SCATTER 32 (LIST 2)
87	INITIALIZE	A6	GATHER AND WRITE 32 (LIST 2)

70 Diagnostics

This command causes the power-up diagnostics to be executed within the V/ESDI 4201. Diagnostic tests include checksum on the EPROMS, memory test on the RAM buffer, handshaking with the disk interface hardware, and operation of the buffer control hardware. Any errors will be reported as command completed with error (command status code 82 plus the appropriate error code). For a complete list of the error codes, refer to pages Appendix D of this User's Guide.

71 Read Long

This is a diagnostic tool used to read and transfer the data field and the four ECC bytes of a disk sector. Error detection is disabled during a Read Long command.

72 Write Long

The Write Long command is the reverse of the Read Long command. This is a diagnostic tool used to write the data field and the four bytes of the ECC to the appropriate sector on the disk. Error detection is disabled during a Write Long command.

74 Read Header

This command causes the drive to read the first header field that it encounters. If logical translation is not enabled, the cylinder number is returned in IOPB Word two, the head number is returned in the upper byte of IOPB Word three, and the sector that was encountered is returned in the lower byte of Word three. Since the command only reads the first header field that it encounters, the sector number will vary from command to command on the same track. If logical translation is enabled, the logical value of the first sector encountered is placed in IOPB Words two and three with Word two as the most significant word.

75 Read ESDI Flaw Map

This command causes the drive to read the ESDI flaw map located in the first sector of the specified track. The amount of data toi be transferred is specified in the UIB. The IOPB must be set up with the appropriate cylinder and head address for the flaw map information. Also, the user must verify the integrity of the data once it has been transferred. This can be done either by appending a CRC onto the end of the data field or by comparing the data with that present on the secondary flaw map track.

76 Read CDC ESDI Flaw Map

This command is functionally the same as the Read ESDI Flaw Map command (75) with the exception that it has a built-in delay of 16 (10h) bytes from the index before activating "rdgate." Like the Read ESDI Flaw Map command, the IOPB must be set up with the appropriate cylinder and head address for the flaw map information. Also, the user must verify the integrity of the data once it has been transferred. This can be done either by appending a CRC onto the end of the data field or by comparing the data with that present on the secondary flaw map track.

77 Report Configuration

This command causes the operating parameters (UIB) of the specified unit to be returned. The data is written to memory and pointed to by the buffer address pointer (Words five and six), the memory type, and the address modifier (Word seven) in the IOPB. The UIB is returned in the exact format in which it was written.

78 Write Sector Buffer

This diagnostic command transfers one sector of data from system memory to the on-board memory of the V/ESDI 4201. It does not transfer the data to the disk. The purpose of this command is to check the handshaking over the VMEbus. It is used in conjunction with the Read Sector Buffer command.

79 Read Sector Buffer

This is a diagnostic command that transfers the equivalent of one sector of data from an on-board buffer to system memory. The purpose of this command is to allow a VMEbus transfer check. It is used in conjunction with the Write Sector Buffer command. The buffer that is loaded with the Write Sector Buffer command is transferred to system memory and no data is read from the disk. If the Read Sector Buffer command is not preceded by a Write Sector Buffer command, an arbitrary sector buffer can be transferred and a status code of 83 will result.

7A Report Sector ID

This command causes the head to read the sector number starting at index. It returns a table (in external memory only) that describes how the sectors are laid out on the disk. These sector entries can be rearranged such that the bad sectors are mapped out. For example, if 20 sectors are specified in the UIB, a bad sector can be moved to position 21 in the table, so that it will never be encountered during normal operation. Once the new table is compiled, a Format With Sector ID command (7B) can be run. (See the "Implementing" section at the end of this document for details of building the table.) Only memory type two or three (external memory) may be specified in this command.

7B Format With Sector ID

This command is used in conjunction with the Report Sector ID command (7A) to format a track when sectors have to be mapped (via command 7A). No data is saved when this command is issued. Only memory type two or three (external memory) may be specified in this command. A normal Format command (84) can be used if there are no sectors to be slipped. All of the extra sectors will be at the end of a track when using the Format (84) command.

7C Initialize Long

This command must be issued to Initialize a unit before multiple sector slipping can be used. This is to accommodate the expanded UIB (from nine to 16 words) that is necessary for the multiple sector slipping procedure. The Initialize Long command functions just like the Initialize command (87) with the exception that 16 words of UIB are also read. Memory type one, two or three may be specified in this command.

7D Report Configuration Long

This command works just like the Report Configuration command (77) with the exception that 16 words of UIB are reported (as opposed to the standard nine words). The UIB is returned in the exact format in which it was written. Memory type 1, 2 or 3 may be specified in this command.

7E Read Flaw Map Header

This command causes a raw copy of the header located at index to be returned in the IOPB STARTING AT WORD TWO.

7F GoTo MACSI

Refer to Chapter 6 for a complete description.

80 Write After Cache

Functionally, this command is the same as the Write Sector(s) command (82). However, the write operation is not performed until after the current caching is complete.

81 Read Sector(s)

This command reads one or more sectors of data from the specified disk (or from cache memory if the data has been cached) and transfers the data to system memory. The starting sector is specified in Words two and three of the IOPB. If the logical translation mode option was specified in the lower byte of the command word, a logical-to-physical translation is done on the specified disk address. The number of sectors transferred is determined by the sector count in Word five of the IOPB. The data is transferred to system memory starting at the address given in the buffer address words of the IOPB (Words five and six). Data is transfered while the V/ESDI 4201 is acting as a Bus Master, thus relieving the host CPU of the burden of performing the transfers.

82 Write Sector(s)

This command is the reverse of Read Sector(s). Data is not cached on write operations.

83 Verify Sector(s)

This command is used to verify that one or more logically sequential sectors of previously written data contains no ECC errors. No data is transferred to the VMEbus during this command. Data is always read from the disk during this command, and the cache is not used. This command does not check the spare sector (if one exists) unless it has already been mapped. The sector or sectors of data to be verified are specified exactly like the Read Sector(s) command.

84 Format Track

This command is used to format a given track with the proper header information. Formatting records header information for each sector on the track to allow the V/ESDI 4201 to verify head position prior to a read or write. All disk surfaces must be formatted before they can be read from or written to. The first two words in the data field of each sector will contain the cylinder number (in the first word), the head number (in the MSB of the second word), and the sector number (in the LSB of the second word). The rest of the sector is filled with the data supplied in Word six of the IOPB. The track to be formatted is specified in Words two and three of the IOPB.

85 Map Track

This command is used to map a bad track on any disk surface to any available spare track on the disk surface. The bad track and the spare track are identified in the IOPB. The track to be mapped is identified in Word two and the upper byte of Word three of the IOPB (cylinder address and head address in physical mode). The alternate (spare) track is specified in Words four and five, with word four as the alternate (spare) cylinder address, and the upper byte of Word five as the alternate (spare) head address. The bad track is formatted with a special header field that allows the V/ESDI 4201 to recognize it as a bad track. This field also tells the V/ESDI 4201 where the spare track is located. When the bad track is encountered during a normal operation, the V/ESDI 4201 recognizes the special format and automatically seeks the specified spare track. This operation is transparent to the user, except for the extra seek time required to get to the alternate (spare) track.

86 Handshake

This command is a diagnostic tool used to verify controller operation. When executed, this command returns product identification information in Words two through seven (Bytes 4 - D, hex) of the IOPB. Each byte is defined below:

Byte	Contents	Byte	Contents
4	Product Code - 2SD and LSD(h)	5	Product Variation (ASCII)
6	Reserved	7	Rev. Level - MSD (ASCII)
8	Rev. Level -2SD (ASCII)	9	Rev. Level - 3SD (ASCII)
A	Month - MSD & LSD(h)	B	Day - MSD & LSD(h)
C	Year -MSD & 3SD(h)	D	Year - 2SD & LSD(h)

Product code =074H, MSD =most significant digit, LSD =least significant digit, 2SD =second most significant digit, and 3SD =third most significant digit.

87 Initialize

This command is used to initialize the V/ESDI 4201 with parameters that identify the type of disk drive being initialized. One Initialize command is needed for each drive, so drives can be mixed regardless of their type. The Initialize command tells the V/ESDI 4201 all of the information necessary to run the drive, including the drive-dependent options as specified in the Unit Initialization Block (UIB). This command can be issued at any time to change the operating characteristics of a drive. This command must be issued following each power-up operation and each hardware reset.

89 Restore

This command causes the specified drive to be recalibrated and any faults to be reset. The Restore command seeks the drive to cylinder zero by issuing either a recalibrate or seek home command to the drive. It then restores any fault conditions by issuing a Clear Fault to the drive. Restore waits for recalibrate to finish and the fault to be cleared before the completion interrupt is generated. If the recalibrate fails, a command complete with error condition will result after a timeout. The error code will be 1C, 1D, 1F or 30.

8A Seek

The Seek command causes the specified drive to move its read/write heads to the specified cylinder and then select the specified head. This command is primarily used for overlapped seeks, since commands such as Read Sector(s) or Write Sector(s) have implied seeks. The seek command generates a command complete interrupt as soon as it is done (i.e., as soon as the seek has started). Thus, the V/ESDI 4201 does not wait for the seek to complete before allowing execution of the next command. The V/ESDI 4201 can be programmed, via the UIB, to generate a disk status change interrupt when a drive completes a seek.

8B Reformat

This command formats a track exactly like the Format command, unless the specified track was previously mapped as bad, or if bad sector mapping has been performed on the track. If the specified track is mapped bad, then the alternate (spare) track is formatted for the bad track, and the bad track is left unchanged. If bad sector mapping was performed, the track will be reformatted with the same sector mapped as bad. This command is used to format without having to worry about remapping tracks that were previously mapped bad. Skew and interleave are calculated as with the normal format.

8C Format Track With Data

This command is used for surface analysis of the disk. Skew and interleave are calculated as with the normal format. It is similar to the Format Track command except that the user can specify a block of data to be written to each sector on the track. Each sector of data that is to be written is pointed to by the buffer address pointer (IOPB Words five and six), the memory type, and the address modifier (IOPB Word seven). The same data is written to each sector, so the sector count in the IOPB is not required when using this command. The number of sectors is specified by the Sectors/Track byte in the UIB.

90 Map Sector

The Map Sector command identifies and maps a sector on a given track that is bad. This command causes the track to be reformatted, and the bad sector to be formatted with a special header field. All sectors following the bad sector are "slipped" out one-by-one towards the end of the track. This allows the spare sector to be used during a Read or Write without requiring extra latency. Skew and interleave are calculated as with normal formatting. To use this command, the "spare sector enable" bit must be set in the UIB. It should also be noted that only one sector per track can be mapped using this command. If a larger area must be mapped, the Map Track command must be used.

91 Read Sector(s) Sequential

This command is similar to the normal read command except the zero latency feature is turned off. This forces the drive to begin reading data at the beginning of the sector specified by the IOPB. This command is usually used in situations that require data to be read in a specific order.

92 Write Sector(s) Sequential

This is the reverse of Read Sector(s) Sequential. Data is not cached on write operations.

93 Verify Sector(s) Sequential

This command is similar to the normal Verify Sector(s) command except the zero latency feature is turned off. This forces the drive to begin verification starting at the beginning of the sector specified in the IOPB. No data is transferred to the VMEbus during this command.

94 Read Noncached

This command executes in the same manner as the Read Sector(s) command, except that the cache buffer is ignored. The data will be read from the disk drive regardless of whether or not it is already contained in the cache. This command is useful for verifying that the data has been correctly written to the disk.

95 Read Sequential, Disable Address Bump

This command is the same as the Read Sequential command except an address disabling feature has also been added. When this command is used, the bus address is not incremented on the VMEbus; therefore, data is transferred to a single address instead of a block of memory.

96 Write Sequential, Disable Address Bump

This command is the reverse of the Read Sequential, Disable Address Bump command.

97 Clear Drive Fault

This command issues a Clear Fault to the specified drive. No head movements are initiated. Clear Drive Fault waits for any fault conditions to be cleared before the completion interrupt is generated. If the fault condition fails to clear, the interrupt is generated after a timeout. If this happens, the Error Code byte and Drive Status byte indicate the remaining fault conditions.

99 Verify Track

This command is used to verify that all of the sectors on a track, except the spare sector (unless already mapped), are readable and contain no ECC errors. No data is transferred to the VMEbus during execution of this command. If this track has been mapped bad, the alternate (spare) track is verified. Data is always read from the disk during execution of this command, and the cache is not used. The track to be verified is specified in Words two and three of the IOPB.

9A Track ID

This command causes the head to read all of the header fields on a given track and return them in the order in which they were encountered starting from the index pulse. This command is useful in determining the format of a particular track (interleave and skew). The track number is specified in Word two of the IOPB, in physical mode only. The headers will be transferred to system memory starting at the buffer address specified in Words five and six of the IOPB. Four words are transferred for each header that is found. The first word contains the sync field and the cylinder number, and the second word contains the head and sector number (upper and lower byte, respectively). The third word contains duplicate head and sector information for error detection, and the fourth word contains all zeros. The total number of words transferred is equal to four times the number of sectors per track specified in the UIB. All headers returned are physical addresses.

9B Fetch And Execute IOPB

This command is used to allow external IOPBs to direct the activities of the V/ESDI 4201. This command code must be written to the resident internal IOPB (Word zero, upper byte), and the location of the external IOPB must be pointed to by the IOPB pointer (Words 10 and 11), the IOPB memory type, and the address modifier code (Word 12). When the GO bit is set, all information in the resident IOPB isignored except for the IOPB pointer address. The V/ESDI 4201 will next go to the address in the IOPB pointer and read in the external IOPB, and then the commands from this IOPB are executed.

The Fetch and Execute command code can only be used in the resident IOPB. If a Fetch and Execute command code is found in an external IOPB (i.e., not in Short I/O space), an error (code 15) will result. If the external IOPB is not fetchedsuccessfully during a Fetch and Execute command, a Busy error will result. The only action taken by the V/ESDI 4201 under these circumstances will be to set the Operation Done bit and the Error bit in the Command/Status Register and to wait for the host to respond. No interrupts will be generated because it is not possible for the V/ESDI 4201 to know which interrupt vector to use, or which command options are in effect when an external IOPB is read incorrectly. This action will be the same in the case of linked external IOPBs. Fetch and Execute commands cannot be nested (i.e., the resident IOPB cannot point to another Fetch and Execute).

9C Verify Track Sequential

This command is the same as the Verify Track command, except the zero latency feature is turned off and the data is always read starting at sector zero.

9E Extended Diagnostics

This command causes diagnostic tests not included in the normal diagnostic command (command code 70) to be performed during system power-up. If the IOPB buffer address is nonzero, the VME buffer read and write functions are tested. If the IOPB cylinder, head and sector numbers are all nonzero, then the following tests are performed: select drive and check for unit ready, issue seek to the drive, wait for "on cylinder" and check write protect, format track, and verify track. Any errors will be reported as commands completed with error (command status code 82 plus the appropriate error code). Once the command has completed, the results of the tests will reside in the last byte of the IOPB.

A0 ESDI Pass-Through

This command issues the ESDI command, located in bytes four and five of the IOPB, to the specifed drive. If the command returns status information, that information is placed in bytes six and seven of the IOPB.

A1 Read And Scatter

The Scatter command allows the user to place contiguous disk data in noncontiguous system memory. The disk control aspects of the Scatter command are identical to those of the read sectors command (81). For more information on scatter, see Section 4.

A2 Gather And Write

The Gather command allows the user to place noncontiguous areas of system memory in contiguous disk areas. The disk control aspects of the Gather command are identical to those of the write sectors command (82). For more information on gather, see Section 4.

A3 Read And Scatter 32

This command has the same function as command A1, Read and Scatter, except that the data is transfered in 32-bit wide bursts.

A4 Gather And Write 32

This command has the same function as command A2, Gather and Write, except that the data is transferred in 32-bit wide bursts.

A5 Read And Scatter 32 (list 2)

This command has the same function as command A3, Read and Scatter 32. However, some users need to use an alternate list format that is not compatible with the standard Scatter/Gather list required by commands A1, A2, A3, and A4. Refer to Section 4 of this user's guide for details of the list format.

A6 Gather And Write 32 (list 2)

This command has the same function as command A4, Gather and Write 32. However, some users need to use an alternate list format that is not compatible with the standard Scatter/Gather list required by commands A1, A2, A3, and A4. Refer to Section 4 of this user's guide for details of the list format.

APPENDIX D ERROR CODES

All error codes are in hexadecimal format.

10 Disk Not Ready

The disk ready signal output is tested at the beginning of any command requiring disk data movement. Error 10 is posted if the disk is not ready. This code is typically posted when an attempt is made to access a disk before the V/ESDI 4201 has received the Ready signal from the drive).

11 Not Used

12 Seek Error

If the V/ESDI 4201 cannot find the required sector of data within two revolutions, it will try to verify that the head is on the right track by reading several sectors. If the cylinder number in the header is incorrect, then error 12 is issued (See related errors 29 and 2A).

13 ECC Error-Data Field

This error is issued if the computed ECC on the data did not agree with the ECC appended to the data on the disk and if no error correction was attempted (see related error 23).

14 Invalid Command Code

The command code in the IOPB (byte zero) was not valid.

15 Illegal Fetch and Execute Command

This error indicates that a Fetch and Execute command was encountered in external memory. A Fetch and Execute command is only valid when it occurs in the on-board Short I/O space.

16 Invalid Sector in Command

The target sector in the IOPB (byte seven) was greater than the capacity of the drive as specified for that drive in byte four of the UIB. This check is performed before the command is executed.

17 Illegal Memory Type

Either the memory type specified for the buffer address is not two, or three, or the IOPB address is not a one, two or three as required.

18 Bus Timeout

This error indicates that bus acquisition was not completed within 100 milliseconds of a request. This error is typically caused by a nonexistent address or address modifier in the data transfer IOPB.

19 Header Checksum Error

This error indicates that there was an error in the header field.

1A Disk Write Protected

This error is issued when attempts are made to write to a disk that is write protected.

1B Unit Not Selected

This error is issued when a unit select was made and the unit failed to respond with Unit Selected. This occurs when either the drive unit number is incorrectly selected, the drive is not powered up, or the cable is not properly connected.

1C Seek Error Timeout

This error occurs when a Clear Fault or Restore failed to correct a seek error from the drive within three seconds. If this error is issued, check to make sure the "B" cable is connected correctly.

1D Fault Timeout

This error is issued when a Clear Fault or Restore failed to correct a fault condition from the drive within three seconds. If this error is issued, check to make sure the "B" cable is connected correctly.

1E Drive Faulted

This indicates that a fault condition exists in a selected unit. The Fault should be cleared by a Restore command. This error is issued when the drive tries to access a nonexistent head or cylinder. Check the drive manual to ensure that the UIB contains the proper settings.

1F Ready Timeout

This error is issued when a Clear Fault or Restore failed to bring the drive ready within three seconds.

20 End of Medium

This error indicates that a multisector transfer exceeded the end of the medium.

21 Translation Fault

This fault indicates that the volume which was specified in the IOPB contains zero heads in the UIB. This error is usually caused by an error in the UIB.

22 Invalid Header Pad

This error indicates that an improper post-header pad byte was encountered.

23 Uncorrectable Error

When this error is posted, error correction was attempted on the data field and the error was found to be uncorrectable.

24 Translation Error, Cylinder

This indicates that the translation of a logical sector resulted in a bad cylinder number. If the drive's UIB is correct, then the logical sector is invalid.

25 Translation Error, Head

This error occurs when the translation of a logical sector resulted in a bad head number. If the drive's UIB is correct, then the logical sector is invalid.

26 Translation Error, Sector

When posted, this error indicates that the translation of a logical sector resulted in a bad physical sector number. If the drive's UIB is correct, then the logical sector is invalid.

27 Data Overrun

This indicates a data timeout error. It is generally caused by a missing TX (transmit) or RX (receive) clock from the drive. If this error is issued, check to ensure that the "B" cable is connected correctly. This error may also be caused if the UIB sectors/track is set to '0'.

28 No Index Pulse On Format

During a Format operation, the V/ESDI 4201 looks for the index pulse from the disk drive. If not found within 65 milliseconds, this error is posted.

29 Sector Not Found

If the target sector cannot be found during a Read or Write, this error is issued (see related errors 12 and 2A). This error is issued if a bad sector on the disk is encountered, or if a track is improperly formatted.

2A ID Field Error-Wrong Head

This error is issued if the head number read from the disk in the header field was wrong (see related errors 12 and 29).

2B Invalid Sync In Data Field

This indicates that the first word read from the data field was not a valid sync character.

2C No Valid Header Found

This indicates that during the Read Header command, no valid header was found. After checking every sector (specified by the UIB) including the runt sector and short sector, every header was invalid. This means that the sync character, checksum, and/or post-header pad were invalid. This error is usually posted when attempting to read a disk that has not been formatted for use with the V/ESDI 4201.

2D Seek Timeout Error

If issued, this error indicates that a seek was made and a normal complete response did not occur within 500 milliseconds.

2E Busy Timeout

This error is set on a dual ported drive if Busy has been active for more than 500 milliseconds. This error indicates that one of the controllers has held the drive for too long.

2F Not On Cylinder

The drive must be on cylinder within three seconds after being selected, or this error will result.

30 Rtz Timeout

This error is issued when a Restore command was executed but a normal complete did not occur within three seconds.

31 Invalid Sync In Header

An invalid sync character in the header field will ause this error code to occur.

32 -3D Not Used

3E UIB Skew Factor

This error occurs if the skew factor set in the UIB exceeds the sectors per track specified in byte four of the UIB (plus any spares specified in byte 13 of the UIB).

3F Not Used

40 Unit Not Initialized

This error indicates that a Write or Format Command was attempted on a unit that has not been initialized.

41 Not Used

42 Gap Specification Error

This error occurs when the value for either Gap 1 or Gap 2 in the UIB is less than five.

43 -4A Not Used

4B Seek Error

This indicates that a seek error was reported by the disk drive.

4C -4F Not Used

50 Sectors Per Track Error

This indicates that the number of sectors/track set in the UIB is zero or greater than 160.

51 Bytes Per Sector Specification Error

The bytes per sector are specified in bytes six and seven in the UIB. When the number is less than 256 or greater than 2048, this error is issued.

52 Interleave Specification Factor

If this error occurs, the interleave factor set in byte six of the UIB is either zero or greater than the number of sectors per track. If this error is issued, check to make sure that the UIB pointer in the initialize command is pointing to the correct place in memory. If it is not, a UIB of random data is read during initialization.

53 Invalid Head Address

The capacity of the drive is specified in bytes zero through three of the UIB. This error indicates that the target head address in byte six of the IOPB exceeded the capacity of the drive.

54 Invalid Cylinder Address

The capacity of the drive is specified in bytes 12 and 13 of the UIB. This error indicates that the target cylinder in Word two of the IOPB exceeded the capacity of the drive.

55 - ESDI Timeout

This error occurs if a V/ESDI 4201 command does not complete within the time limit indicated by the ESDI specification.

56 Zero Sector Count

IOPB Sector Count cannot be 0 for read or write operation.

57 -5C Not Used

5D Invalid DMA Transfer Count

This error indicates that the specified transfer count caused the V/ESDI 4201 to attempt to transfer an odd number of bytes.

5E -5F Not Used

60 IOPB Failed

When this error is posted, a bus error occurred during the transfer of an external IOPB. The IOPB pointer (Words 10 and 11), shows the starting address of the IOPB on which the bus error occurred. (See error code 61 for details of bus errors during DMA transfers.)

61 DMA Failed

This error indicates that a bus error occurred during the DMA transfer of data to or from the buffer or the bus. Words five and six of the IOPB (the buffer address) point to the start of the sector block in system memory where the error occurred. Words two and three of the IOPB (the disk address) point to the disk location where the disk transfers were when the error occurred. If the disk was addressed in physical mode, the disk error location will be a physical location. If logical address mode was used, the disk error location will be a logical location.

62 Illegal VME Address

For 8- or 16-bit transfers, the starting address of the VME buffer must fall on a word boundary (even address, multiple of two). For 32-bit transfers, the starting address of the VME buffer must fall on a long word boundary (even address, multiple of four).

63 -69 Not Used

6A Unrecognized Header Field

During a read/verify command, one or more of the requested headers were not found. This error differs from error 29 (Sector Not Found) in that one or more headers were ignored because of invalid sync, checksum, or post-header pad fields. Possible causes could include unformatted sectors (UIB sectors/track less than disk sectors/track), UIB Gap one too small, or "short sector" pulse from drive.

6B - Mapped Header Error

This indicates that the sync field of a header appeared to be a valid mapped field, but the remainder of the header was unrecognizable. See error code 6A for probable causes.

6C -6D Not Used

6E Spare Sector Number

This error indicates that the spare sector number to be mapped is beyond the end of the track. The spare sector number must be less than the number of sectors/track.

6F No Spare Sector Enabled

A Map Sector command was issued and the UIB did not specify spare sector mapping.

70 -76 Not Used

77 Command Aborted

This indicates the V/ESDI 4201 observed and serviced the Abort bit in the CSR.

78 ACFail Detected

This indicates the V/ESDI 4201 received the VME ACFAIL control signal. This error requires a Reset and ACFAIL release before normal operation can continue.

79 -7F Not Used

80 Transfer Assertion Timeout

The ESDI command transfer acknowledge signal was not returned within the specified time limits.

81 Transfer Release Timeout

The ESDI command transfer release signal was not returned within the specified time limits.

82 Status Transfer Timeout

The ESDI status transfer acknowledge signal was not returned within the specified time limits.

83 Status Release Timeout

The ESDI status release acknowledge signal was not returned within the specified time limits.

84 -9F Not Used

A0 S/G List too Large

The Scatter/Gather list size exceeds the sector size. The list cannot be larger than a sector buffer. List size is calculated by multiplying the list element size (eight bytes) by the number of elements specified in byte 27 of the IOPB.

A1 Illegal Element Byte Count

For Scatter/Gather commands (A1 and A2), the element byte count must be a multiple of the bytes/sector parameter. If an illegal element byte count is specified, this error will be returned by the Scatter/Gather commands codes.

A2 -5 Not Used

A2 -AA Not Used

A6 Illegal Entry Count

S/G entry count in IOPB cannot be 0 for a S/G operation.

AB Illegal Element Size

All scatter gather elements must contain an even number of bytes. The controller cannot transfer less than a word; therefore, if an element contains an odd number of bytes, this error will be returned.

AC Illegal List Byte Count

The total byte count specified by the Scatter/Gather list must be a multiple of the bytes/sector parameter. The controller will not complete a Scatter or Gather command with partial sectors, and this error will be returned.

AD Illegal IOPB Sector Count

The IOPB sector count does not agree with the total bytes specified in the Scatter/Gather list.

AE - BF Not Used

C0 Both Bits Set

This error code will be returned if both the Spare Sector Enable bit and the Multiple Spare Enable bit are set.

C1 MSE Without Init Long

This error code is returned if the Multiple Spare Enable bit is set and the unit is not Initialized with the Initialize Long command (7C).

C2 -DF Not Used

E0 - Already in MACSI mode:

The MACSI MODE command has been issued to the controller when the controller is already in MACSI mode.

E1 - Bad CIB address:

The address given for the controller initialzation block is invalid. This address cannot be 0.

E3 - Number of CQEs is bad:

The number of CQEs must be greater than 0 and less than 9.

E5 - No SCSI daughter Card is attached:

The desired command can only be run when a SCSI daughter card is attached.

E7 - MACSI mode is required:

The desired command can only be run in MACSI mode.

E8 - IOPB size is invalid:

The IOPB size entered in the CQE must be larger than 0.

E9 - Number of Commands to Q is invalid:

The number of commands to Q given in the CIB must be greater than 0 and less than 65.

EA -EF Not Used

F0 Mapped Header

A mapped bad header was encountered on the current track.

F1 Sector Not Flagged

The sector to be transferred was not flagged as error 29 in sector ID table (rdeach @ rdca100:).

F2 -FB Not Used

FC No Write List

No write list was available to start gather operation.

FD No Write Buffers

No write buffers were available for gather operation.

FE Out Of Buffers

All buffers were full before gathered data could be transferred.

FF Command Not Implemented

The command issued will be supported in a later release.

APPENDIX E
" C" LANGUAGE DATA STRUCTURES

```
/*
 * C data structures for Interphase 32xx/42xx Short i/o with and without MACSI
 */

/*
 * typedefs used by all the other Interphase structures
 */

typedef unsigned char   BYTE;                   /* 8 bit unsigned       */
typedef unsigned char   UBYTE;                  /* 8 bit unsigned       */
typedef          short  SHORT;                  /* 16 bits signed       */
typedef unsigned short  USHORT;                 /* 16 bits unsigned     */
typedef          short  WORD;                   /* 16 bits siggned      */
typedef unsigned short  UWORD;                  /* 16 bits unsigned     */
typedef          int    INT;                    /* 32 bits signed       */
typedef unsigned int    UINT;                   /* 32 bits unsigned     */
typedef unsigned long   ULONG;                  /* 32+ bits unsigned    */
typedef          long   LONG;                   /* 32+ bits signed      */

#define BIT(x)          (1 << x)

#define LOWORD(x)       (x)
#define HIWORD(x)       (x >> 16)
#define W_SIZ(x)        (sizeof(x)/sizeof(SHORT))  /* size in words     */
```

```
/*
 *      The sructure definition below is for the:
 *
 *               VSMD, 3201, 4200 & 4201 disk IOPB.
 *
 *      NOTE:
 *              Short i/o space can only be written/read by short memory
 *              accesses ie 16 bit.
 *
 */

typedef struct
{
        BYTE  i_cmd;                            /* command code         */
        BYTE  i_cmdopt;                         /* command options      */
        BYTE  i_status;                         /* status of command    */
        BYTE  i_error;                          /* error code           */
        union
        {
                union
                {
                        UINT i_blkno;           /* logical block number */
                        struct                  /* for short i/o access */
                        {
                                USHORT i_blkno_h;
                                USHORT i_blkno_l;
                        } i_shio;
                } i_log;
                struct
                {
                        USHORT i_cyl;           /* cyl number           */
                        BYTE  i_head;           /* head number          */
                        BYTE  i_sect;           /* sector number        */
                } i_phys;
        } i_disk_addr;
        USHORT  i_secnt;                        /* # sectors            */
        USHORT  i_buf_h;                        /* buffer address hi    */
        USHORT  i_buf_l;                        /* buffer address lo    */
        BYTE    i_memtype;                      /* memory type          */
        BYTE    i_adrmod;                       /* addr modifier code   */
        BYTE    i_intlevel;                     /* interrupt level      */
        BYTE    i_nvector;                      /* normal interrpt vect */
        BYTE    i_burstlen;                     /* DMA burst length     */
        BYTE    i_evector;                      /* error interrupt vect */
        USHORT  i_nxt_h;
        USHORT  i_nxt_l;
        BYTE    i_imemtype;                     /* IOPB memory type     */
        BYTE    i_iadrmod;                      /* IOPB addr mod code   */
        BYTE    i_skew;                         /* spiral skew          */
        BYTE    i_entry_cnt;                    /* Entry count          */
} VSMD_IOPB;
```

```
/*
 *      Status codes for i_status
 */

#define IP_OK           0x80
#define IP_DBUSY        0x81
#define IP_ERROR        0x82
#define IP_EXCEPT       0x83

/*
 *      The sructure definition below is for the:
 *
 *              VSMD, 3201, 4200 & 4201 disk UIB.
 */

typedef struct
{
        BYTE    u_v0shead;              /* volume 0 starting head       */
        BYTE    u_v0nhead;              /* # heads per cyl              */
        BYTE    u_v1shead;              /* volume 1 starting head       */
        BYTE    u_v1nhead;              /* # heads per cyl              */
        BYTE    u_nsect;                /* # sectors per track          */
        BYTE    u_skew;                 /* spiral skew factor           */
        USHORT  u_secsize;              /* sector size                  */
        BYTE    u_gap1;                 /* size of gap 1                */
        BYTE    u_gap2;                 /* size of gap 2                */
        BYTE    u_intrlv;               /* interleave factor            */
        BYTE    u_retry;                /* # of retries                 */
        USHORT  u_ncyl;                 /* # cylinders                  */
        BYTE    u_attrib;               /* attribute flags              */
        BYTE    u_rattrib;              /* reserved                     */
        BYTE    u_istchgl;              /* status change interrupt level*/
        BYTE    u_istchgv;              /* status chng interrupt vector */

/* NOTE: The rest of this structure is ONLY used for an INITLONG cmd    */

        BYTE    u_wtdelay;              /* write delay                  */
        BYTE    u_nspares;              /* number of spares             */
        BYTE    u_exresrv[12];          /* unused bytes in extended uib */
} VSMD_UIB;
```

```
/*
 *      bit values for UIB attributes & Reserved attributes
 *      (really there are 2  attribute fields)
 *
 *      reserved attribute bits 4 & 5 are ESDI select
 *                      00      Hard Sector Drive
 *                      01      Embedded Servo Drive
 *                      10      Address Mark Drive
 */

#define UA_RSE          BIT(7)                  /* runt sctr enable     */
#define UA_SSE          BIT(6)                  /* spare sctr enable    */
#define UA_CE           BIT(5)                  /* cacheing enable      */
#define UA_STC          BIT(4)                  /* status change int en */
#define UA_DLP          BIT(3)                  /* Dual port enable     */
#define UA_INH          BIT(2)                  /* increment by head    */
#define UA_MBD          BIT(1)                  /* move bad data        */
#define UA_RSK          BIT(0)                  /* reseek enable        */

#define RA_MSPARE       BIT(0)                  /* multi spares         */
#define RA_4UNIT        BIT(6)                  /* 4 unit enable        */
#define RA_XADDR        BIT(7)                  /* SMD extd addressing  */

#define ESDI_HARD       0x00                    /*  esdi hard-sector    */
#define ESDI_SERVO      0x01                    /*  esdi embedded servo */
#define ESDI_SOFT       0x10                    /*  esdi address-mark   */


#define U4STATBASE      (0x1fa)         /* address of 4 drive status    */
#define XSLOP           (U4STATBASE - 4 - sizeof(VSMD_IOPB) - sizeof(VSMD_UIB))

/*
 *      The sructure definition below is for the:
 *
 *               VSMD, 3201, 4200 & 4201 short/io memory space
 *                      (NON-MACSI mode,
 */
typedef struct
{
        BYTE            vsmd_2dstatus[2];       /* old  2 drive status  */
        USHORT          vsmd_cstatus;
        VSMD_IOPB       vsmd_iopb;
        VSMD_UIB        vsmd_uib;
        BYTE            bs_slop[XSLOP];
        BYTE            vsmd_dstatus[4];
        USHORT          vsmd_ostatus;   /* optional status change/reg   */
} VSMD_SHIO;
```

```
/*
 *      bit values for drive status registers
 *
 *              drive ready is a signal from the actual drive
 *                                  - - -
 *              unit ready is a bit generated from the combination
 *              of other status bits.
 */

#define DS_UREADY       BIT(7)                  /* unit ready           */
#define DS_PRESENT      BIT(6)                  /* drive present        */
#define DS_SEEKERR      BIT(5)                  /* seek error           */
#define DS_FAULT        BIT(3)                  /* drive faulted        */
#define DS_ATTENT       BIT(2)                  /* ESDI attention bit   */
#define DS_WRTPROT      BIT(1)                  /* disk write protected */
#define DS_DREADY       BIT(0)                  /* drive ready          */

/*
 *      bit values for command/status register
 *              ( USED in NON-MACSI mode)
 */

#define CS_SLED         BIT(15)                 /* status LED           */
#define CS_BOK          BIT(14)                 /* board OK             */
#define CS_SFEN         BIT(13)                 /* sysfail enable       */
#define CS_BDCLR        BIT(12)                 /* board clear          */
#define CS_ABORT        BIT(11)                 /* abort operation      */
#define CS_BERR         BIT(8)                  /* bus error            */
#define CS_GO           BIT(7)                  /* start a command      */
#define CS_DONE         BIT(6)                  /* operation complete   */
#define CS_STATCHG      BIT(5)                  /* status change        */
#define CS_ERR_LC       BIT(4)                  /* error on last cmd    */
#define CS_BUSY         CS_GO                   /* executing command    */

/*
 *      bit values for command options
 */

#define COP_LNKIOPB     BIT(5)                  /* linked  IOPBs        */
#define COP_LOG_TRAN    BIT(4)                  /* logical translation  */
#define COP_DISERR      BIT(2)                  /* no err detection     */
#define COP_INT_EN      BIT(1)                  /* int on completion    */
#define COP_ECC_EN      BIT(0)                  /* apply ecc            */
#define MEMT_32BIT      0x03                    /* 32-bit memory        */
#define MEMT_16BIT      0x02                    /* 16-bit memory        */
#define MEMT_I16BIT     0x01                    /* 16-bit internal mem  */

#define ADRM_STD_N_D    0x39                    /* standard normal data */
```

```
/*
 * Product codes returned from the handshake command
 */

#define IP3201          0x65            /* Guerrila product code         */
#define IP4201          0x74            /* Panther product code          */
#define IP3200          0x49            /* V/SMD product code            */
#define IP4200          0x66            /* Cheetah product code          */
#define IP4400          0x82            /* Phoenix product code          */

typedef struct
{
        BYTE    p_code;         /* product code        */
        BYTE    p_var;
        BYTE    p_revh;
        BYTE    p_revl;
        BYTE    p_idh;
        BYTE    p_idl;
        BYTE    p_month;        /* month stored as BCD */
        BYTE    p_day;          /* day stored as BCD   */
        USHORT  p_year;         /* year stored  as BCD */
}HNDSHK;


/*--------------------------------------------------------------------*/
/* MACSI structures                                                   */
/*--------------------------------------------------------------------*/

#define IOPB_BUF_SIZ    32

typedef struct
{
        char iopb_buf[IOPB_BUF_SIZ];
} IOPB_BUFF;

typedef struct cqe      {                       /* command queue entry  */
                        USHORT control;
                        USHORT iopbaddr;
                        USHORT cmdtag_h;
                        USHORT cmdtag_l;
                        BYTE length;
                        BYTE queue;
                        USHORT rsrvd;
                        }CQE;

typedef struct mcsb     {                       /* Master status block  */
                        USHORT msr;
                        USHORT mcr;
                        USHORT iqar;
                        USHORT qhead;
                        USHORT rsrvd[4];
                        }MCSB;
```

```
typedef struct crb      {                       /* command response block */
                        USHORT crsw;
                        BYTE    type;           /* IOPB type */
                        BYTE    rsrvd;
                        USHORT   cmdtag_h;
                        USHORT   cmdtag_l;
                        BYTE   length;
                        BYTE   queue;
                        USHORT rsrvd1;
                        IOPB_BUFF   copyiopb;
                        }CRB;

typedef struct cib      {                       /*MACSI initialization*/
                        USHORT cib_num_cqes;
                        USHORT cib_crb_off;
                        USHORT cib_m_flags;     /* MACSI FLAGS  */
                        USHORT cib_num_free;    /* NUM free entries */
                        USHORT cib_ce_vectr;
                        USHORT cib_cn_vectr;
                        USHORT cib_bback;       /* Back to Back write cnt */
                        USHORT cib_rrobin;      /* round robin switch count */
                        USHORT cib_res8;
                        USHORT cib_res7;
                        USHORT cib_res6;
                        USHORT cib_res5;
                        USHORT cib_res4;
                        USHORT cib_res3;
                        USHORT cib_res2;
                        USHORT cib_res1;
                        }VSMD_CIB;
```

```
#define NUM_CQES        8

#define TSTATBASE       0x1e0   /* address where SCSI tape status starts */

/*
 *      The sructure definition below is for the:
 *
 *               VSMD, 3201, 4200 & 4201 short/io memory space
 *                        (MACSI mode)
 */
#define CRB_OFFSET      (0x1a0)         /* address of Command RES. block */
#define CSLOP           (CRB_OFFSET - sizeof(MCSB) - (NUM_CQES * (sizeof(CQE) +
sizeof(IOPB_BUFF))) )
#define MSLOP   (TSTATBASE  - CRB_OFFSET - sizeof(CRB))
typedef struct
{
        MCSB            macsi_mcsb;
        CQE             macsi_cqe[NUM_CQES];
        IOPB_BUFF       macsi_iopb[NUM_CQES];
        BYTE            bs_slop[CSLOP];
        CRB             macsi_crb;
        BYTE            bs_slop2[MSLOP];
        USHORT          macsi_tsreg[8]; /* tape status registers */
        USHORT          bs_resrvd[5];
        BYTE            macsi_dstatus[4];
        USHORT          macsi_bsb;      /* MACSI board status bits */
} MSHIO;

/*
 * MASTER STATUS register values
 */

#define MSR_BOK         BIT(1)

/*
 * BOARD status BITS
 */

#define BSB_SCSI        BIT(0)          /* Firmware saw a SCSI daugher card */

/*
 * IOPB TYPES
 */

#define DISK_IOPB_TYPE  1               /* 4201/4200 IOPBs      */
#define PSCSI_IOPB_TYPE 2               /* panther scsi IOPBS   */
```

```
/*
 * Command status Word values
 */
#define CRSW_CRBV       BIT(0)          /* VALID/DONE BIT        */
#define CRSW_CC         BIT(1)          /* Command complete      */
#define CRSW_ER         BIT(2)          /* ERROR last cmd        */
#define CRSW_EX         BIT(3)          /* Exception last cmd    */
#define CRSW_CQA        BIT(6)          /* CQE available         */

/*
 * IQAR register bits
 */

#define IQAR_IQEA       BIT(15)         /* Interrupt on CQE available   */

/*
 * CQE control register values
 */
#define CQE_GO          BIT(0)
#define CQE_HPC         BIT(2)          /* set for high priority        */
#define CQE_SORT        BIT(7)          /* set for sorting              */

/*
 * MACSI CIB flags
 */
#define M_OVL_SEEK      BIT(1)  /* enable overlapped seeks */
#define M_RESTORE       BIT(2)  /* enable auto-restore on drive faults */
```

```
/*
 * Structure of IO Parameter Block (IOPB) SCSI VTAPE
 */

typedef struct {
        BYTE    t_cmd;                  /* command register                  */
        BYTE    t_cmdopt;               /* command code options              */
        BYTE    t_status;               /* status code register              */
        BYTE    t_error;                /* error code register               */
        USHORT  t_bytrecd_h;            /* bytes per record MSW              */
        USHORT  t_bytrecd_l;            /* bytes per record LSW              */
        USHORT  t_reccnt;               /* record count                      */
        USHORT  t_buf_h;                /* buffer address MSW                */
        USHORT  t_buf_l;                /* buffer address LSW                */
        BYTE    t_memtype;              /* memory type                       */
        BYTE    t_adrmod;               /* address modifier code             */
        BYTE    t_intlevel;             /* interrupt level                   */
        BYTE    t_burstlen;             /* DMA  burst count                  */
        BYTE    t_nvector;              /* normal comp. interrupt vector     */
        BYTE    t_evector;              /* error Interrupt vector            */
        BYTE    t_fmkcnt;               /* filemark count                    */
        BYTE    t_rsvl;                 /* reserved                          */
        USHORT   t_rsv2;                /* reserved                          */
} STAPE_IOPB;

/*
 * Structure of a SCSI command descriptor BLOCK
 *      MUST be a long word multiple!
 */
typedef struct {
        BYTE    cmd;
        BYTE    lba_msb;                /* logical block address             */
        USHORT  lba_lsw;                /* logical block address             */
        BYTE    nblk;                   /* number of blocks                  */
        BYTE    c_byte;                 /* control byte                      */
        BYTE    rsrvd[2];               /* NOT used     YET...               */
}SP_CDB;
```

```
/*
 * Structure of IO Parameter Block (IOPB) SCSI PASS-THRU
 */

typedef struct {
        BYTE    s_cmd;                  /* command register                 */
        BYTE    s_cmdopt;               /* command code options             */
        BYTE    s_status;               /* status code register             */
        BYTE    s_error;                /* error code register              */
        USHORT  s_mtlen_h;              /* Maximum transfer length (bytes)  */
        USHORT  s_mtlen_l;              /* Maximum transfer length (bytes)  */
        USHORT  s_rsv1;                 /* RESERVED                         */
        USHORT  s_buf_h;                /* buffer address MSW               */
        USHORT  s_buf_l;                /* buffer address LSW               */
        BYTE    s_memtype;              /* memory type                      */
        BYTE    s_adrmod;               /* address modifier code            */
        BYTE    s_intlevel;             /* interrupt level                  */
        BYTE    s_burstlen;             /* DMA  burst count                 */
        BYTE    s_nvector;              /* normal comp. interrupt vector    */
        BYTE    s_evector;              /* error Interrupt vector           */
        SP_CDB  s_cdb;                  /* SCSI cmd descriptor block        */
} SPASS_IOPB;
```

GLOSSARY OF TERMS

Address

A number which identifies a specific location in memory.

Address Modifier

A VMEbus-specific 6-bit code used to modify the way a VMEbus address is interpreted. The V/ESDI 4201 uses address modifiers 29 and 2D to specify short I/O access.

Bad Track Mapping

The process by which the V/ESDI 4201 identifies a track as unusable and makes the proper adjustments. High-density media and recording technologies make it impossible to guarantee flawless media. It is therefore necessary to "map" or identify the bad areas of the disk. The "mapping" tells the controller where the bad areas are and where to find the replacement area.

Bus Clear

A signal driven by the system controller to recommend that a bus master relinquish control of the bus.

Bus Grant

The process by which the system controller lets a bus master have control of the bus.

Bus Request

The process by which a potential bus master indicates to the system controller that bus control is needed.

Caching

Caching is an intelligent algorithm used to predict which data will be required next by the system after a read request has been received. The desired sectors of data are read from the disk, and stored in the buffer.

Cylinder

The three-dimensional cross section of a given track on a stack of platters.

Data Strobe

A signal from a bus master to a bus slave that indicates that a data transfer is about to occur.

Data Transfer Acknowledge (DTACK)

An affirmative acknowledgment sent across the bus to a master to indicate that either data has been received by the slave or that the slave has data available.

DMA (Direct Memory Access)

Direct memory access is an activity that transfers data directly from a peripheral into system memory without requiring the system CPU.

DTB (Data Transfer Bus)

Data transfer bus as used in the VME standard.

Dual Ported

A disk drive capable of interfacing with two independent SMD controllers is said to be dual ported.

ECC (Error Correction Code)

The V/ESDI 4201 uses written data to calculate a four-byte error code which it appends to the data field during WRITE operations. During data READS, the V/ESDI 4201 uses these four bytes to detect and correct (if necessary and possible) data errors. Any number of data bit errors can be detected, but to be correctable, data errors cannot span more than eleven consecutive bits.

EXP (Expanded Double Bus)

Expanded double bus VME board as used in the VME standard.

Explicit Seek

A seek that is specifically requested by the host, i.e., one not inherent in another command, is said to be explicit.

Firmware

A computer program written onto a storage device such that it cannot be accidentally erased, (i.e., it is stored in Read Only Memory (ROM)).

Head

A device which reads, records, or erases information on the magnetic medium.

Host

Refers to the primary or controlling processor in a system.

Implied Seek

Seeks that are embedded within certain commands and are necessary for the command to be completed, are said to be implied.

Interleave

The sectors on a track are numbered logically from 0 to N, but do not necessarily have to be placed in a physically sequential order. The interleave of a track defines the ordering of the sectors. An interleave of 1 (or 1:1) means that the logically consecutive sectors are also physically consecutive. On a disk with an interleave of 2 (2:1), logically consecutive sectors are physically separated by one sector. For an interleave of 3 (3:1), logically consecutive sectors are physically separated by two sectors, and so on. The idea is for the disk head be approaching the data of interest when the data is requested. Due to zero latency and caching features of the V/ESDI 4201, an interleave of 1 is usually optimum.

Interrupt

The interrupt capability of the VMEbus provides a means by which devices can interrupt normal bus activity. These interrupts are prioritized (priority interrupts) into seven levels. Priority interrupts utilize signals IACK, IACKIN, and IACKOUT to acknowledge that the interrupt has been generated.

IOPB

The I/O Parameter Block is a 14-word list of parameters necessary to define a command.

Latency

Latency refers to the time delay from the start of a transaction to the point at which data transfer to/from a disk actually begins. It is a combination of seek time, rotational latency, and controller delays.

Linked IOPBs

Linking IOPBs is a method of processing a series of IOPB commands; thereby reducing driver overhead and host CPU intervention.

Logical Disk Addressing

Most operating systems store data in logical addresses, and in order to access or move the data from a disk storage unit, the host CPU must translate the logical addresses into physical addresses.

Media Flaw Map

The media flaw map is provided by the manufacturer of most disk drives to illustrate the integrity of the media.

Memory Type

The memory type specifies 8-, 16-, or 32-bit data transfers. The V/ESDI 4201 supports memory types for 16- and 32-bit VMEbus transfers.

Multitasking

Refers to the ability of a device to perform two or more tasks concurrently.

Physical Disk Addressing

When physical disk addressing is used, the actual physical address (in terms of cylinders, heads, and sectors) on the disk must be explicitly specified. That is a cylinder address of zero would specify the first cylinder on the disk in Physical Mode.

Rotational Latency

After the disk head settles on a track, the rotational latency is the time that elapses before the first sector of interest (to the controller) comes under the head.

Sector

A sector is the smallest unit of memory on a disk that can be accessed separately.

Sector Mapping

Sector mapping is a method of bad area mapping that requires a spare sector to be included on each track to accommodate defective media.

Sector Slip

Sector slip refers to the process of moving a bad sector during formatting. When a bad sector is mapped, the track is reformatted such that the bad sector is moved to the next consecutive sector, which is in turn moved to the next consecutive sector, and so on until the last sector is moved into the spare sector space.

Seek

This command causes the read/write head to move to the requested location on the disk.

Seek Time

The time that is required for the head to move from its previous position on the drive to the requested cylinder.

Short I/O

Short I/O is a VMEbus-specified block of memory to facilitate processing of certain information. On the V/ESDI 4201, Short I/O is a 512-byte block of memory set aside for handling command parameters, control information, and statuses that occur when the host CPU is acting as the bus master. It is called Short I/O space because the upper 16 VMEbus address lines are ignored, and only the lower 16 are used for transactions that take place in Short I/O.

Skew

The skew defines the number of sectors that sector zero is offset from the last sector of the previous track.

State Machine

A state machine is a digital device that is capable of generating an output signal based on its current "state" and the input signal.

Track

The track is a path along which the head travels as it reads, records, or erases information on a magnetic medium.

UIB (Unit Initialization Block)

The (UIB) Unit Initialization Block is used in conjunction with the Initialize command to set the operating parameters for the drive(s) being used. One UIB is required for each drive.

Virtual Buffer

Virtual buffering is a scheme where buffers (blocks of RAM) are allocated and deallocated as they are requested or released by the VMEbus or disk. At any given time, individual buffers may be allocated to the disk, the VMEbus, or the cache. This environment creates what is effectively a buffer much larger than the physical memory.

VMEbus

The VMEbus is an industry standard high-performance 32-bit bus designed with an open bus architecture.

Wait State

The period of dormancy between memory operations.

INDEX

LET US KNOW WHAT YOU THINK!

At Interphase, we are constantly working to improve our products to better meet the needs of our customers. One area we are always concerned with is user documentation. Make a copy of this form, fill it out and mail it to us, but save the original. That way, we can get a better understanding of what would be most helpful to you, and in the future if you have any further comments, you will still have a handy way of communicating them to us.

	Excellent	Good	Average	Poor
Contents	□	□	□	□
Readability	□	□	□	□
Organization	□	□	□	□
Appearance	□	□	□	□
Accuracy	□	□	□	□
Usefulness as a Reference	□	□	□	□

Have you used User Manuals provided with other manufacturer's disk/tape controllers? yes_____ no_____

If yes, how does this one compare?____

____

Although every effort is made to insure the integrity of the information provided to our customers, sometimes oversights do occur. Inform us of any errors you may have noticed in the text and give the page number on which it occurs if possible.

____

____

Comments:____

____

____

____

____

____

Name:_________________________
Title:_________________________
Company:______________________
Mailing Address:_____________________________________
City:_________________________ State:__________________
Telephone:(   )_______________ Zip:____________________

We at Interphase want to hear from you.

Interphase Corporation
Technical Writing Dept.
2925 Merrell Road
Dallas, Tx 75229-9990

INTERPHASE
CORPORATION

OPEN SYSTEMS CONTROLLERS™

Disk • Tape • Networking

2925 Merrell Road • Dallas, Texas 75229 • (214) 350-9000 • FAX: (214) 352-4124 • NASDAQ-NMS:INPH
Interphase International
Astral House, Granville Way • Bicester, Oxon OX6 OJT • (01144) 869-321222 • FAX: (01144) 869-247720