CONVEX VMEBUS CONFIGURATION GUIDE

Document No. 081-013530-000

First Edition, Rev. 1
September 23, 1991

CONVEX Computer Corporation
Richardson, Texas USA

CONVEX VMEBUS CONFIGURATION
Guide

Printed in the United States of America

VMEbus Controller Configuration Guide

Compatibility Guidelines Summary:

- All controllers installed in a Convex VMEbus must have a discrete address with adequate address separation between each controller in the same bus.
- All controllers installed in a Convex VMEbus must use Bus Request/Grant Level 3.
- Each controller installed in a Convex VMEbus must have a discrete interrupt relative to the other controllers in the same VMEbus.
- VMEbus controllers used in the Convex VMEbus should act as bus Masters with DMA capabilities to provide the desired performance, it is not a requirement. Controllers that are not bus Masters will run slower.
- Each VIOP is allowed a maximum of 1024, 4096byte memory windows. These windows may be mapped to main memory or to local memory on the VIOP. The VIOP uses some of the windows and each controller uses some of the windows. The total number of windows used per VIOP must not be greater than 1024. This is the total of both VMEbuses and the VIOP.
- Each VIOP has its own local memory of 512K bytes. This memory will be used by EGOS, the operating system, and by the VMEbus device drivers for the controllers installed in its own buses. Care must be taken not to allow a controller to exceed the total allowed local memory. The requirements vary with the Release of the OS. Refer to the ConvexOS VIOP Device Driver Memory Usage Guide, Doc No 710-003430-022.
- It is necessary that developers of software and installers of hardware understand the bandwidth limitations of each bus in the VMEbus system. The throughput of the VIOP peaks out at a sustained 12MB/sec when both VMEbuses on one VIOP are in use. Since this is an agregrate, each bus is allowed a maximum of 6MB/sec.
- The sustained throughput of the VIOP when using only one VMEbus peaks out at 10MB/sec.
- The Convex VIOP does not support all address modifiers. If in doubt read the section below on address modifiers.

OVERVIEW:

This document was written to provide an informational guideline for attaching a VMEbus controller to a Convex VMEbus system. The goal of this document is to provide a general description of the interface between the Convex systems and the VMEbus and point out the required details for integrating a VMEbus device to a Convex Input/Output System. This document will not give all of the details on each part of the VME system in a Convex system but, it does list all of the documents. To accomplish the goal the topics discussed will be the Convex Memory Control, the I/O bus, Channel Control Units, VMEbus Input/Output Processor, VMEbus Control Unit, VMEbuses, Main Memory Windows, VIOP Local Memory Windows, VMEbus Addresses, Address Modifiers, Interrupts, Convex standard controllers, and Customer added controllers.

Convex Memory Control

In the different classes of Convex computer systems there are similar architectures, however, there are differences. This paper will touch upon the basic differences between a C1, C2, Neptune, and Javelin type machines. One of the differences between machines is the Memory Control Unit (MCU), and the interconnect between the Pbus and Memory.

In a C1 Class machines the main processors, vector and scalar, share a common memory with the Input/Output System through an MCU. The MCU is the bus arbiter and allows access to the Central Processors or to the I/O processors in a time shared, round robin fashion. Access to the Central processor is through the Physical Cache Unit(PCU), memory bus, MCU, and memory array bus, to the Memory Array Units (MAU). Access for the Channel Control Units (CCU) is through the PBUS, MCU, memory array bus, and to the MAUs. The MCU, the bus arbiter, controls access to memory by granting access to the CPU, a CCU, or the SPU. Each VIOP, accesses system memory via the PBUS through 1024, 4096-byte windows, which are programmed by the CCU to access main memory through the PBUS. The hardware that implements these windows converts CCU internal read/write requests from the CCU microprocessor or from one of the VMEbus control units into the appropriate PBUS transactions. There can be up to five CCUs installed in a C1 class PBUS with no more than one PBUS in a C1 system.

In the C2 class machine the PIA (PBUS Interface Adapter), or PI2, (Pbus Interface-2), performs the work of the MCU. With the PIA only one PBUS can be supported in the C2 system. With the PI2, (and adding a second PI2), two PBUSs can be supported. The PBUS in the C2 only supports four CCUs per PBUS.

The Javelin I/O system is the same as the C2 class machine.

The Nepture I/O system uses a PBUS that only supports 2 CCUs. These two CCUs install into a PBUS that interfaces to memory through the Neptune Interface Adapter (NIA). There can be four PBUSs attached to one NIA, each with two CCUs for a total of eight CCUs. By removing a CPU cabinet from the Neptune cluster and replacing it with an I/O cabinet, an additional NIA can be installed with four PBUSs, each PBUS supporting two CCUs.

I/O BUS (PBUS)

The PBUS is an 8-byte wide block-mode interface with parity on each byte. With a 10Mhz clock, the Pbus has a burst transfer rate of 80Mbytes/second. Our concern in this discussion is the CCU interface to the PBUS. There are up to five CCUs allowed on a PBUS. Each CCU is allowed access to the bus long enough to transfer a header, data, and terminate. The request for the bus, the bus grant, and the cycle terminate, are all Pbus protocol signals executed within one clock for each signal. The header is 8 bytes of data transfered as the first 8 bytes after the bus grant. The header contains address, bytes count, and command. The commands contained within the header would be read, write, scrub, or test and set. The header is followed by the data to be transferred and may be up to 64 bytes of data for one transfer, and then a bus terminate.

Then the bus is checked to see if another device is requesting service in a round robin fashion. There is a buslock signal that will allow a high-speed device to control the bus for the duration of the transfer. Such a transfer may contain more than 64 bytes of data.

On a C1, if a CCU asserts buslock, the arbiter will allow the whole DMA to proceed. The MCU will not negotiate for any other PBUS ownership until the transfer is completed. Although PBUS transfers are 64 byte transfers between each PBUS negotiation, the use of the bus will not be passed to another CCU until the transfer is complete. The buslock signal is used by other CCU's which transfers 4096 bytes at a time (per PBUS transaction) when doing DMA. On a C2, the PIA arbiter assumes buslock is asserted, and allows the whole DMA requested by the CCU to proceed.

Channel Control Units

Channel Control Units are programmable, intelligent I/O processors designed and manufactured by Convex to control the low level details of input/output in a Convex system. CCUs interface with the PBUS and provide connection to standard buses. There are different types of CCUs in the Convex product line for supporting various standard busses. One of the CCUs is the VMEbus Input/Output Processor (VIOP). The VIOP uses the Event Governed Operating System (EGOS), developed at Convex, to manage the attached buses and provide direct memory access (DMA) to main memory. The VIOP uses the VMEbus as a front-end and accepts third party VMEbus controllers. Two VMEbuses can be supported by one VIOP. Each VMEbus that connects to a VIOP uses three cables and a VMEbus control unit (VBCU) designed by Convex to interface the VMEbus card cage to the VIOP. The VIOP is based on the Motorola 68020 micro processor with 512Kbytes of fast local memory. The VIOP contains a cache that acts as a buffer between the VMEbuses and the Pbus. The cache maps into main memory through map registers in the VIOP. There are 1024 map registers which are used by the EGOS system to communicate commands and replies, and to provide a DMA path from the VMEbus into main memory.

The amount of memory in the VIOP is important because the device driver for each controller requires a specific number of pages in the VIOP memory. Drivers also require a driver-dependent number of windows for carrying out DMA. It is possible to configure hardware that is not bootable or useable because of resource conflicts between the drivers, each vying for RAM and windows.

VMEbus Requirements

Convex's VMEbus come in three different configurations.

- First, a nine slot VMEbus that will accommodate a VBCU and seven VMEbus controllers, this unit will occupy an entire VME chassis. The Marketing number for this chassis is VBS-003.
- Second, a Dual VMEbus, with two five slot VMEbuses that will accommodate one VMEbus control unit (VBCU) and four additional VMEbus controllers per bus. The Marketing number for this chassis is VBS-004.
- Third, a Combination Chassis, a six slot VMEbus that will accommodate a VBCU and five VMEbus controllers. This unit is configured in a combination chassis that contains a VMEbus and a five slot Multibus. The Marketing number for this chassis is VBS-005.

The VMEbuses described above are used in C1, C2, and C3 class machines. A VBS-004 VMEbus is mounted in the cabinet of the Javelin Junior. The VBS-004 chassis is removed from the rack-mount drawer and fixed in the bottom of the Javelin Junior. Access to the chassis is from the top by removing a panel that covers the chassis under normal operations.

The VMEbus in Convex systems is modified to Convex specifications. Each slot appears to be adjacent to the VBCU. Bus request 3 and bus grant 3 are assigned to each slot. The slots in the VMEbus share address, data, and most of the control bus, but are not daisy chained to each other as in the standard VMEbus. There is no priority established by the slot position in the VMEbus.

Convex uses VMEbus controllers in the 6U or 9U format. The 6U boards are 6Ux160. The 9U boards are 9Ux160. In comparision, a SUN VMEbus board is 9Ux240. In the Convex VMEbus, controllers must not use more than 6 amps of the 5 volts DC since there are only two connectors provided on the backplane to supply power.

VMEbus Master

Controllers need to be capable of VMEbus Mastership and provide DMA capabilities to achieve high data transfer rates. Non Bus Master VMEbus controllers will work but the performance will be lower. Also, any Data Strobe inter-cycle times must be kept to a minimum to achieve maximum performance through the cable interface. Performance will approach 10 Mega-Bytes/sec on a single port and about 6 MB/sec/port if both VME chassis are operational. The total agregate throughput for one VIOP will not exceed 12MB/sec.

Block Mode Transfers

Block mode VMEbus transfers (and the associated address modifiers 3B, 3F, 0B, and 0F) are not supported. Unaligned data transfers are also not supported. Convex VMEbus transfers will be either single byte, as in byte zero, byte one, etc.. or word zero and word one, or longword. Convex VMEbuses will not do transfers of multiple bytes that start on odd byte addresses.

VMEbus Addresses

The Convex VMEbus supports three address modes. The modes are short, standard, and extended. The short address is used as the base address of the controllers installed in the VMEbus. The address of a controller is actually the first address of the control and status register (CSR), on the controller. The number of addresses associated with the CSR differ with each Vendor's VMEbus controller. The CSR address is driven from the VIOP to the VMEbus only, it does not go up the bus to the VIOP.

The standard address is used when data is being transferred between the VIOP cache and the VMEbus. The standard address is contained in the map register during times when data is in transit between memory and the device.

The extended address is supported by the VIOP.

VBCU Registers

The VBCU internal registers occupy the uppermost 16 bytes of the 65536 byte VMEbus short address space. Within the 68020 address space, the short address space for VME chassis 0 is located from C00000 through C0FFFF hex, and the short address space for VME chassis 1 is located from C10000 through C1FFFF hex. The VBCU internal registers for chassis 0 and 1 are thus located from C0FFF0 through C0FFFF and from C1FFF0 through C1FFFF respectively in the 68020 address space.

The 68020 can also access VMEbus standard and extended address spaces. No portions of these address spaces are reserved, the entire range is available to the 68020. There are physically only 22 address bits used on the cable from the VIOP to the VME chassis. VMEbus address bits A<22..31> are generated according to the contents of registers located on the VBCU. These registers are described in the Internal Register Specification for the VBCU. The locations of VBCU registers and of VMEbus short, standard, and extended address spaces within the 68020 memory map are summarized in the following table:

VMEbus Addresses in 68020 Memory Map		
Address Space	Chassis 0	Chassis 1
VBCU	C0FFF0 - C0FFFF	C1FFF0 - C1FFFF
Short	C00000 - C0FFEF	C10000 - C1FFEF
Standard	1000000 - 13FFFFF	1400000 - 17FFFFF
Extended	1800000 - 1BFFFFF	1C00000 - 1FFFFFF

Address Modifiers

The address modifiers used by the VMEbus to select either short, standard, or extended addressing, are generated on the VBCU whenever the 68020 drives an address on the bus.

When the controller on the VMEbus starts a DMA, that controller must drive the address modifier to the VBCU. Convex VMEbus ONLY support the standard VMEbus address modifiers listed below:

1) When the 68020 Micro Processor on the VIOP is VMEbus Master, the VBCU generates address modifiers as follows:

   - 16-bit address: 2D (Short Supervisory Access)
   - 24-bit address: 3D (Standard Supervisory Data Access)
   - 32-bit address: 0D (Extended Supervisory Data Access)

2) When a controller is VMEbus Master and wants to access memory through the VIOP cache, we respond to address modifiers as follows:

   - 24-bit address: 3D (Standard Supervisory Data Access)
   - 24-bit address: 39 (Standard Non-Priviledged Data Access)

   - 32-bit address: 0D (Extended Supervisory Data Access)
   - 32-bit address: 09 (Extended Non-Priviledged Data Access)

Standard and Extended Address Bits

The VMEbus address bits A22-A23 must match the address in the map register in the VBCU for 24-bit Addresses, and bits A22-A31 must match the contents of the VBCU map register for 32-bit Addresses. This is essentially a software issue. The DMA transfer address has to be set up consistent with the VBCU address map registers. The address bits of the VBCU map register for A22-A31 are set to zero at boot time.

VMEbus Controllers

VMEbus controllers that will function in the Convex supported VMEbus are of three basic groups, Convex Supported, Special Systems Supported and User Supported.

- Convex Supported - The Convex VMEbus controllers are a mixture of VMEbus controllers designed and manufactured by Convex and VMEbus controllers manufactured by third party vendors. Convex develops and supports device drivers for the operating system and diagnostics for all Convex supported VMEbus controllers. In addition, Convex integrates and qualifies all VMEbus controllers that are Convex supported. Configuration data are supplied for each Convex supported controller. Control and status register (CSR) address allocations for each controller type are defined in the 32K of addresses contained in the short address space. Although the standard address space is supported and used for DMA, by convention, Convex assigns the CSR address of each controller to the VMEbus short address space.

- Special Systems, a development group at Convex, designs and manufactures products specifically for a requesting Customer. Although these products are supported by Convex, they are not considered to be standard product. Special Systems designs and builds to Customer requirements. For that reasons, standard guidelines for product development and release may not be observed. Address space in the short address space is allocated for Specials Systems.

- User Supported - The User supported VMEbus controllers are not part of the standard Convex product line. For this reason the User must develop the driver and diagnostics to support any VMEbus controller used by the customer but not supported by Convex. Address space in the short address space is reserved by Convex for use by the end User.

REFERENCES:

The following provides:

- A list of useful reference documents

- A list of Convex supported VMEbus controllers by their type and application, part number, marketing number, software mnemonic, required number of window space in main memory and VIOP memory

- A list of recommended addresses and interrupt levels for Convex supported VMEbus configurations

- A recommended list of reserved address spaces for end-users to add non-supported devices

- Tables, guidelines and examples to follow when installing or using I/O controllers in a Convex VMEbus configuration.

VMEbus Reference Documents

The table below list reference documents by Manual, Convex Part Number, and Market Number. These documents are included in the documentation that comes with a Convex system. They may also be ordered from stock.

Table -1: VMEbus Reference Documents

CONVEX VME REFERENCE DOCUMENTATION		
Manual	Part Number	Marketing Number
VIOP/VBCU Service Guide	081-000030-201	DHW-051
Guide to Writing Device Drivers	710-001830-203	
Input/Output Bus(PBUS) Functional Spec	081-000109-000	
IOP Reference Manual	900-000287-000	
VIOP Difference Manual	081-012730-000	
VBCU Internal Register Specification	081-013430-000	
Convex 68000 Tools User's Guide	740-001930-200	
ConvexOS VIOP Device Driver Memory Usage	710-003430-022	
SMD Disk Controller Service Guide	081-000730-200	DHW-052
ESDI Disk Controller Service Guide	081-000130-201	DHW-053
Tape Controller Service Guide	081-001330-200	DHW-054
Ethernet Controller Service Guide	081-000630-201	DHW-055
Ethernet Controller Service Guide Addendum	081-000199-202	DHW-055a
UltraNet Interface Service Guide	081-001630-000	DHW-049
FDDI Interface Service Guide	081-012630-000	DHW-275
Async/Printer Controller User's Guide	081-001430-000	DHW-057
SCSI Host Adapter Service Guide	081-006830-000	DHW-060
Versatec Print/Plot Interface Installation	081-007830-000	
Versatec Print/Plot Software Guide	081-007930-000	
Versatec Print/Plot Device Driver Release Notes	081-009630-002	
VMEbus DMA Interface (Uses same manual as below)	081-005130-000	
Tape Recorder/Reproducer Interface (Same as above)	081-005130-000	
Reconfigurable VME Interface	(Not Published Yet)	

Standard Convex Controllers

The Convex VMEbus and Unix I/O device drivers are preconfigured to work with several specific controllers. The following table provides a list of these controllers along with Part Number, Marketing Number, and the correct entry for the Service Processor's configuration file, /ioconfig.

Table -2: Convex Standard VMEbus Controllers

Controller	Part Number	Marketing No	/ioconfig No
SMD Disk	220-000010-200	DKC-204	DKC-204
ESDI Disk	220-000011-200	DKC-203	DKC-203
Ethernet	220-000012-200	ETH-202	LAN-007
Tape	410-001152-200	MTC-201	MTC-201
UltraNet	211-000108-200	ULT-201	LAN-202
FDDI	220-000021-200	LAN-208	LAN-208
Async/Printer	410-001193-200	ACM-201	ACM-201
SCSI Host Adpt	220-000019-200	MTC-202	MTC-202
Hyperchannel	211-000110-200	HYP-201	LAN-204
Versatek Plotter Intf	211-000111-200	SP-152	
RVI board	424-001232-200	HSR-002	(Un-Listed)

Controller Configurations

Convex offers a variety of standard VMEbus controllers. Each controller used in Convex's standard product line has Service Guides that give all of the information required to install and service the device.

The table below list the controllers by name, recommended address, and recommended interrupt. In addition the number of 4096 byte windows required to operate the controller are listed.

- The name of the controller identifies the type, ie.. Ciprico Rimfire SCSI Host Adapter, gives a fair description of what the board does. It must be noted that most controllers purchased from third party vendors are modified during the evaluation and integration process. For this reason, users that purchase controllers directly from the vendor must specify the Convex part number.

- Most VMEbus controllers installed in Convex system are flexible in the address range. Some are very limited, ie. although the short address space uses the address bits 15..0, some controller are configurable to only four addresses out of the possible 32K addresses. For that reason the addresses are scattered over the range of possible addresses. The end user integrating a non-standard controller must be aware of the assigned addresses to avoid future conflicts.

- The interrupts assigned to controllers in the Convex VMEbus system are based on the type and number of the controllers installed. Character based controllers are given the higher priority interrupts and block based devices are given the lower priority interrupts. This is certainly flexible and in cases where the installation requires, any available interrupt may be used. At the same time, some controllers have the interrupt hard-wired and some are programmable. Care must be taken not to assigned conflicting interrupts.

- When a controller is assigned to a VMEbus, there are a specific number of windows allowed for the device driver for that specific type of controller. The cache in the VIOP is four megabytes of fast storage in the VIOP 16MB address space. This cache is divided into 4096 byte pages with 1024 map registers to point to each page. The page represents a window into main memory. The main memory address is stored in the map register and selects the page in the cache that will act as the buffer for the data. The user must be aware of the restriction this places on the system to avoid conflicts and enhance perfomance. There are two types of memory allocations made for each

controller:

1) Each VIOP is allowed 1024 windows to main memory. Each window is 4096 bytes. The VIOP requires a small number of windows to communicate with the system, and each controller requires a number of windows. If the controller being installed requires 512 windows, only one controller of that type could be installed in the two VMEbuses attached to that VIOP. In most applications, the required number of windows is much less than 512 per controller. The total number of windows required for all controllers installed in the two VMEbuses, plus the windows required for the VIOP must not exceed 1024.

2) For each Controller there are pages required in the VIOP local memory. These are defined for the standard controllers. Integrators must observed the allocation of memory pages in the VIOP for the device driver. EGOS and all of the the device driver must reside within the total 512K of local ram.

Table -3: Standard Controllers and Addresses

Controller Type	CSR Address	Int No.	Memory Windows	VIOP Pages
IKON Versatec Hardcopy Interface	0x100	4	1	1
IKON Versatec Hardcopy Interface	0x140	3	1	1
IKON Versatec Hardcopy Interface	0x180	2	1	1
IKON Versatec Hardcopy Interface	0x1A0	1	1	1
Interphase SMD Disk	0x800	1	35	4
Interphase SMD Disk	0xC00	2	35	4
Interphase SMD Disk	0xA00	3	35	4
Interphase SMD Disk	0xE00	4	35	4
Interphase ESDI Disk	0x800	1	35	4
Interphase ESDI Disk	0xC00	2	35	4
Interphase ESDI Disk	0xA00	3	35	4
Interphase ESDI Disk	0xE00	4	35	4
Convex VBTC Tape	0x1000	1	34	14
Convex VBTC Tape	0x1040	2	34	14
Convex VBTC Tape	0x1080	3	34	14
Convex VBTC Tape	0x10C0	4	34	14
Ultra Network Interface	0x7740	7	512	54
Convex VASYNC/Printer	0x1100	7	18	14
Convex VASYNC/Printer	0x1120	6	18	14
IKON Hyperchannel	0x2000	1	1	1
IKON Hyperchannel	0x2020	2	1	1
IKON Hyperchannel	0x2040	3	1	1
IKON Hyperchannel	0x2060	4	1	1
IKON Hyperchannel	0x2080	5	1	1
Interphase FDDI Network	0x6000	1	39	15
Interphase FDDI Network	0x6200	2	39	15
Ciprico RF3516 SCSI Host Adapter	0xEE00	4	135	15
Ciprico RF3516 SCSI Host Adapter	0xEC00	3	135	15
Ciprico RF3516 SCSI Host Adapter	0xEA00	2	135	15
Ciprico RF3516 SCSI Host Adapter	0xE600	1	135	15
Federal Technology EXOS 302 Ethernet	FD80	7	83	11
Federal Technology EXOS 302 Ethernet	FE00	6	83	11
Federal Technology EXOS 302 Ethernet	FE80	5	83	11
Federal Technology EXOS 302 Ethernet	FF00	4	83	11
Convex Reconfigurable VME Interface	0x9000	4	18	5

Configuration Restrictions

Throughput in the Convex VMEbus peaks out at a sustained rate of about 10MB/sec when one bus is being used. If both buses are in use the maximum throughput approaches 12MB/sec. Most VMEbus controllers will burst data at rates exceeding 30MB/sec. In the four edge asynchronous handshake used in the Convex VMEbus, the bus will not get errors due to overrun, but, it is possible to install multiple VMEbus controllers with high speed devices attached, in a configuration so that the performance would be affected. An example is the Interphase SMD Disk Controller, the 4200. The 4200 will allow a device to transfer data up to 3MB/second from the device to the controller. The data is then burst over the VMEbus at rates up to 10MB/sec if only one bus is in operation and 6BM/sec if both buses are in operation. Four of the 4200 disk controllers installed in one VMEbus, configured with a striped partition where data is coming from the disk drives as an agregrate 12MB/sec, would cause the disk drives to miss revolutions waiting to be serviced by the VMEbus. This would have the effect of decreasing overall system performance. The data would not be lost but the disk drives would miss revolutions and performance would decrease. To prevent bus problems of this type, make sure the configuraton allows for transfer rates, correct number of windows, etc....

Interrupts

VMEbus Interrupt Guidelines

There are only seven available interrupt levels per VMEbus. In a standard product configuration document for Convex controllers, at least two addresses and two interrupts are listed. When there are more than two controllers in any one VMEbus, the installer must reference the technical manual for the specific controller if the Configurator does not list all interrupt options. Conflicts will occur if two controllers are located at the same interrupt level. Each Controller in a Convex VMEbus must be assigned to a UNIQUE interrupt level.

There is also a correlation between an interrupt number and its interrupt priority level. Priority levels begin at one (for the lowest priority device) and go to seven (for the highest priority device). Controllers of the same types maintain the same relative priorities. Character mode devices are assigned a higher priority than block mode devices. For example, the VASYNC controller has a higher priority than the 4200 SMD Disk Controller.

Some VMEbus controllers allow programmable interrupts. That is, the interrupt can be set by the host at initialization. The information to tell the host which interrupt to assign is in the configuration file on the Service Processor. The file is ioconfig in the root directory. In this case, you may assign any interrupt number that is not in use. Observe the priority of the interrupt when assigning the number.

Interrupt Assignment Example:

The first VASYNC should be assigned to the highest priority, the second VASYNC in the same VMEbus is assigned the next highest priority. VASYNC controllers can transfer data at 38.4 baud. This will allow only two VASYNC controllers to be installed per VMEbus. The interrupt should be 7 or 6 since the VASYNC is a character mode device. Character mode means it will be doing lots of short fast transfers.

The ESDI Disk controller, the ESDI 4201, can burst data up to 10MB/sec but the input to the controller from the disk drives is only a maximum of 2.46MB/sec. This relatively low transfer rate allows four ESDI disk controllers to be installed in one VMEbus when a single VMEbus is attached to a VIOP. When two VMEbuses are attached to one VIOP the configuration should limit the number of controllers to two/VMEbus. Because the ESDI Disk Controllers is a block mode device, the interrupt assigned is a lower priority, between 1 and 4.

Table -4: Convex VMEbus Recommended Interrupt Priorties

Product Number	controller Type	Interrupt Level
SMD Disk	DKC-204	1 to 3
ESDI Disk	DKC-203	1 to 4
Ethernet	ETH-202	7 to 4
VTape	MTC-201	3 to 6
UltraNet	ULT-201	7 to 4
VAsync/Printer	ACM-201	7 to 6
SCSI Host Adpt	MTC-202	1 to 4
FDDI	LAN-208	7 to 4
Hyperchannel	HYP-201	7 to 4
VPlotter	VPC-001	5
RVI board	HSR-002	4

Convex VMEbus Assigned Address List

The following list defines CSR addresses for Convex Controllers and allows a space for Special Systems, Customers,and Convex available space.

Address	Controllers	Assigned Vendor and Controller Type
0x0100	Controller 1	IKON Plotter Interface
0x0140	TO 0x07ff	Available address space
0x0800	Controller 1	INTERPHASE 4201 ESDI DISK CONTROLLER
0x0a00	Controller 2	INTERPHASE 4201 ESDI DISK CONTROLLER
0x0c00	Controller 3	INTERPHASE 4201 ESDI DISK CONTROLLER
0x0e00	Controller 4	INTERPHASE 4201 ESDI DISK CONTROLLER
0x0800	Controller 1	Interphase 4200 SMD DISK CONTROLLERS
0x0a00	Controller 2	Interphase 4200 SMD DISK CONTROLLERS
0x0c00	Controller 3	Interphase 4200 SMD DISK CONTROLLERS
0x0e00	TO 0x0fff	Available address space
0x1000	Controller 1	Convex Vtape STK Interface Tape Controller
0x1040	Controller 2	Convex Vtape STK Interface Tape Controller
0x1080	Controller 3	Convex Vtape STK Interface Tape Controller
0x10c0	Controller 4	Convex Vtape STK Interface Tape Controller
0x10e0	TO 0x10ff	Available address space
0x1100	Controller 1	Convex Vasync Communication/printer Controller
0x1120	Controller 1	Convex Vasync Communication/printer Controller
0x1140	TO 5FFF	Available address space
0x6000	Controller 1	FDDI Interphase 4211
0x6200	Controller 2	FDDI Interphase 4211
0x6a00	TO 0x773f	Available address space
0x7740	Controller 1	ULTRANET HOST ADAPTER NETWORK CONTROLLER
0x8000	To 0x8fc0	Available address space
0x9000	Controller 1	CONVEX Reconfigurable VME Interface board
0x9020	TO 0xafff	CONVEX RESERVED for SPECIAL SYSTEMS
0xb000	TO 0xbfff	Reserved for Customer-Defined
0xc000	TO 0xdfff	Reserved for Real Time Products
0xe600	Controller 4	SCSI Host Adapter Ciprico RimFire 3516
0xea00	Controller 3	SCSI Host Adapter Ciprico RimFire 3516
0xec00	Controller 2	SCSI Host Adapter Ciprico RimFire 3516
0xee00	Controller 1	SCSI Host Adapter Ciprico RimFire 3516
0xf000	TO 0xfd7f	Available address space
0xfd80	Controller 1	EXOS 302 Ethernet Controllers Federal Technology
0xfe00	Controller 2	EXOS 302 Ethernet Controllers Federal Technology
0xfe80	Controller 3	EXOS 302 Ethernet Controllers Federal Technology
0xff00	Controller 4	EXOS 302 Ethernet Controllers Federal Technology