VCT-V, VCU-V, VCD-V

GRAPHICS BOARDS
USER'S MANUAL

Peritek Corporation
5550 Redwood Road
Oakland, CA 94619
(510) 531-6500
FAX (510) 530-8563

email: support@peritek.com

Release 2.3
April 19, 1995

Applies to:
VCT-V FAB REV 4 and up
VCU-V FAB REV 4 and up
VCD-V FAB REV 2 and up

Table of Contents

Introduction

This introductory chapter contains information about the organization of this manual, how to get technical support, and the typographical conventions used throughout the manual.

The Organization of This Manual

This manual provides information about how to configure, install, and program the Peritek 34020-based VMEbus graphics controllers. Products covered include the VCU-V ultra-high resolution controller, the VCT-V 24-bit true-color controller, and the VCD-V analog/digital controller. The boards can be covered in one manual because their feature set is largely the same, and the software is identical.

This manual is broken down into six chapters:

Chapter 1: Overview of the Peritek graphics boards
Chapter 2: Installing Peritek graphics boards
Chapter 3: Summary of Peritek's Software Products
Chapter 4: Theory of Operation
Chapter 5: Programming On-board Devices and Memories
Chapter 6: Troubleshooting

Chapter 1 provides interesting background material about Peritek graphics boards. Understanding the information in the chapter, however, is not essential for the hardware or software installation.

If you want to perform the installation as quickly as possible, start with Chapter 2. If you have problems installing the hardware, refer to Chapter 6 for help.

Getting Help

This installation manual gives specific steps to take to install your Peritek graphics board. There are, however, variables specific to your computer configuration and monitor that this manual cannot address. Normally, the default values given in this manual will work. If you have trouble installing or configuring your system, first read Chapter 6, "Troubleshooting". If this information does not enable you to solve your problems, do one of the following:

1) call Peritek technical support at **(510) 531-6500**,

2) fax your questions to **(510) 530-8563**,

3) or send E-mail to ***support@peritek.com.***

If your problem is monitor related, Peritek technical support will need detailed information about your monitor.

Manual Revisions

Revision 2.0	January 11, 1995	First Word for Windows 2.0 Master
Revision 2.1	March 27, 1995	Compensate for WFW bug which yielded incorrect Table of Contents and Index page numbers, fix minor factual errors in Chapter 1.
Revision 2.2	April 6, 1995	Fixed cursor address error in Ch. 5.
Revision 2.3	April 19, 1995	Fixed some format errors. Changed the pagination style.

Notices

Information contained in this manual is disclosed in confidence and may not be duplicated in full or in part by any person without prior approval of Peritek Corporation. Its sole purpose is to provide the user with adequately detailed documentation to effectively install and operate the equipment supplied. The use of this document for any other purpose is specifically prohibited.

The information in this document is subject to change without notice. The specifications of the VCU-V, VCD-V, VCT-V, and other components described in this manual are subject to change without notice. Although it regrets them, Peritek Corporation assumes no responsibility for any errors or omissions that may occur in this manual.

Peritek Corporation assumes no responsibility for the use or reliability of software or hardware that is not supplied by Peritek, or which has not been installed in accordance with this manual.

PX Windows and **Peritek** are trademarks of Peritek Corporation. The products, HP-UX, OS-9, pSOSystem, SunOS, OpenVMS, VAXeln, Ultrix, VMEexec, V/68 SVR3, V/88 SVR3, and VxWorks are registered trademarks of Hewlett-Packard, Microware, Integrated Systems, Sun, DEC, DEC, DEC, Motorola, Motorola, Motorola, and Wind River, respectively.

Conventions Used In This Manual

The following list summarizes the conventions used throughout this manual.

`Code fragments`	Code fragments, file, directory or path names and user/computer dialogs in the manual are presented in the `courier` typeface.
Commands or program names	Commands, or the names of executable programs, except those in code fragments, are in **bold**.
System prompts and commands	Commands in code fragments are preceded by the system prompt, a percentage sign (%), the standard prompt in UNIX's C shell, a dollar sign ($), the OS-9 prompt, or the hash-mark (#), the standard UNIX prompt for the Super-User.
Note	**Note** boxes contain information either specific to one or more platforms, or interesting, background information that is not essential to the installation.
Caution	**Caution** boxes warn you about actions that can cause damage to your computer or its software.
Warning!	**Warning!** boxes warn you about actions that can cause bodily or emotional harm.
Keyboard usage	**<CR>** stands for the key on your keyboard labeled “RETURN” or “ENTER”

Chapter 1
General Information

1.1 Introduction

This chapter provides an overview of the VCT-V, VCD-V, and VCU-V graphics controllers. Additional sections contain a bibliography, specifications, monitor requirements, and common configurations.

This is summary information, and is not critical to the one who wishes to press on to the installation procedures, which are contained in Chapter 2.

1.2 Functional Description

The Peritek VCD-V, VCT-V and VCU-V are based on the second generation TMS 34020 32-bit Graphics System Processor (GSP). The boards offer a high degree of on-board intelligence and functionality, as well as a straightforward frame buffer interface.

The boards are differentiated chiefly by the bits/pixel of the primary display memory and the video output sections. The VCD-V and VCU-V have 8 bits/pixel in the primary plane and the VCT-V has 24-bits. The VCU-V and VCT-V have only analog RGB outputs and the VCD-V offers both analog and digital.

The common feature set of the VCU-V, VCT-V and VCD-V includes:

- 40 MHz 34020
- Optional 34082 Floating Point Unit (FPU)
- 4 RS-232 serial I/O ports
- PC Keyboard
- 4 Kb serial EEPROM
- up to 2 MB autoboot Flash PROM
- up to 32 MB 34020 memory
- Optional multiple display pages
- Hardware pan, zoom, and scroll
- Hardware bitmapped cursors,
- 4 bit overlay
- SIMMs for display and 34020 memory
- PLL controlled pixel clock
- genlock support for system wide synchronization
- analog RGB video output
- Up to 72 Hz display refresh rate
- Optional 32-bit High Speed Data port
- Optional autoboot simple console terminal emulator
- BiCMOS bus transceivers and AMD MACH FPGAs (for low power consumption)
- Single 6U VMEbus board
- Graphics Subroutine Package
- X11R6 X Window System Server

Special Features of the VCU-V

The VCU-V has an 8-bit primary screen, an optional 8-bit SCSI port, and up to two pages of 1600 x 1280 display. Programmable screen resolution ranges from 640 x 480 pixels up to better than 1600 x 1280 with refresh rates between 60 and 72 Hz vertical and 31 to 85 KHz horizontal refresh rates non-interlaced.

Special Features of the VCT-V

The VCT-V supports displays up to 1280 x 1024, has a true color (24-bit) primary screen, an optional 8-bit SCSI port, and up to four pages of 1280 x 1024 display. Programmable screen resolution ranges from 640 x 480 pixels up to 1280 x 1024 with refresh rates between 30 and 72 Hz vertical and 15.7 to 73 KHz horizontal refresh rates, non-interlaced or non-interlaced, including NTSC sync compatible 640 x 483.

Special Features of the VCD-V

The VCD-V supports displays up to 1280 x 1024, has an 8-bit primary screen, both analog and digital (flat panel) outputs (digital is limitied to 1024 x 768), simultaneous analog and dital operation (with VGA timing compatible panels) and up to two pages of 1280 x 1024 display. Programmable screen resolution ranges from 640 x 480 pixels up to 1280 x 1024 with refresh rates between 30 and 72 Hz vertical and 15.7 to 73 KHz horizontal refresh rates, non-interlaced or non-interlaced, including NTSC sync compatible 640 x 483.

The VCT and VCU are also available in DEC compatible Q-Bus and TURBOchannel versions.

TMS 34020 Graphics Processor

The TMS 34020 is a CMOS 32-bit processor with hardware support for graphics operations such as PIXBLT and curve-drawing algorithms. Included is a complete set of general purpose instructions with addressing modes tuned to high level languages. In addition to addressing a 512 MB external memory range, the 34020 contains 30 general purpose 32-bit registers, stack pointer, and a 512 byte LRU instruction cache. On chip functions include 64 programmable registers used for CRT timing, I/O control, and instruction parameters. The 34020 can receive interrupts from the the VMEbus, serial I/O, and SCSI.

The 34020 mediates all host accesses to display and processor memory and control registers through a byte addressable 32 bit interface port. Bus transceivers between the 34020 bus and VMEbus support D16 and D32 data transfers for the VMEbus.

The 34020 features single-cycle execution of general purpose instructions and most common integer arithmetic and Boolean operations from instruction cache. A 32-bit barrel shifter supports single cycle shift and rotation for 1 to 32 bits.

The 34020 graphics processing hardware supports pixel and pixel-array processing. It incorporates two and three operand raster operations with Boolean and arithmetic operations, XY addressing, window clipping and checking, 1 to n bits/pixel transforms, transparency, and plane masking. Operations on single pixels (PIXT instruction) or two-dimensional arrays (PIXBLT) are supported.

TMS 34082 Floating Point Coprocessor

For floating point intensive applications, a socket is provided for a 34082 FPU coprocessor. It conforms to the IEEE floating point standard 754-1985 for binary floating point single or double precision addition, subtraction, multiplication, division, square root, and comparison. In addition, it offers 32-bit integer arithmetic, logical comparisons, and shifts.

Complex operations for graphics support include: matrix operations (1 x 3, 3 x 3, 1 x 4, and 4 x 4), backface testing, polygon elimination and clipping, viewport scaling and conversion, 2D and 3D linear interpolation, 2D window compare, 3D volume compare, 2 plane clipping (X, Y, Z), 2 plane color clipping (R, G, B, I), 2D and 3D cubic splines, 3 x 3 convolution, vector operations (add, subtract, dot and cross products, magnitude, scaling, normalization and reflection), polynomial expansion, multiply/accumulate, and 1D and 2D min/max.

Video RAM

The display memories use advanced 2 Mbit (256K x 8) 2-port Video RAM (VRAM) technology, which gives approximately 95% memory availability to the 34020 and host processors. A writemask register supports write protection of bit planes.

The 34020 supports the VRAM accelerated functions such a block write and fill with special VFILL and VBLT instructions. These can be used to quickly replicate one and two dimensional patterns in memory, at up to 16 times the single pixel rate. On the VCD-V and VCU-V, up to sixteen 8-bit pixels can written in each 100 ns page mode cycle, resulting in a 160 Mpixel/sec VFILL time. The VCT-V, which has 32-bit pixels, has a 40 Mpixel/sec VFILL time.

34020 Processor Memory (DRAM and PROM)

The 34020 has its own "system" memory, which is independent of the video memory. However, it does share a common address space with the display memory and can thus be used for program store or off-screen display data. The standard size is 1 MB of 0 wait state DRAM, and is expandable to 32 MB.

There are four 32-pin PLCC sockets which support up to 2 MB (using 29F020 devices) of 0 wait state Flash PROM. Jumpers can be installed which cause the 34020 to automatically start executing from PROM on power-up. An additional 512 byte serial EEPROM can be installed which can be used by an PROM-based program to store information necessary at power-up (such as initialization data). PROM sets can be ordered from Peritek which include a simple console terminal emulator combined with the graphics subroutine package or X11R6 X Windows server.

Display Features

All boards support binary vertical zoom (1, 2, 4, 8, 16, 32), horizontal zoom (except some VCD-Vs), multi-pixel horizontal pan and vertical smooth scroll.

For the VCD-V and VCU-V, the display memory data is directed to the analog monitor via a Brooktree RAMDAC color map control chip which provides a programmable 24 bit wide color map (8 bits each red, green, and blue). The 8-bit pixel is used as an index into the lookup table, giving 256 colors out of a palette of 16.7 Million. A two bit cursor with a 64 x 64 x 2 bit map function is also included on chip.

For the VCT-V, the display memory data is directed to the analog monitor via a Brooktree BT463 color map control chip which provides a programmable 24 bit wide color map (8 bits each red, green, and blue). The 24-bit pixel is used as an index into the lookup table, giving a full 16.7 Million colors. A two bit cursor with a 64 x 64 x 2 bit map function is also included, but incorporated in separate BT431 cursor chips.

For all boards, additional color map entries are provided for the overlay screen and cursors. Any plane can be blanked or blinked. The analog Red, Green, and Blue signals from the RAMDAC are connected to a monitor such as the Peritek CVM-19/E-7.

On the VCD-V (only) the display memory data is also directed to a proprietary Digital LookUp Table (DLUT) which provides a programmable 8-bit wide map. Depending on how the VCD-V is configured, 1, 4, or 8 bit monochrome or 8-bit color (3 bits red, 3 bits green, and 2 bits blue) can be supported. **The VCL-V, available in May 1995, will support expanded 12 to 24 bit color map options.** The eight bit pixel is used as an index into the lookup table, giving 256 colors out of a palette of 16.7 Million. A two bit cursor with a 64 x 64 x 2 bit map function is also included. Additional color map entries are provided for the overlay screen and cursors. Contact Peritek for information on what panels are supported.

VMEbus Interface

The VMEbus host interface accesses Peritek graphics board memory and on-board devices through four control registers and a 1 KB line buffer which are located in VMEbus A16 space. The line buffer can also appear in A24 space.

The CSR contains device interrupt enables, line buffer response enable, and 34020 hardware reset. The LAR is a 16 bit register which maps a portion of the address space of the graphics board into a 1 KB line buffer. Two additional registers include programmable line buffer address, interrupt vector address and programmable extended address (A32) decoder. Access to the board through the A16 space provides a "lowest common denominator" access mode which allows the board to be compatible with any host CPU. In an A16 VMEbus system it is necessary to "window" into on-board memory because it is so large (maximum memory capacity on the board is more than 48 MB!) The 1 KB window is actually an efficient way of doing this.

The graphics boards also have an 64 MB window in A32 VMEbus address space which allows direct access to all on-board memory.

VMEbus D32 block transfers are supported for A16/A24 and A32 address spaces, which allows up to 256 bytes to be transferred at high speed over the VMEbus. Another performance feature for the boards is a hardware byte swapper. When enabled, four 1KB buffers are mapped to the board which provide unswapped, byte, word, and long swaps, respectively.

The board has a VMEbus interrupt controller which supports a vectored interrupt from the 34020.

Peripheral Support

The graphics board has four asynchronous RS-232 data-leads-only serial I/O ports. Ordinarily, the board is configured to support 3 channels (serial mouse/trackball, LK401-type keyboard, and console port), with the I/O lines for the fourth port being "stolen" to support RTS/CTS on the console port. Each port can be programmed separately for transmit and receive baud rates up to 38.4 Kb. Each receive buffer is quadruply buffered to minimize the possibility of data overrun. Each channel has an internal loopback mode for testing.

An 8242PC keyboard controller gives a PC-compatible keyboard port and a PS/2 mini-DIN connector is used.

The VCU-V and VCT-V also have an optional 8 bit Small Computer Systems Interface (SCSI) peripheral port.

All three designs can be supplied with a 32-bit High Speed Port (HSP) which allows 32-bit data to read or written directly by the 34020 to/from system or graphics memory.

The Peritek graphics boards are highly configurable for special requirements. In order to ensure optimum performance at the lowest OEM cost, please contact Peritek for quotes for customized feature sets.

1.3 Additional References

Peritek documentation includes User's Manuals, Graphics Subroutine Package Manual, and Peritek PX Windows Server Installation and User's Guide. Due to lack of demand, data sheet extracts are no longer included as Appendices to the manual. They are available upon request. The manufacturer sources of this information are:

TMS 34020 User's Guide Order # SPVU019 **TMS 340** Math/Graphics Function Library Order # SPVU006 **TMS 340** Assembly Language User's Guide Order # SPVU004 **TMS 340** Family Code Generation Tools Order # SPVU020B	**Texas Instruments** Customer Response Center 1-800-232-3200
SCN2681 Dual Asynchronous Receiver-Transmitter (DUART) Data Sheet Signetics Microprocessor Data Manual 1986, pages 2-189 to 2-208	**Philips Semiconductors** 811 E. Arques Avenue Sunnyvale, CA 94088-3409 800-234-7381
BTxxx Product Descriptions Product Data Book, 4th Edition	**Brooktree Corporation** 9950 Barnes Canyon Road San Diego, CA 92121 619-452-7580
NCR5380 SCSI Processor 1988 Standard Products Data Book pages 75 - 115	**NCR Microelectronics** 1635 Aeroplaza Drive Colorado Springs, CO 80916 1-800-525-2252
VMEbus Specification	**VITA** 10229 N. Scottsdale Road Suite B Scottsdale, AZ 85253 (602) 951-8866

Graphics Textbooks

Fundamentals of Interactive Computer Graphics Addision Wesley, 1982.	Foley and Van Dam
Principles of Interactive Computer Graphics McGraw-Hill, 1979	Newman and Sproull

1.4 General Specifications

Graphics Processor: 40 MHz TMS 34020 Graphics System Processor has a complete instruction set 32-bit CPU, vector and pix-blt functions, and programmable video timing.

Floating Point Unit: A socket is provided on the board for the companion 40 MHz TMS 34082 Floating-Point Unit (FPU) coprocessor, which can accelerate floating-point intensive operations by an order of magnitude.

Non-Display Memory: Memory is 1 MB of 32-bits/word, byte addressable, no-wait state, dynamic RAM. This memory is in the same memory space as the display memory, so it can hold program store and off screen display data. It is expandable in steps of 4, 8, 16 and 32 MB. A minimum of 4 MB is required for PX Windows.

PROM Memory: Four 8-bit Flash PROMs support no-wait state firmware storage of up to 2 MB (total) of 32-bit wide permanent storage. A user jumper allows the board to auto-start from EPROM.

Peritek can supply dumb terminal emulation (PTERM), Graphics Subroutine Package (CnP), and Peritek's PX Windows X11R6 server in PROM. Possible combinations are PTERM alone, CnP alone, PX Windows alone, or a combination of PTERM and ***either*** CnP or PX Windows.

An optional 4 Kbit (512 byte) serial Electrically Eraseable Programmable Read Only Memory (EEPROM), programmed via DUART 0, supplies non-volatile read-mostly memory for an EPROM-based application to retain some changeable data during power down. Software support for this function is under development.

Video Display:

On the VCD-V and VCU-V, the display word size is 32 bits, and is divided into 1 byte per pixel. Overlay is allocated a byte per pixel, but only the low 4 bits are valid. The lookup table (LUT) resolves the display priority between the primary, overlay, and cursor (last through first, respectively) screens.

On the VCT-V, the display word size and pixel size is 32 bits. Bits 0-7 are Red, bits 8-15 are green, bits 16-23 are blue, bits 24-27 are overlay, bits 28-29 are for window type table, and bits 30-31 are read/write but not used. The BT463 RAMDAC lookup table resolves the display priority between the primary, overlay, and cursor (last through first, respectively) screens.

Display Memory:

The basic display memory size for the VCD-V is 1 MB of 32-bits/word, byte addressable, no-wait state, dual-port Video RAM. This provides 1024 x 1024 x 8 bit/pixel primary screen. Overlay may be specified at order time. The display memory is expandable to 8 MB. The /2M option can give either 1280 x 1024 displayable or two pages of 1024 x 1024 pixels. The /4M option can give two pages of 1280 x 1024 displayable or four pages of 1024 x 1024 pixels.

The basic display memory size for the VCU-V is 2 MB, which provides 2048 x 1024 addressable pixels (8 bit/pixel primary, 4 bit/pixel overlay). The memory is expandable to 16 MB. The /4M option gives one page of 1600 x 1280, two pages of 1280 x 1024 or four pages of 1024 x 1024 pixels. The /8M option can give two pages of 1600 x 1280, four pages of 1280 x 1024 or eight pages of 1024 x 1024.

The standard memory size for the VCT-V is 8 MB of 32-bits/word, byte addressable, no-wait state, dual-port Video RAM. This provides a 1280 x 1024 display size with 2048 x 1024 x 32 bit/pixel addressable pixels. The display memory is expandable to 16 MB. The /4M option can give one page of two pages of 1280 x 1024 or four pages of 1024 x 1024.

Writemask Register:

A 32-bit Writemask register permits individual bits in display memory to be write protected. This allows write operations (as opposed to read-modify-write) on display memory.

Pixel Clock: The VCU-V, VCT-V, and (optionally) the VCD-V utilize an ICS 1562 PLL controlled user programmable pixel clock generator. Most VCD-V configurations use a standard fixed frequency oscillator.

Scroll, Pan, and Zoom: Scroll - single line (smooth scroll).
Pan - resolution depends on initialization and color map:
VCU-V: anywhere on 8 or 16 pixel boundaries.
VCT-V: anywhere on 4 or 8 pixel boundaries.
VCD-V: anywhere on 2, 4, or 8 pixel boundaries.
Zoom - vertical (1, 2, 4, 8, 16, 32) - horizontal (depends on board type):
VCT-V and VCU-V: sub-integer, uses the ICS1562 to adjust master pixel clock.
VCD-V: BT459 color map based supports 1-16
BT482 and DLUT based has no zoom function

VCD-V Color Maps: The VCD-V analog output uses a Brooktree BT482 RAMDAC for low frequency (up to 1024 x 768) analog displays. The BT482 contains a 32 x 32 x 2 bitmapped cursor and works correctly with interlaced displays.

The VCD-V uses a Brooktree BT459 RAMDAC for high frequency (above 1024 x 768) analog displays. The BT459 contains a 64 x 64 x 2 bitmapped cursor, which does not work correctly with interlaced displays.

The VCD-V digital output consists of a 32K x 8 lookup table (LUT) and a 32 x 32 x 2 bitmapped cursor controller. 8 bit primary, 4 bit overlay, and 2 bit cursor data are passed through the LUT, which provides an 8 bit pixel output to the digital output connector.

VCU-V Color Map: The VCU-V output uses a Brooktree BT468 RAMDAC for displays ranging from 640 x 480 up to better more than 1600 x 1280 non-interlaced (only). The BT468 also contains a 64 x 64 x 2 bitmapped cursor which does not work correctly with interlaced displays.

VCT-V Color Map: The VCT-V output uses a Brooktree BT463 True Color RAMDAC for displays ranging from 640 x 480 interlaced up to better more than 1280 x 1024 non-interlaced. A separate dual BT431 cursor contains both crosshair and a 64 x 64 x 2 bitmapped cursors and works correctly with interlaced displays.

Serial I/O Ports: Four asynchronous serial I/O ports are contained in two Signetics 2681 DUARTs. Each port can be programmed for transmit and receive baud rates up to 38.4 Kb. Receive buffers are quadruply buffered to minimize the possibility of data overrun. Each DUART also contains a programmable timer/counter. Ordinarily, one port is for a serial mouse, one port is for the console (PTERM), one port is for LK401-type keyboard. The I/O lines for the fourth port are normally allocated as RTS/CTS for the console port, but can be rejumpered as a serial port.

PC Keyboard Port: An Intel 82C42PC keyboard controller supports standard PC keyboards (PTERM and PX Windows software support available). Connection is made via a mini-DIN PS/2 connector.

SCSI Port: On the VCT-V and VCU-V an optional Small Computer Systems Interface (SCSI) peripheral port, using an NCR 5380 controller, supports up to 7 high speed (1 MB/second) 8-bit parallel intelligent devices. It allows any compatible device (such as a SCSI disk) to be used.

For improved performance, an alternate port address mode allows the 34020 to utilize the 5380 pseudo-DMA mode, which automatically operates the SCSI handshake lines during data transfers.

Since the 5380 is highly programmable, the SCSI port may also be used as an 8-bit parallel I/O port.

At this time, no direct software support is available for the SCSI.

High Speed Port (HSP): A direct reading 32-bit port allows the 34020 to connect to an external device and read data directly into memory at page-mode speeds (100 ns per transfer). A simple handshake interface is used to control the external device. Special routines in the graphics subroutine package support HSP transfers.

The HSP is connected via the VMEbus P2 connector, using the VSB pinout for most signals. However, it is **not** VSB compatible.

VMEbus Access:

All device registers (34020, color map(s), DUARTs, PC Keyboard port, High Speed Port, and SCSI) and on-board memory are accessible to the bus through a 1K byte window in the A16 I/O space which uses the 16-bit Line Address Register (LAR).

The Line Buffer may also be located in A24 space if the host CPU only supports A16/D16. Best performance results if the VMEbus address space supports D32 transfers. Contact Peritek if you need to use this mode.

An optional direct A32 address mapping gives a 64 MB window into board memory. Except for PX Windows when used in a multi-processor environment, Peritek software does not use the A32 addressing feature.

Control Registers:

The graphics board has a four register block in the A16 space which contains the Control Status Register (CSR), Line Address Register (LAR), Line Buffer Address Register, Extended Address Register, and Interrupt Vector Address Register.

VMEbus Interrupts:

VMEbus interrupt controller supports a vectored interrupt from the 34020.

Bus Loading:

Two bus loads

Data Strobe to DTACK:

Times were measured using an HP1650A logic analyzer at the VME P1 connector, using 1000 test cycles. The 34020 was halted. The host CPU was a Motorola MVME162.

Depending on the host, you need to add about 150 ns of VMEbus overhead to get the total cycle time. The long maximum access time is due to access during a memory refresh cycle.

Assuming 150 ns VMEbus overhead, write transfer rates will be about 12.7 MB/s, and read transfer rates will be about 6.2 MB/s (when doing long word accesses).

34020 arbitrated accesses:

Write:	min:	140 ns	**Read:**	min:	420 ns
	max:	1.2 us		max:	2.0 us
	average:	168 ns		average:	497 ns

Module Size: 6U Eurocard, 233 mm x 160 mm.

Power Requirements: +5V +/- 5%, 3.0 A typical.

Environment: Temperature: 0 to 70 degrees C, operating
Humidity: 10% to 90%, non-condensing

Analog Video Connections: 15-pin VGA style, with Red, Green with Composite Sync, and Blue, separate horizontal and vertical sync. On the VCD-V, the pixel clock can optionally be output.

Genlock option includes HSYNC in, VSYNC in. Contact Peritek for details regarding genlock operation.

Digital Video Connector: On the VCD-V, a 26-pin (2 x 13) header supplies TTL level 8-bit digital, sync, blanking, and +5 to flat panel displays. A variety of 1, 4, and 8 bit monochrome and color panels have been tested and qualified. In the case of color panels, 9-bit panels such as the LQ10DH011 are connected 3 bits red, 3 bits green, and 2 bits blue. Contact Peritek about information regarding panel compatibility.

Serial Connector: DB-9 connectors are provided for the console and mouse connectors. A 4-pin modular (phone, RJ-11) connector is provided for the LK401 type keyboard. When a fourth serial port is desired, the RTS/CTS lines on the console connector are rejumpered for serial I/O. Fused +12 volts is provided on the the LK401 and mouse connectors. Fuses are actually Positive Temperature Coefficient (PTC) resistors which reset automatically when overload is removed.

SCSI Connector: Connection to the optional SCSI port is made on the VMEbus P2 connector following the standard P2 SCSI connector pinout. See Chapter 5.

Standard Display Timing Specifications

Display Format	Vertical Refresh	Horizontal Refresh	Pixel Clock
640 x 480	60 Hz	31.5 KHz	27 MHz
1024 x 768	70 Hz	60 KHz	55 MHz
1024 x 1024	57 Hz	60 KHz	80 MHz
1024 x 1024	60 Hz	64 KHz	100 MHz
1280 x 1024	67 Hz	64 KHz	110 Mhz
1280 x 1024	72 Hz	72 KHz	125 MHz
1600 x 1280	60 Hz	79 KHz	170 MHz

See Table 5-14 for more initialization table information.

Composite Video Signal: 1 Volt peak to peak consisting of:

660 mV Reference White +
54 mV Reference Black +
286 mV Sync Level

1.5 Monitor Requirements

Peritek graphics boards can be used with a wide variety of monitors. For best performance a monitor should have the following features:

- Color RGB with composite sync on green analog video input
- Switchable Termination (for monitor loopthrough)
- Height, pincushion, width, phase, and position controls
- Autotracking horizontal and vertical synchronization
- High bandwidth - 70 MHz (VCD-V)
  135 MHz (VCT-V)
  180 MHz (VCU-V)
- Horizontal refresh rate - 55 KHz (VCD-V)
  70 KHz (VCT-V)
  90 KHz (VCU-V)

1.6 Configuration Information

The basic graphics board includes:

- 40 MHz TMS 34020 Graphics Systems Processor,
- 1 page (1024 x 1024) of display memory
- 1 MB 34020 memory,
- hardware cursors,
- hardware pan, scroll, and zoom
- hardware byte swapper
- VMEbus interrupts.

Everything else is controlled by the options. Please contact Peritek and/or refer to the short form catalog for more information about configurations and accessories. The table on the next page shows some common models.

Note

X Windows requires serial I/O and 4 MB minimum 34020 system memory. All boards with a /Xn designation are X compatible.

Table 1-1 Common Board Configurations

Model	Overlay Memory	Serial I/O	4 MB 34020 Memory	X Windows Compatible	Display Format	Pixel Size
VCU-V/X12	yes	yes	yes	yes	1280 x 1024	8 + 4
VCU-V/X16	yes	yes	yes	yes	1600 x 1280	8 + 4
VCT-V/X12	yes	yes	yes	yes	1280 x 1024	24 + 4
VCD-V /X6/XD8	yes	yes	yes	yes	640 x 480	8 + 4

Options:

/X6	640 x 480 display (1024 x 1024 addressable)
/X10	1280 x 1024 display (1024 x 1024 addressable)
/X12	1280 x 1024 display (2048 x 1024 addressable)
/X16	1600 x 1280 display (2048 x 2048 addressable)
/nSM	34020 system memory in megabytes, where n = 4, 8, 16, or 32
/2M	2 pages of 1024 x 1024 (2048 x 1024 addressable pixels) primary and overlay
/4M	4 pages of 1024 x 1024 (2048 x 2048 addressable pixels) primary and overlay
/8M	8 pages of 1024 x 1024 (2048 x 4096 addressable pixels) primary and overlay
/SC	SCSI port
/4S	4 RS-232 data leads only serial I/O ports
/FPU	34082 Floating Point Coprocessor
/A6	non-X analog VCD-V configuration - no serial I/O and 1 MB of 34020 memory
/D8	non-X digital VCD-V configuration - no serial I/O and 1 MB of 34020 memory

Chapter 2 Installing Your Peritek Graphics Board

2.1 Introduction

There are 2 steps involved in getting your Peritek Graphics board to work in your system:

- Unpack and install the Peritek graphics board.
- Install the software

This chapter shows you how to install the Peritek graphics board in your computer. The PX Windows Manual and the Graphics Subroutine Package Manual provide instructions on how to install the software.

2.2 Unpacking Your Board

When you unpack your board, inspect the contents to see if any damage occurred in shipping. If there has been physical damage, file a claim with the carrier at once and contact Peritek for information regarding repair or replacement. Do not attempt to use damaged equipment.

Caution

Be careful not to remove the board from its antistatic bag until you are ready to install it. It is preferable to wear a grounded wrist strap whenever handling computer boards.

Some operating systems require that you reboot your system after installing a device driver, because only after the reboot will your system utilize the driver and recognize the board. If yours is such an operating system, you might like to install PX Windows or the Subroutine Package **before** installing the board since you will have to shut down the computer to install the board anyway. If you want to install the software before shutting down the computer, proceed to the correct part of the relevant software manual and return to this chapter afterwards.

2.3 VMEbus Installation

Before installing the board in the backplane, you must confirm that the addresses used by the Peritek graphics board are not used by other devices in your computer.

Caution

Only use software designed for your CPU. Do not, for example, use software designed for a 68030 on a 68040.

Also, do not change the board’s standard register address unless there is an address conflict. Changing the address might affect the success of the installation.

.

2.3.1 Default Interrupt Settings on Peritek Video Boards

Peritek boards are normally configured for interrupt level 3 (IRQ3). If you change this setting, the device driver needs to be changed accordingly.

Peritek boards have a programmable interrupt vector address, which is usually set by the software to default to E0 (hex). However, some platforms, such as Sun, permit the vector to be chosen transparently by the operating system. In these cases, you do not need to specify an interrupt vector address.

Make sure that any boards which do not use interrupts have their interrupt pass-grant jumper installed. Conversely, remove the jumpers for all boards that use interrupts. Finally, make sure you install the jumper in slot 0 IACK to IACKIN. Don't confuse this jumper with the IACKIN to IACKOUT jumper. On many backplanes, slot 0 IACK to IACKIN does not have a removable jumper; IACK is always connected to slot 0 IACKIN.

2.3.2 Checking Board Addresses

The Peritek VMEbus graphics boards have three address ranges:

• Control Registers	Jumper programmable
• Line Buffer	Software Programmable
• 64 MB Memory Window	Software Programmable

Note

Only the multiprocessor version of the PX Windows server uses the 64 MB memory window.

Before installing the board into your backplane, make sure no other devices in your computer respond to Peritek's graphics board addresses, listed in Table 2-1:

Table 2-1 VMEbus graphics board addresses

	Standard Address	Address Type	Data Type
Control Registers	`xxxxC000-xxxxC00F`	A16	D16, D32
Line Buffer	`xxxxy000-xxxxy3FF`	A16	D8, D16, D32
Full Memory	`A0000000-A3FFFFFF`	A32	D8, D16, D32
Interrupt Vector	`E0`	–	D8 interrupter

The xxxx is a place holder for digits that are processor specific. Some common values are shown in Table 2-2. Full memory settings are only used in multiple Peritek board configurations. The y in the line buffer address is a placeholder for a digit which is processor specific. The table on the following page gives you values for xxxx and y for some common CPUs.

Consult Chapter 6 for more information on determining addresses for boards not shown in the table.

Table 2-2 CPU board addresses

Processor/Mfgr.	Value of xxxx	Value of y	Addressing Modes
Force 68K	FBFF	0	A16
Force SPARC	FBFF	0	A16/D32
GMS 68K	FBFF	8	A16
Heurikon 68K	0100	8	A16
Motorola 68K,88K	FFFF	8	A16
Themis 68K	FFFF	0	A16/D32
Themis SPARC	FFFF	0	A16/D32
Sun, HP	0000	8	A16

These addresses are the defaults used by Peritek. Only the Control Register address is set by jumpers on the graphics board. The Line Buffer and Full Memory addresses and the Interrupt Vector are software configurable.

2.3.3 Installing the Graphics Board

Use the following procedure to install the Peritek graphics board into the VMEbus backplane.

1. Shut down the operating system and **turn off the power**.

Warning!

Never open the computer without turning off the power supply. Unless internal AC wiring is exposed, leave the power cord plugged in, so as to ground the computer chassis. You can easily get shocked, ruin computer parts or both unless you turn off the power. Even with power switched off, lethal voltages can exist in the equipment.

2. Open the computer and identify the empty slot in the card cage that is closest to the CPU. Do not leave any slots empty between the graphics board and the CPU.

Figure 2-1 Example VMEbus Backplane

SLOT	P1	P2
1	MVME167 (or equivalent) Single Board Computer	
2	Peritek Graphics Board (VCT-V, VCU-V, or VCD-V)	
3	Spare	
4	Spare	
5	Spare	
6	Spare	
7	Spare	
^	Spare	
|	Spare	
v	Spare	
21	Spare	

Note

Note: There must not be any open slots between the first and last boards which use either DMA or interrupts (this includes the Peritek graphics board, which uses interrupts).

The shorting jumper for Slot 2 IAKIN/OUT should be removed (assuming the graphics board is installed in that slot). Shorting jumpers should be installed for all unused slots..

3. Remove the interrupt pass/grant from the board slot.

   The jumper may be on the front or back of the backplane. Some backplanes don't have jumpers to remove. The jumper is an integral part of the slot. It is activated automatically when the card is inserted.

4. Wear a grounded wrist strap. Touch a metal part of the computer chassis, remove the graphics board from its anti static bag, and immediately slide it into the slot.

> **Caution**
>
> The static electricity that your body builds up normally can seriously damage the integrated circuits on the graphics board. You should first touch the metal part of the chassis, which will short circuit the static charge on your body to ground. It is preferable to wear a grounded wrist strap whenever handling computer boards.
>
> Handle the graphics board only by its edges. Oils from your hand can break down the metal used in the circuit board.

5. After making sure the board is seated correctly, tighten the screwlock on each end of the board.

6. Close the computer and plug the video cable into the monitor and the graphics board. Make sure to plug the three BNC cables, colored red, green, and blue, into the monitor's corresponding red, green, and blue inputs. Also, make sure the 75 ohm switch on the monitor is turned on. VGA monitors which use a 5-wire cable (which can be obtained on special order), may also require modified initialization tables.

2.3.4 What's Next?

Now at this point you can continue to the next section, ***2.3.5 Connecting the Mouse, Keyboard, and Console***, or if you are not using them, skip it and go on to the following section, ***2.3.6 Checking your Display***.

2.3.5 Connecting the Mouse, Keyboard, and Console

This section applies only to applications which use a mouse (or trackball) and keyboard. Plug in the mouse and the keyboard cables.

> **Note**
>
> If you have an older style Peritek board, it will have a 20-pin header which accepts a ribbon connector instead of separate keyboard, mouse and terminal connectors.
>
> If your graphics board does not use PTERM, your ribbon cable will have two connectors, labeled M and K. These letters stand for "mouse" and "keyboard." Plug these connectors in to the mouse and keyboard.
>
> If your graphics board uses PTERM, your ribbon cable will have four connectors, labeled M, K, C, and S. These letters stand for "mouse," "keyboard," "console," and "special." The S plug is not used. Plug the other three connectors in to the mouse, keyboard, and the console ports on your computer.

Plug the mouse cable into the 9-pin male connector labelled *MOUSE*.

Plug an LK201 or LK401 keyboard into the RJ-11 socket labelled *LK401*, ***or*** plug a PC keyboard with a PS/2 style connector or adapter into the round 6-pin socket labelled *PC KBD*.

If you are using the PTERM terminal emulator, plug the console cable from the computer into the 9-pin female connector labelled *CONSOLE*.

PTERM supports 9600 baud. Jumpers control the data bits, parity, and RTS/CTS and XON/XOFF protocol. See Section 2.4.10.

The console port of your computer should be set to these values. If you have trouble matching the board and computer console ports, refer to Chapter 6 or contact Peritek.

2.3.6 Checking your Display

Turn on the power and check your monitor's display.

If your graphics board does not use a PTERM terminal emulator or the CnP Graphics Subroutine Package autoboot PROM there will be no display. This is because the board doesn't have an automatic bootup sequence to initialize itself.

Only when you boot your computer and the graphics board software has been downloaded will you see anything. In the case of PX Windows, your monitor should display a uniform stippled raster and a colored cross cursor, which is controlled by the mouse. For the CnP, you have to load both the CnP.RAM and a test program (e.g. TQT01.RAM) before you will see anything.

If your graphics board uses the PTERM terminal emulator, a white, rectangular cursor should appear in the upper left corner of the monitor. As the computer boots, it should print messages on the screen. If none appear, make sure the console connector is correctly plugged in and the console terminal parity and data bits are set correctly (see Jumper Settings).

Once you have a picture on the screen, you may need to adjust the width, height, brightness, contrast, and hold controls on your monitor to get a good, centered image. If these controls don't adjust the image properly, the parameters used to set the 34020 graphics timing registers might be wrong. If you encounter display problems with PTERM, the timing parameters may need to be changed. However, are not user definable; they are hard-coded into the PROM. Contact Peritek for a different PROM set to set the correct display timings for your installation.

If you encounter display problems when the X server or CnP is running, the values in the initialization table you used may not be correct (see the Section 5.4. You can select a different table or call Peritek for assistance.

If you have any trouble with any part of the installation, refer to Chapter 6. Otherwise, proceed to the instructions supplied in your software manual.

Figure 2-2 Jumper Locations for the VCT-V and VCU-V

Figure 2-3   Jumper Locations for the VCD-V

2.4 Option Selection

It is best to first test out a board with the factory default configuration whenever possible. This minimizes the chance of introducing a problem into a known good board when you are not very familiar with the product. Sometimes, however, something must be changed. That is what this section is for. Before changing addresses, please read the commentary on VMEbus addressing which is in Section 6.2 of this manual.

The VCD-V, VCT-V, and VCU-V share virtually all jumper configurations. Only the jumper **locations** differ. Therefore, this section covers the changes to all three products. The VCT-V and VCU-V actually use the identical PC board. They differ only with plug in parts such as FPGA, color map, cursor, and memory configuration.

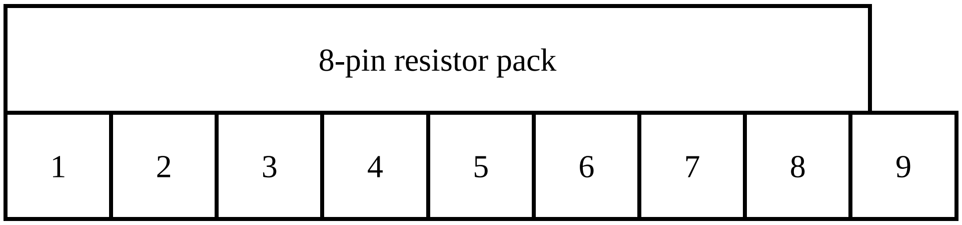
The following instructions tell how to modify the VCD-V FAB REV 2 (and on) and the VCU-V and VCT-V FAB REV 4 (and on) to a non-standard configuration. Refer to Figure 2-3 Jumper Option Locations for VCD and VCT/VCU (previous two pages) for jumper locations. For wire-wrap changes, only KYNAR or TEFLON, not enamel or plastic coated, insulated wire should be used.

2.4.1 CSR Addresses

The address range for the CSR block is jumper selectable to certain addresses in A16 space. As configured at the factory, bits 4, 5, 14, 15 are used in the address selection, bits 0-3 are used in the register selection, and bits 6-13 are hardwired low. If required, bits 6-13 can be some other pattern - contact Peritek if you need this.

Remember that the base addresses for the Line Buffer, Extended Address Block, and Interrupt Vector are all software programmable. Furthermore, if you select the Alternate group 12-15, you will conflict with the standard Line Buffer address (FFFF0000-FFFF03FF) used by Peritek software for the Line Buffer. Section 5.2 has complete information about programming these registers. Refer also to Chapter 6 and note the comments concerning use of the Motorola MVME167 and MVME187.

Figure 2-4 CSR Address and Interrupt Grant Level Jumpers



Pin Number	**Function**
1	GRD
2	A15
3	A14
4	A5
5	A4
6	VS2
7	VS1
8	VS0
9	GRD

Address Selection	**VMEbus (Hex)**	**default jumpers**
Standard	xxxxC000-xxxxC00F	A4, A5 installed
Alternate 1	xxxxC010-xxxxC01F	A5 installed
Alternate 2	xxxxC020-xxxxC02F	A4 installed
Alternate 3	xxxxC030-xxxxC03F	--
Alternate 4-7	xxxx80n0-xxxx80nF	A14, A5, A4
Alternate 8-11	xxxx40n0-xxxx40nF	A15, A5, A4
Alternate 12-15	xxxx00n0-xxxx00nF	A14, A15, A5, A4

Note: xxxx depends on host processor's A16 VMEbus address space.

2.4.2 Interrupt Grant Receive/Acknowledge

The VMEbus has a seven level interrupt grant receive/acknowledge protocol which requires each board to acknowledge that it is responding to the interrupt grant level that it requested. Three jumpers set this response level. Refer to the Jumper Location Figures for VCD and VCT/VCU and Figure 2-4 (above) for the location of the jumpers. In the table below, 0 equals jumper installed.

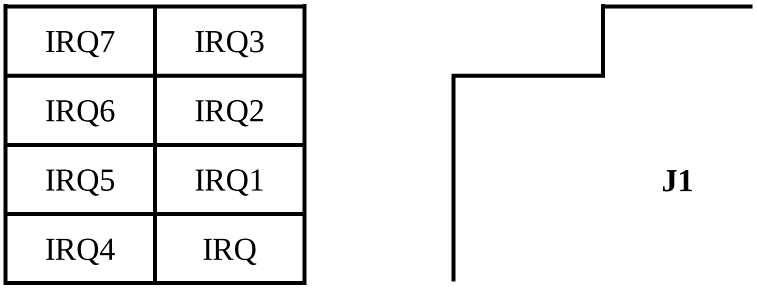
Table 2-3 Interrupt Grant Level

Grant Level	VS Jumper Number			Default
	2	1	0	
1	0	0	1	
2	0	1	0	
3	0	1	1	yes
4	1	0	0	
5	1	0	1	
6	1	1	0	
7	1	1	1	

2.4.3 Interrupt Priority

The VMEbus interrupt request priority is jumper programmable for the seven levels (1-7). The lower the priority number the less likely the board will be serviced. The Interrupt Request Priority jumper is located to the left side of J1 (VME P1 connector) and is labeled JP1. The pin layout is as follows:

Figure 2-5 Interrupt Priority Jumpers



IRQ7	IRQ3
IRQ6	IRQ2
IRQ5	IRQ1
IRQ4	IRQ

Caution

The Vector Priority setting must match the Interrupt Request Priority setting.

2.4.4 Flash EEPROM

The graphics boards allocate address space for 32-bit wide EEPROM. There are sockets for four 8-bit 32-pin PLCC Flash EEPROMS. Sizes supported are 32Kx8 e.g. AM28F256-150JC (for 128 KB total) through 256Kx8 e.g. AM28F020-150JC (for 1 MB total). 1 wait state is provided so access time should be less than 200nS. By virtue of the pin and address arrangement on these devices no size jumpers are required.

Peritek has a program which is available upon request which can be used to load images into the the EEPROMs. In order to allow the devices to program, a jumper must be installed on the board to enable 12 Volts to the EEPROMs (see below). If you have any questions, please contact Peritek.

VCD-V Program Voltage Enable

As shown in Figure 2-2, install a jumper between pins 2 and 3 of JP6.

VCT-V and VCU-V Program Voltage Enable

As shown in Figure 2-3, install a jumper between pins 2 and 3 of JP6.

> **Caution**
>
> **Remove the jumper once you are done programming.**

2.4.5 Autoboot Enable

The graphics boards can run automatically from on-board EEPROM on powerup or anytime SYSRESET is asserted. Peritek can supply a terminal emulator (PTERM) preloaded and ready to run (see Chapter 3). Note: if this jumper is installed **then autoboot EEPROMS must be installed**, otherwise the 34020 executes garbage.

VCD Autoboot Enable

As shown in Figure 2-2 and Figure 2-6 (below), install a jumper in the jumper pair labelled BEN to enable autoboot operation.

VCT-V and VCU-V Autoboot Enable

As shown in Figure 2-3, the jumper pair just to the right of U27 (MACH230 chip near the center of the board) and is labeled JP18. Install a jumper to enable autoboot operation.

2.4.6 DRAM and VRAM Size

The on-board 34020 DRAM and display memory VRAM are contained in SIMMs (Single Inline Memory Module) and may be changed in the field. Two pairs of jumpers allow for different size memories. The video memory size is in pages, where one **page = 1024 x 1024 pixels.**

On the **VCD-V**, the jumpers are part of jumper strip JP5.

Figure 2-6 VCD-V DRAM and VRAM Size and Autoboot Enable

GRD	GRD	GRD	GRD	GRD	GRD	GRD
D0	D1	V0	V1	resv	resv	BEN
RP4						

On the **VCT-V** and **VCU-V**, the jumpers are part of jumper strip JP19:

Figure 2-7 VCT/VCU DRAM and VRAM Size

Lower edge of U27 - MACH230	
GRD	D1
GRD	D0
GRD	V1
GRD	V0
GRD	resv

The tables on the next page shows how to set the DRAM and VRAM jumpers. ***Note that for both the DRAM and VRAM tables, 0 = Jumper Installed***

Table 2-4 DRAM Size Jumpers

D1	D0	DRAM Size
0	0	1 or 4 MB
0	1	8 MB
1	0	16 MB
1	1	32 MB

Table 2-5 VRAM Size Jumpers

V1	V0	VCD-V & VCT-V VRAM Size	VCU-V VRAM Size
0	0	1 page	2 page
0	1	2 pages	4 pages
1	0	4 pages	8 pages
1	1	4 pages	8 pages

2.4.7 Master Pixel Clock Oscillator Frequency

VCU-V and VCT-V

VCU-V and **VCT-V** boards have a software programmable pixel clock, which allows virtually any frequency pixel clock to be chosen. Peritek distributes an number of standard initialization tables, and can provide custom versions upon request. See Section 5.4 for more information

The **VCU-V** uses the ICS1562 clock chip, which actually gives a maximum pixel clock in excess of 200 MHz. Note that the BT468 color map chip must be upgraded from its standard 170 MHz to take advantage of this.

The **VCT-V** uses either the ICS1562 or the ICS1572 (in some special configurations). The ICS1572 has a 150 MHz maximum clock frequency, which is more than what the BT463 color map chip is limited to (135 MHz). The 1562 and 1572 take exactly the same program parameters.

VCD-V

Most VCD-V boards have a fixed frequency crystal controlled oscillator to maintain compatibility with earlier revision VCD-Vs. However, a software programmable pixel clock (ICS1572) is being phased into the standard configuration **VCD-V/X6/D8** and **VCD-V/X12/D8** for new customers only**.** If this presents a problem, please contact the factory.

Routines provided in Peritek's Graphics Subroutine Package and PX Windows software are used to set the pixel clock. Please contact Peritek if you require assistance in selecting a correct table from those distributed with the software.

2.4.8 Interlaced Operation and VCU-V Slow Mode

Peritek's VCT-V and VCD-V/X6 type boards support interlaced operation when the proper initialization is specified. No other changes are required.

The VCU-V and VCD-V/X12 type boards will operate in interlaced mode, but the cursor appearance and positioning will not be correct. This is because the cursor built into the color map does not know about interlaced operation.

In order to run the VCU-V in 640x480 interlaced mode (or any mode with a pixel clock of less than 18.25 MHz) the blanking jumper must be changed to the slow mode. This jumper is labeled JP7 and is located just below the BT468 (VCU-V) - see Figure 2.3. Connect pins 1 and 2 (leftmost 2 pins) for regular mode or connect pins 2 and 3 (rightmost 2 pins) for slow mode. The jumper may be left in the slow position for resolutions up to 1280x1024, but the horizontal cursor position will be off 8 or 16 pixels. The minimum pixel frequency in the slow mode is 4 MHz.

2.4.9 BT482 Output Level (VCD-V/T - special order only)

A 15-turn potentiometer (R74) can be used to precisely adjust the peak voltage level. Refer to Figure 2.2 for the location of the control, which can be accessed through a small hole in the front panel near the VGA connector.

2.4.10 Selecting PTERM Options

Table 2-6 PTERM Serial Control Options

Jumper	Open	Shorted
A	8 bits/No parity	7 bits/Even parity
B	XON/XOFF disabled	XON/XOFF enabled
C	RTS/CTS disabled	RTS/CTS enabled

The default with no jumpers installed is 9600, 8 bits, no parity, and no flow control. The PC Keyboard is auto detected. If it is installed it will be used, otherwise the LK401 serial interface will be used.

The XON/XOFF flow control only applies to data being sent to PTERM. Keyboard data send from PTERM to the host is not flow controlled with XON/XOFF.

PTERM has a 8KB input buffer so that it can handle bursts of up to 8KB without requiring XON/XOFF flow control.

The RTS/CTS flow control is done by programming the UART to enable UART control of the RTS/CTS lines. ***THIS OPTION HAS NOT BEEN COMPLETELY TESTED.*** It is unlikely that the CTS signal will be asserted because the 34020 should in general service UART interrupts prior to the next character being received. The RTS signal will control output from PTERM to the host. If this signal is not connected the UART will refrain from sending characters to the host.

Figure 2-8 VCD-V PTERM Serial Jumpers (JP33)

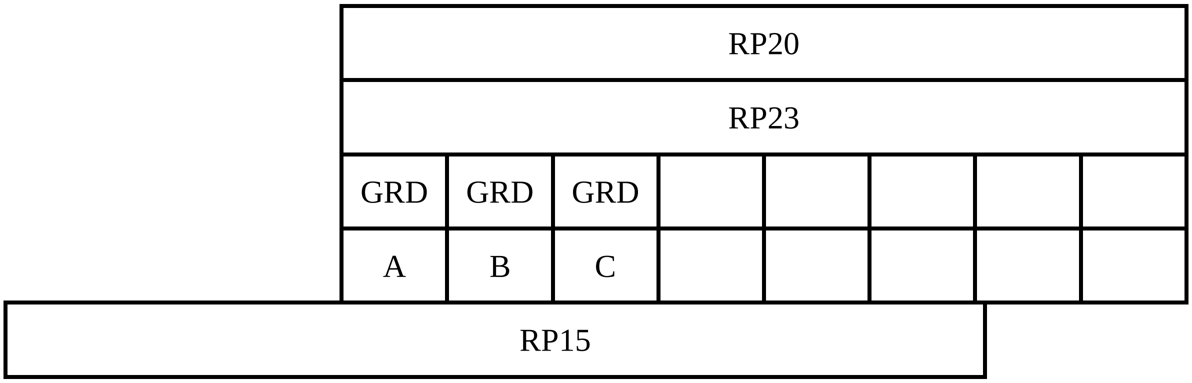
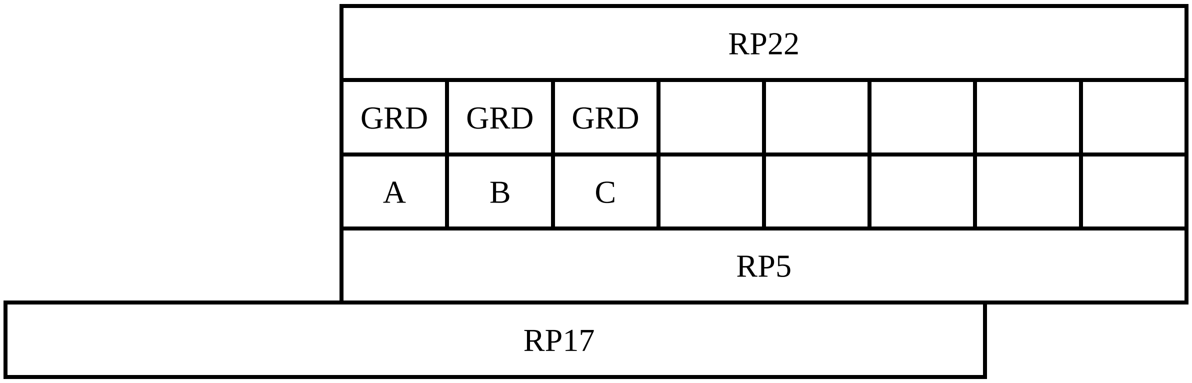


Figure 2-9 VCT-V and VCU-V PTERM Serial Jumpers (JP22)



2.4.11 Selecting Serial I/O Options

Section 2.5.1 has connector pinouts for the Serial I/O ports. In the follwoing tables, CnP Port refers to the way the Graphics Subroutine Package uses these ports.

Note

Total current draw for all fused +5 volt outputs should not exceed .5 A. Total current draw for all fused +12 volt outputs should not exceed .5 A.

There is a 470 ohm current limiting resistor in the -12 volt line. There is a 1K ohm current limiting resistor power source to the Mouse Connector pins 4 and 7.

Table 2-7 Mouse Port or CnP Port 0 (DUART 0 channel A)

Jumper	Mouse Connector Pin Option	Default
JP8 1-2	pin 3 to -12V	yes
JP8 2-3	pin 3 to Port 0 TX	no
JP20 1-2	pin 6 to Port 1 RX	no
JP21 1-2	pin 8 to Port 1 RX	no
JP10 1-2 (only)	pin 9 to fused (.5A) +12 volts	no
JP20 2-3 (only)	pin 9 to fused (.5A) +5 volts	no

The mouse port is a DB9 male connector.

Combined current draw on pins 4 and 7 should not exceed 10 mA. Current draw on pin 3 should not exceed 10 mA.

Table 2-8 LK401 Keyboard Port or CnP Port 1 (DUART 0 channel B)

There are no jumper options.

RJ11 "handset" type jack. Keyboard plugs straight in. Pin 2 is on the fused (.5A) +12 volt supply.

Table 2-9 Console Port or CnP Port 2 (DUART 1 channel A)

Jumper	Console Port Connector Pin Option	Default
JP13 1-2	pin 7 to CTS	yes
JP13 2-3	pin 7 to Port 3 RX	no
JP12 1-2	pin 8 to RTS	yes
JP12 1-3	pin 8 to Port 3 TX	no
JP11 1-2 (only)	pin 9 to fused (.5A) +12 volts	no
JP11 2-3 (only)	pin 9 to fused (.5A) +5 volts	no

The console port is a DB9 female connector.

At the connector, CTS is an input and RTS is an output. Not all installations require these signals.

Table 2-10 Extra Port or CnP Port 3 (DUART 1 channel B)

No connector is supplied for this port, but the signals can be output from the Console Port with option jumpering. See Console Port Options, above.

2.5 Connections to the VCU-V, VCT-V, and VCD-V

With the introduction of the VCU-V and VCT-V FAB REV 4 and the VCD-V FAB REV 2, the connector layout has been completely revamped in the interest of providing a more "user friendly" front panel. The mouse and keyboard connectors now use standard interface connectors and the SCSI (available only by special order) has been moved to the VMEbus P2 standard SCSI pinout. A new option, the High Speed Port (HSP), also makes use of the P2 A and C rows and **cannot coexist** with the SCSI option. Only the Video connector and the VCD-V digital connector remain un
changed from earlier revisions.

The 34020 emulator connector, while still included in the PCB artwork, has been deleted as an orderable option. It is for factory use only. Contact Peritek if you need emulator connections.

There are five unique connectors installed on the VCU-V, VCT-V, and VCD-V. An additional 26 pin header is installed on the VCD-V for digital output.

The connectors include:

Section 2.5.1a	a console (PTERM) 9-pin female connector
Section 2.5.1b	a Serial PC Mouse/Trackball DB-9 male connector
Section 2.5.1c	a PS/2 mini-DIN PC Keyboard connector
Section 2.5.1c	an LK401 (RS-232 serial) RJ-11 modular connector
Section 2.5.2	a VGA-style high density DB-9 video connector
Section 2.5.3	VMEbus P2 connections for High Speed Port (HSP)
Section 2.5.4	VMEbus P2 connections for SCSI port (VCT and VCU)
Section 2.5.5	Digital Video connector (VCD-V only)

2.5.1 Console, Mouse, Trackball, and Keyboard Connectors

The graphics board connectors match the standard connectors for the Console, Keyboard, Mouse, and Trackball. Connectors are provided for both LK401 (RS-232 type) keyboard and PC compatible keyboard. Note that the graphics boards supports Data-Leads-Only RS232C for the keyboard, but RTS/CTS is supported for the console terminal port. The RTS/CTS lines can be redefined as an additional Data-Leads-Only RS232C data port. Data-Leads-Only means that the XON/XOFF software protocol must be used to control data flow.

> **Note**
>
> Section 2.4.11 provides information on the jumper configuration options for the RS-232 ports.

Fused +12 and +5 are provided since the mouse and keyboard require power. The +5 and +12 are protected by auto-resetting fuses, which are actually PTC elements which reset automatically when an overload is removed.

The Peritek Graphics Subroutine Package (CnP) includes general purpose serial I/O routines, but has no knowledge of the device connected to the port. Sample programs exist which process keyboard and mouse inputs, but no "intelligent" keyboard or mouse software is available. That is why we have PX Windows!

Peritek can supply cables and devices - please contact the factory for ordering information.

2.5.1a Console

The console port is used in conjunction with the PTERM terminal emulator, and allows the graphics board to function as a "dumb terminal". The console port is connected to the host CPU serial terminal connector (serial port 1 on most systems). The graphics board functions as a terminal on power up. Once the PX Windows or Graphics Subroutine Package program starts up the console terminal function goes away and may not be recalled. No "hot-key" provision to dynamically switch between the application program and PTERM exists at this time. However, in PX Windows, the PTERM may be restarted by running a special program which kills the server and starts up PTERM.

The console port is suitable for applications which require long cables, because it uses the RS-232C electrical protocol which can support cable lengths well in excess of 100 feet. Peritek uses a DB9 female D-Sub connector.

Table 2-11   Console Connector Pinout

9-pin D-Sub Pin Number	Description
1	not used
2	Transmit Data to Console Port
3	Receive Data from Console Port
4	not used
5	Ground
6	not used
7	CTS (from Console Port)
8	RTS (to Console Port)
9	not used

2.5.1b Mouse and Trackball

If you buy the Keyboard and Mouse or Trackball directly from Peritek they will come tested and prepared to work correctly with Peritek PX Windows and PTERM Terminal Emulator. The software will work automagically with either a PC compatible keyboard or an LK401 compatible keyboard. If no keyboard is installed the default is the LK401. The PX Windows mouse or trackball should be a 3 button, Mouse Systems protocol (5 byte) device, although the 2-button Microsoft Mouse protocol is now supported.

The Peritek Optical Mouse is a Mouse Systems Serial PC Mouse. It uses optical technology for postioning, thus requiring a small pad, which is included. Peritek can also supply the low-cost Peritek Roller Mouse which uses a small rolling ball and mechanical position encoders. A pad is not supplied, but can sometimes make the roller mouse operation smoother. The Peritek Trackball works like an upside-down roller mouse, but the ball is much larger.

In all cases, the unit operate in 3 button Mouse Systems protocol, which is best suited for PX Windows. However, 2 button Microsoft Mouse mode can be supported with a special command line option when starting the PX Windows server (see the PX Windows User Manual "Man Page" section. If you purchase your own mouse, make sure it can operate in the more desirable 3 button mode.

The mouse relies on the current which can be sourced through the serial port's transmit, RTS and CTS lines for power. However, since the graphics board mouse port is really just data leads only, fused +12 and -12 volts are supplied to these lines instead.

Both Peritek Mouse units are suitable for applications which require long cables, because they use the RS-232C electrical protocol which can support cable lengths well in excess of 100 feet. Peritek uses a DB9 male D-Sub connector to pass data and power to the mouse, which requires +12 and -12.

Note

If you experience difficulty getting a device to work, especially a Mouse or Trackball, you may be drawing too much current from pins 3, 4, or 7. There are current limiting resistors in series with the power sources for these lines.

Table 2-12   Mouse Connector Pinout

9-pin D-Sub Pin Number	Description
1	not used
2	Data from Mouse
3	-12 Volts via 470 ohm resistor
4	+12 Volts via 1000 ohm resistor
5	Ground
6	not used
7	+12 Volts via 1000 ohm resistor
8	not used
9	not used

The resistors are to make the marginal RS-232 circuits of low-cost mice and trackballs work correctly. They also limit current inrush which can damage some devices.

2.5.1c Serial and PC Keyboards

Peritek Serial Keyboard

The **Peritek Serial Keyboard** is a DEC LK401-AA unit which is especially suited for applications which require long cables. The Serial Keyboard uses the RS-232C protocol which can support cable lengths in excess of 100 feet. It is a 4800 baud unit, and supplies a keyswitch matrix code to the software. It uses an RJ11 4 pin (handset) phone connector to pass data and power (+12) to the Keyboard.

Table 2-13 LK401 Connector Pinout

RJ11 Pin Number	Description
1	Data from Serial Keyboard
2	Fused +12 Volts
3	Ground
4	Data to Serial Keyboard

Peritek PC Keyboard

The **Peritek PC Keyboard** is suitable for applications which **do not** require long cables, because it uses a TTL level electrical protocol which cannot support cable lengths in excess of about 10 feet. 3rd party cable extender/amplifiers are available to overcome this limitation. Peritek uses the mini-DIN PS/2 keyboard connector to pass data and power (+5) to the Keyboard. If you use a standard PC keyboard, you will need a PC DIN to PS/2 mini-DIN adapter which is available from most computer stores.

Table 2-14 PC Keyboard Connector Pinout

PS/2 Mini-DIN Pin Number	Description
1	Bidirectional Keyboard Data
2	not used
3	Ground
4	Fused +5 Volts
5	Bidirectional Keyboard Clock
6	not used

2.5.2 Video Connector

The video connector is a VGA style compressed 15 pin D-SUB. The R, G, and B video outputs are driven by the RAMDAC, which is capable of driving terminated cable (75 ohms) to standard RS-330/IRE levels. Cable length should be limited to 50 feet unless you use low loss RG-59.

A VGA monitor can be plugged in directly, using a standard VGA connector. You must use the correct initialization table, since a VGA monitor depends on the sync polarities to determine operating frequency. If you use the Peritek **VGA-3/20** VGA to BNC cable, only composite signals are carried to the monitor, and it will "autoscan", if the monitor is so equipped.

The direction of the TTL Dot Clock, Vertical/Composite Sync, and Horizontal Sync and the polarity of the Sync signals are controlled by a combination of jumpers and the Zoom Control Register (see Section 5.5).

Table 2-15 Video Connector Pinout

Pin	Description	Pin	Description
1	RED	8	GND
2	GREEN	9	GND
3	BLUE	10	GND
4	NC	11	HSYNCIN (for genlock option)
5	GND	12	VSYNCIN (for genlock option)
6	GND	13	HSYNCOUT
7	GND	14	VSYNCOUT or CSYNCOUT
		15	DOTCLOCKOUT on VCD-V/A6 and X6

Notes for External Sync

Genlock requires separate horizontal and vertical sync signals (active low) to be input and a jumper change. HSYNCOUT (horizontal sync out) polarity is programmable. CVSYNCOUT (composite or vertical sync out) mode (composite or vertical) and polarity are programmable. Refer to Section 5.5 for information on programming these pins. Contact Peritek if you are interested in using this feature.

2.5.3 High Speed Data Port (HSP)

The output buffers on the User Equipment (UE) should be 74ACT374 (FCT, BCT, and ABT are OK), with 64 mA output drivers terminated with 220/330 resistors on the UE. Input signals (from the graphics board to the UE) will be conditioned by a hysteresis device such as a 74F14. **See Section 5.15 for HSP signal functions**.

Table 2-16 HSP P2 Connector Pin Connections

Signal Name	Pin	Polarity	Direction	UE Type	Board Type
DATA_00	A1	active high	VCD input	74ACT374	74ACT652
DATA_01	C1	active high	VCD input	74ACT374	74ACT652
DATA_02	A2	active high	VCD input	74ACT374	74ACT652
DATA_03	C2	active high	VCD input	74ACT374	74ACT652
DATA_04	A3	active high	VCD input	74ACT374	74ACT652
DATA_05	C3	active high	VCD input	74ACT374	74ACT652
DATA_06	A4	active high	VCD input	74ACT374	74ACT652
DATA_07	C4	active high	VCD input	74ACT374	74ACT652
DATA_08	A5	active high	VCD input	74ACT374	74ACT652
DATA_09	C5	active high	VCD input	74ACT374	74ACT652
DATA_10	A6	active high	VCD input	74ACT374	74ACT652
DATA_11	C6	active high	VCD input	74ACT374	74ACT652
DATA_12	A7	active high	VCD input	74ACT374	74ACT652
DATA_13	C7	active high	VCD input	74ACT374	74ACT652
DATA_14	A8	active high	VCD input	74ACT374	74ACT652
DATA_15	C8	active high	VCD input	74ACT374	74ACT652
DATA_16	A9	active high	VCD input	74ACT374	74ACT652
DATA_17	C9	active high	VCD input	74ACT374	74ACT652
DATA_18	A10	active high	VCD input	74ACT374	74ACT652
DATA_19	C10	active high	VCD input	74ACT374	74ACT652
DATA_20	A11	active high	VCD input	74ACT374	74ACT652
DATA_21	C11	active high	VCD input	74ACT374	74ACT652
DATA_22	A12	active high	VCD input	74ACT374	74ACT652
DATA_23	C12	active high	VCD input	74ACT374	74ACT652
DATA_24	A13	active high	VCD input	74ACT374	74ACT652
DATA_25	C13	active high	VCD input	74ACT374	74ACT652
DATA_26	A14	active high	VCD input	74ACT374	74ACT652
DATA_27	C14	active high	VCD input	74ACT374	74ACT652
DATA_28	A15	active high	VCD input	74ACT374	74ACT652
DATA_29	C15	active high	VCD input	74ACT374	74ACT652
DATA_30	A16	active high	VCD input	74ACT374	74ACT652
DATA_31	C16	active high	VCD input	74ACT374	74ACT652
REL	C25	active low	VCD output	74F14/LS244	74ACT244
HSL	C28	active low	VCD output	74F14/LS244	74ACT244
VSL	C27	active low	VCD output	74F14/LS244	74ACT244
PRDYL	C26	active low	VCD input	74LS244	74F14/LS244

GND A17, A25, A26, A27, C17, C18, C19, C20, C24, B2, B12, B22, B31
VCC B1, B13, B32

2.5.4 8-bit SCSI Port (VCT-V and VCU-V)

For its optional SCSI port, Peritek follows the "standard" 8-bit SCSI pinout on the VMEbus P2 connector popularized by Motorola. P2 is a 96 pin header (3 rows x 32 pins), with the B, or center row reserved for use by the VMEbus address and data buses. Rows A and C are user assignable. The SCSI port shares some of the same pins with the HSP (see Section 2.5.3, above). Therfore, you can't have both options on the same board.

The following table lists the P2 pin assignments and the corresponding cable connections for the SCSI option. Remember that only the SCSI or the HSP can be used on a particular board.

Table 2-17 VCU-V and VCT-V SCSI Connections to VMEbus P2

	P2 connector	**Standard SCSI Connector**	
Pin	**Signal Name**	**Pin**	**Signal Name**
A1	Data Bit 0, Low Active	2	Data Bit 0, Low Active
A2	Data Bit 1, Low Active	4	Data Bit 1, Low Active
A3	Data Bit 2, Low Active	6	Data Bit 2, Low Active
A4	Data Bit 3, Low Active	8	Data Bit 3, Low Active
A5	Data Bit 4, Low Active	10	Data Bit 4, Low Active
A6	Data Bit 5, Low Active	12	Data Bit 5, Low Active
A7	Data Bit 6, Low Active	14	Data Bit 6, Low Active
A8	Data Bit 7, Low Active	16	Data Bit 7, Low Active
A9	Data Bit Parity, Low Active	18	Data Bit Parity, Low Active
		20	Spare
		22	Spare
		24	Spare
		26	Spare
		28	Spare
		30	Spare
A10	Attention, Low Active	32	Attention, Low Active
	Spare, Low Active	34	Spare, Low Active
A11	Busy, Low Active	36	Busy, Low Active
A12	Acknowledge, Low Active	38	Acknowledge, Low Active
A13	Reset, Low Active	40	Reset, Low Active
A14	Message, Low Active	42	Message, Low Active
A15	Select, Low Active	44	Select, Low Active
A16	Command, Low Active	46	Command, Low Active
A17	Request, Low Active	48	Request, Low Active
A18	Input, Low Active	50	Input, Low Active
		1-49	ODD-numbered pins are Ground

Build the cable so that only the signal lines are connected. The grounds can be connected just on the peripheral end.

2.5.5 Digital Video Connector (VCD-V only)

The digital video connector is configured at the factory to support a variety of different requirements. **Type 1** supports 1 pixel per clock time, up to 8 bits/pixel. **Type 2** supports 2 pixels per clock, up to 4 bits/pixel. Connection tables are shown on the next 2 pages.

The issue of whether the graphics board should source power to the panel is somewhat controversial. This is because there is a wide variety of panels available and some draw a considerable amount of power. Peritek recommends that you evaluate the power requirements carefully. In addition, some panels require controlled power sequencing. It has been Peritek's experience that if the panel and the graphics board are powered on and off simultaneously that additional sequence control is not required. Nevertheless, some users feel more strongly. The VCD-V as it exists today does not include sequenced power. An ECO procedure does exist and can be supplied if necessary.

The next circuit revision (available 4/95) will include sequenced +5 and +12 sources on the digital connector. It will also include a wider video data output path (up to 24 bits).

With respect to the Type 1 panels, since the VCD-V has an 8 bit output port, and the human eye is least sensitive to variations in blue, the three bits of blue available on the color panels has been reduced to two bits by connecting B0 (LSB) and B2 (MSB) together for PX Windows applications. If you are not using PX Windows, you are free to connect the lines up in any way you wish. The following table shows some sample color values.

Table 2-18 PX Windows Basic Color Table for Type 1 TFT-LCD Panels

	RGB Input Values							
Color	**R0**	**R1**	**R2**	**G0**	**G1**	**G2**	**B1**	**B0/B2**
Black	0	0	0	0	0	0	0	0
Blue	0	0	0	0	0	0	1	1
Green	0	0	0	1	1	1	0	0
Light Blue	0	0	0	1	1	1	1	1
Red	1	1	1	0	0	0	0	0
Purple	1	1	1	0	0	0	1	1
Yellow	1	1	1	1	1	1	0	0
White	1	1	1	1	1	1	1	1
Red Scale	0	0	0	0	0	0	0	0
(darker)	1	0	0	0	0	0	0	0
.	.	.	.	.	.	.	.	.
(brightest)	1	1	1	0	0	0	0	0
Green Scale	0	0	0	0	0	0	0	0
(darker)	0	0	0	1	0	0	0	0
.	.	.	.	.	.	.	.	.
(brightest)	0	0	0	1	1	1	0	0
Blue Scale	0	0	0	0	0	0	0	0
(darker)	0	0	0	0	0	0	1	0
.	0	0	0	0	0	0	0	1
(brightest)	0	0	0	0	0	0	1	1

Table 2-19 J4 - Type 1 Digital Video Connector to Sharp TFT-LCD Panels

VCD-V Pin	J4 connector Signal Name	PX Windows Signal Name	LQ10D011 Pin	LQ10DH11 Pin	LQ10DH15 Pin	LQ10D021 Pin
1	Data bit 0	R0	3	4	4	CN1-5
3	Data bit 1	R1	4	3	3	CN1-6
5	Data bit 2	R2	5	6	6	CN1-7
7	Data bit 3	G0	7	8	8	CN1-9
9	Data bit 4	G1	8	7	7	CN1-10
11	Data bit 5	G2	9	10	10	CN1-11
13	Data bit 6	B1	12	11	11	CN1-14
15	Data bit 7	B0,B2	11,13	12,14	12,14	CN1-13,15
17	/Composite Blank	n/a	--	--	22	CN2-5
19	/Horizontal Sync	n/a	15	16	16	CN1-3
21	/Vertical Sync	n/a	17	18	18	CN1-4
23	Pixel Clock	n/a	1	2	2	CN1-1
25,26	+5 Volts	-- use direct power supply connection --				
2	Ground	n/a	2	1	1	2
4	Ground	n/a	-	5	5	-
6	Ground	n/a	6	-	-	-
8	Ground	n/a	-	9	9	8
10	Ground	n/a	10	-	-	-
12	Ground	n/a	-	13	13	12
14	Ground	n/a	14	15	15	-
16	Ground	n/a	16	-	-	-
18-24	Ground	n/a	-	-	-	-
Computer power supply		+5	18	17	17	CN2-1,2
		+12	20	19	19	--
		Ground	19	20	20	CN2-3,4
	Lighting	n/a	Backlit	Backlit	Backlit	Edgelit
Power Sequencing?		n/a	yes	yes	yes	no

Typical power sequence: power-up: +5 on --> logic on --> +12
power-down:+12 off --> logic off --> +5 off

Panel Model	Logic/Power Connector		Power Connector(s)	
LQ10D011	Hirose	DF11-22DS-2C	Molex	51005-0800 (backlight)
LQ10DH11	Hirose	DF11-22DS-2C	Molex	51005-0800 (backlight)
LQ10DH15	Hirose	DF11-22DS-2C	Molex	51005-0800 (backlight)
LQ10D021	Hirose	DF13-15S-1.25C	Hirose	DF13-6S-1.25C (logic)
			JST	S2B-EH (backlight)

Table 2-20   J4 - Type 2 Digital Video Connector to Sharp EL Panels

Sharp EL Panel Model LJ64ZU48/9

J4 Pin	**J4 connector Signal Name**	**Standard Pin Name**	**Sharp Pin Name**	**Sharp Signal Name**
1	Pixel 1, Data bit 0	24	B2	D10
3	Pixel 1, Data bit 1	23	A2	D11
5	Pixel 1, Data bit 2	22	B3	D12
7	Pixel 1, Data bit 3	21	A3	D13
9	Pixel 0, Data bit 0	20	B4	D00
11	Pixel 0, Data bit 1	19	A4	D01
13	Pixel 0, Data bit 2	18	B5	D02
15	Pixel 0, Data bit 3	17	A5	D03
17	no connect			
19	/Horizontal Sync	11	A8	HSYNC*
21	/Vertical Sync	9	A9	VSYNC*
23	Pixel Clock/2	13	A7	2CLK
25,26	+5 Volts	-- use direct power supply connection --		
2,4,6,8	Ground	--	--	--
10	Ground	10	B9	GND
12	Ground	12	B8	GND
14	Ground	14	B7	GND
16,18,20,22,24	Ground	--	--	--
Power Supply	+5 volts	3,4	A12,B12	VL
	+24 volts	5,6	A11,B11	VD
	Ground	7,8	A10,B10	GND

Logic/power connector is a standard dual row standard .1" connector

Chapter 3
Software Summary

3.1 Introduction

This chapter provides an overview of Peritek's software offerings. Peritek also has Software Product Descriptions and complete Technical Manual sets for the PX Windows and Graphics Subroutine Package products.

Peritek provides software for the VCU-V, VCT-V, and VCD-V including Peritek PX Windows (X11R6 X Windows Server), 34020 Compiler Tools, Peritek simple console Terminal emulator (PTERM), and a comprehensive Graphics Subroutine Package (generically CnP). The following table summarizes the **current** availability. Contact Peritek if your choice is not shown.

3.2 Software Availability by Platform and OS

Table 3-1 Peritek Software and Operating Systems Support

Operating System	Current OS Version	CPU Type	PX Windows	Subroutine Package
HPUX	9.0	PA/RISC	yes	no
LynxOS	2.2	68K	yes	no
OSF/1	3.1	Alpha	yes	no
OS9	3.0	68K	yes	yes
pSOSystem	2.0	68K	yes	yes
SGI	5	R3000	yes	yes
Solaris	2.3	SPARC	yes	no
SunOS	4.1.3	SPARC	yes	yes
Unix/V68	SVR3	68K	yes	yes
Unix/V88	SVR4	88K	yes	yes
VMEexec 68K	3.0	68K	yes	no
VMEexec 88K	3.0	88K	yes	no
VxWorks	5.1	68K	yes	yes
VxWorks.alpha	5.1	Alpha	yes	no
VxWorks.sparc	5.1	SPARC	yes	no

In addition to being available on tape media, the board side of PX Windows, PTERM, and CnP Subroutine Package can be provided in PROM. Jumpers permit the board to "autoboot" into the terminal emulator for use as a console terminal (see Section 3.4 below).

3.3 Write Posting

Features of many of the newer CPU designs include pipelining and write posting. The CPU, which is much faster than the VMEbus interface, is allowed to store (or post) a write operation to the CPU board's VMEbus controller. The controller takes care of the write within the timing requirements of the VMEbus. Pipelining is a procedure whereby the CPU can process more than one instruction at a time. As a result, instructions are not necessarily completed in the order that they were started.

In the case of sequential accesses to the VMEbus, which the Peritek baords use, it can happen that the a write of the Peritek graphics board Line Buffer can occur before a write to the Line Address Register (LAR) has been completed. If you had wanted to change the LAR and then write, you are not guaranteed that this has happened. This results in incorrect operation. The way to get around this is to immediately read back the data which has been written to the LAR, which flushes the pipeline and ensures correct operation. Since this is a problem just for the LAR, the performance impact is minor.

Peritek can supply its software with the read after write operation already incorporated. When ordering software, be sure to specify the CPU. Known offenders include 68040 and MIPS R3000 based CPUs.

3.4 PX Windows Server

Peritek's PX Windows Server is a Motif client compatible X Windows X11R6 board based server for a variety of Operating Systems (see Table 3-1). All functions of the server are actually executed by the 34020, which maximizes performance and eliminates many host processor responsibilities.

Peritek supplies the hardware specific parts of the X Window System, which is the server. Peritek has written its own highly optimized graphics layer for the 34020. The software is broken up into 2 functional parts: the board-based X server and the CPU host side "stub program" which provides a communication link between the server, clients, and the CPU network and file system resources.

The board side server code also provides complete support for PC-compatible keyboard or LK401-AA keyboard and Microsoft 2-button and Mouse Systems 3-button compatible pointing devices (i.e. mouse or trackball).

X Windows is a machine independent network based windowing system. It divides graphics functions into two parts:

1) The server, which controls the hardware dependent functions such as the mouse, keyboard or trackball, and graphics display; and

2) The client(s), which is (are) the actual programs which the user wants to interact with. This might include a terminal emulator, desktop publishing program, or an image processing package. The client application is usually linked with the standard XLIB library which manages the actual communications between clients and the server.

Most operating systems come supplied with a local xlib and a standard client package. Many also come with the Motif window manager. Contact your OS vendor for specifics on what they supply.

Under certain circumstances and for particular operating systems, Peritek can supply an extended version of PX Windows which includes a client side package (including Motif). As this software is currently in development, please contact Peritek for availability.

3.5 Graphics Subroutine Package

The Graphics Subroutine Package, comprises a significant "value-added" component for the Peritek graphics controllers. It is called CDP for VCD-V, CUP for the VCU-V, and CTP for the VCT-V and is termed generically here as CnP. It is intended for the user who wishes to interface an application program directly to the board. The subroutine package is based on the TI math/graphics library. Modified and enhanced by Peritek, this package contains over 200 subroutines. This package is designed to allow the user to program the board without having to contend with all the hardware details.

The package is compatible with BSD and System V Unix and many real-time operating systems. Operating systems using memory management must allow the user to map to the portions of the I/O page where the board registers are accessed. The packages will map the I/O page for operating systems using memory management.

The package is a library of subroutines which run under a shell on the graphics board and provide functions for the board. All characters are software defined patterns which are drawn in the graphics memory. The subroutine package supplies a variety of bit-mapped fonts including contemporary and typewriter styles in different weights and pitches. Two versions of CnP are included:

a) A hybrid version wherein a front end process running in the host computer interprets subroutine calls and directs commands to be executed by the 34020 on the graphics board. In some cases it is more efficient to directly execute these functions, so the 34020 is not used.

b) A board based version for standalone programs. The user links the CnP with an application program developed with Peritek's Program Development Package (compiler, assembler, and linker - see section 3.2). CnP and the application run entirely on the graphics board.

Other software shipped with CnP includes initialization, demo programs, 34020 downloader, and 34020 utilities. Most programs are supplied in source and executable and are written in C.

A complete list of the Graphics Subroutine Package library functions is available upon request from Peritek. A sampling of the functions is shown on the following page.

Table 3-2 Graphics Subroutine Package Library Routines

Initialization Functions	clear_screen, new_screen, init_grafix, init_palet, init_screen, init_text, init_video, init_vuport
3D Transformation Functions	copy_matrix, copy_vertex, init_matrix, perspec, rotate, scale, transform, translate, vertex_to_point
Text Output Functions	draw_char, draw_string
Text Attribute Functions	add_text_space, char_high, char_wide_max, get_ascent, get_descent, get_first_ch, get_last_ch, get_leading, get_width
Font Management Functions	get_font_max, install_font, select_font
Graphics Output Functions	bound_fill, bound_patnfill, draw_line, draw_oval, draw_ovalarc, draw_point, draw_polyline, draw_rect, fill_convex, fill_oval, fill_piearc, fill_polygon, fill_rect, frame_oval, frame_rect, patnfill_oval, patnfill_polygon, patnfill_rect, patnfill_convex, patnfill_piearc, patnframe_oval, patnframe_rect, patnpen_line, patnpen_ovalarc, patnpen_piearc, patnpen_point, patnpen_polyline, pen_line, pen_ovalarc, pen_piearc, pen_point, pen_polyline, styled_line, seed_fill, seed_patnfill
Pixel Functions	bit_expand, get_pixel, get_rect, move_pixel, put_pixel, run_decode, run_encode, zoom_rect

Table 3-2 Graphics Subroutine Package Library Routines (continued)

Graphics Attribute Functions	gets_patn_max, gets_pmask, gets_ppop, gets_psize, gets_transp, install_patn, select_patn, set_color0, set_color1, set_pensize, set_pmask, set_ppop, transp_off, transp_on, cmmw_command, cmmw_readmask
Color Palette Functions	cmmr_command, cmmr_readmask, hls_rgb, rgb_hls, rgbrd, rgbwrt, cmmrblk, cmmwblk, hlsrd, hlswrt
Viewport Functions	close_vuport, copy_vuport, cpw, get_vuport_max, move_vuport, open_vuport, select_vuport, set_cliprect, set_origin, size_vuport
Miscellaneous Functions	delay, lib_id, lmo, rmo, peek_breg, poke_breg, rep_pixel, wait_scan, xytoaddr, pan, panrl, zoom_vert, zoom_horiz
Double Precision Functions	acos, asin, atan, atan2, ceil, cos, cosh, cotan, exp, fabs, floor, fmod, frexp, ldexp, log, log10, modf, pow, sin, sinh, sqrt, tan, tanh
Array Conversion Functions	fix_to_float, fix_to_long, fix_to_short, float_to_fix, long_to_fix, short_to_fix
Serial I/O Routines	sio_break, sio_error, sio_iflush, sio_init, sio_oflush, sio_peek, sio_read, sio_write

3.6 PTERM Terminal Emulator

Peritek has written a terminal emulator for use as a simple interface where a console terminal is not available. It is not a VT100 emulator and doesn't support escape sequences - its functionality is at the level known as "dumb terminal emulator". It is, nevertheless, very useful. PTERM can be combined in PROM with PX Windows or CnP.

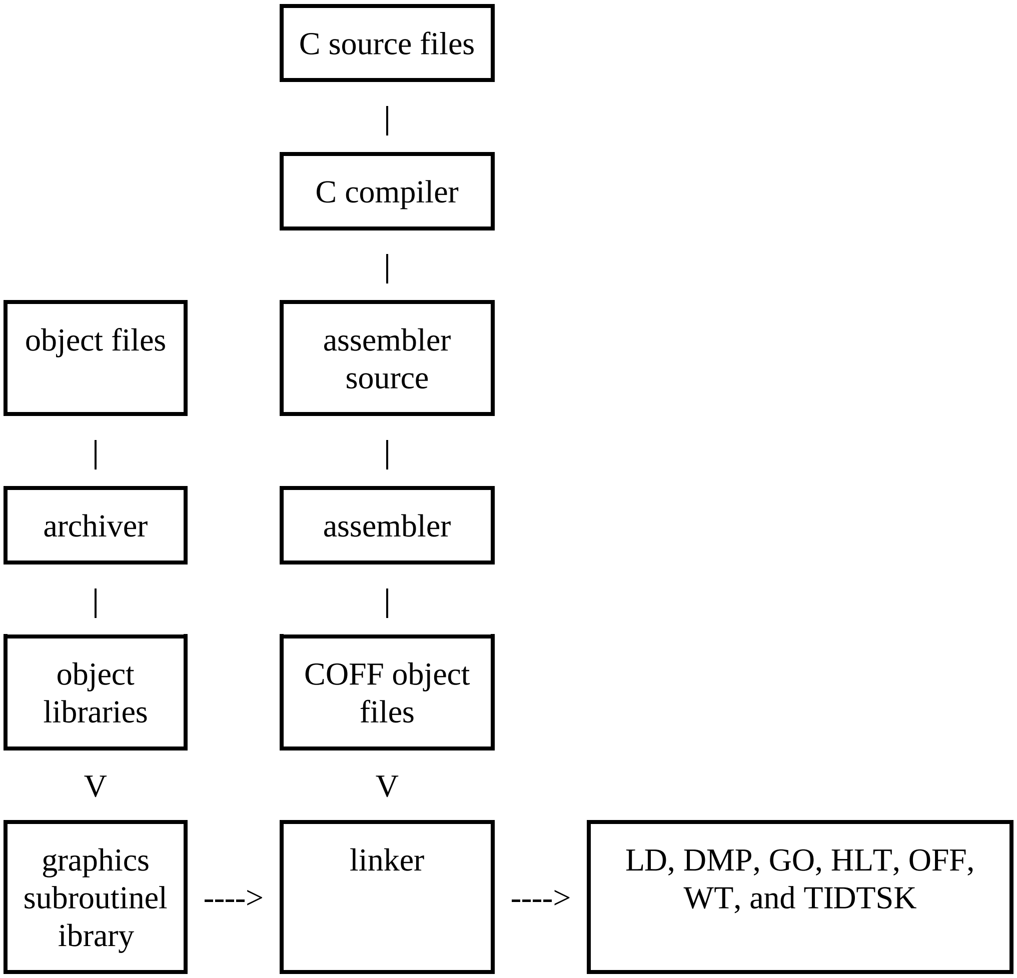
PTERM can be used to initiate an OS boot procedure. Once the OS is up, PX Windows can be started, whereupon PTERM ceases to function. Console terminal output can be redirected to an xterm window, By running a special program, PX Windows can be killed and PTERM restarted. There is, at this time, no hot-key function to permit dynamic switching.

A cable is connected between the host computer's console port and the graphics board Console Port. In addition, a Peritek keyboard must be connected to either the graphics board's LK401 or PC Keyboard port. The program runs the console link at 9600 baud, selects automatically between PC Keyboard or LK401 Keyboard and can be jumper configured for 7 or 8 bit data, and RTS/CTS or XON/XOFF (see Section 2.4.10).

3.7 Software Development Package

A SunOS-based C compiler, cross-assembler, and linker for user written 34020 applications is available from Peritek. Its general characteristics are described below. Contact Peritek for availability.

Figure 3-1 Software Development Flow



Features of the 34020 Development Tools

***** Standard Kernighan and Ritchie C Compiler with extensions - compiles standard C programs as defined by The C Programming Languge. This is a full-featured optimizing compiler, using advanced techniques for generating efficient, compact assembly code. The compiler supports these standard extensions: enumeration types, structure assignments, passing structures to functions, and returning structures from functions.

***** Assembly Language output is generated by the compiler from the C source. An interlist utility associates each C source line with its corresponding assembly code output. The Assembler translates the assembly language source output from the compiler into 34020 machine language object files. The Linker combines all object files into a single executable module.

***** The archiver allows you to collect a group of source or object modules into a library. It also permits you to modify a library by deleting, replacing, extracting, or adding members. It is functionally equivalent to ar (Unix).

3.8 Ancillary Programs

Some basic utilities are included with the Graphics Subroutine Package (CnP) which can be used to see that the board works. This section will tell you a little about these programs, but for complete information, please refer to Peritek's TMS34020 Software Manual.

Program Name	**Manual Section**	**Description**
Initialization	3.8.1, 5.4	Initialization tables
Install	3.8.2	Software installation examples.
VCnVINT	3.8.3	Initialize the graphics board.
VCnVTST	3.8.4	Test memories and control registers.
VCnVLD	3.8.5	Load and execute a 34020 demo programs
VCnVWT	3.8.6	Wait for a task to complete.
VCnVHLT	3.8.7	Halt a 34020 program.
VCnVGO	3.8.8	Restart a 34020 program.
VCnVOFF	3.8.9	Turn of the MEMON bit.
VCnVDMP	3.8.10	Formatted dump of 34020 memory.
TIDTSK	3.8.11	Formatted dump of COFF data.
--	5.4	34020 initialization table examples

3.8.1 Initialization Tables

Programs which initialize the graphics board are shipped in source and executable form. Initialization parameters that are used within some programs are provided as ASCII files. A list of tables is provided in Section 5.4. Both PX Windows and CnP are distributed with a host of tables appropriate for a wide variety of applications. Please contact Peritek if you have difficulty selecting a table or getting a good display.

3.8.2 Software Installation Examples

Be sure to consult the printed release notes which accompany your software before attempting to install or run any programs. Some of the programs described here or elsewhere in this manual **may not be included** in your distribution. Conversely, there may be programs included on the tape which are not mentioned at all in this manual. Both the PX Windows and Subroutine Package manuals have complete installation procedures.

While the recovery procedures described below should be sufficient to recover the files, please refer to the printed release notes which accompanied the media. The Unix distribution is made in TAR format tape. The following procedure is for installing the tape.

1) Insert tape
2) Create a directory on the disk where you want to install the tree.
3) Change the default directory to that directory
4) Activate TAR
   ```
   tar x
   ```
5) when finished, type
   ```
   mt offl
   ```
   to rewind and unload the tape.

3.8.3 VCnVINT

Purpose: To initialize CnP and clear all memories. When done, load a simple task into system memory and start the 34020. It is recommended that this program be included in the user's startup command file. If the graphics board registers and memory are not found at the expected addresses, VCnVINT gives an error message and exits.

VCnVINT was designed to run before the Graphics Subroutine Package is loaded into the board. VCnVINT has several command line options which supports specification of a non-standard CSR address and line buffer addresses. Refer to the sample scripts for the different OS's to see how to use this program. A number of standard initialization tables are supported and are supplied. See Section 5.4 for more information.

3.8.4 VCnVTST

Note: Contact Peritek for availability of this program. Supported only on VxWorks.

Purpose: To test the control registers, interrupts, TMS34020 Graphics System Processor chip (CnP), the primary and overlay graphics display memory, 34020 system RAM, DUARTs, cursors, and color map chips.

Description: VCnVTST is a menu driven program which allows the user to select any or all tests, and to change 34020 parameter registers interactively.

Three menus are supported: The master menu, the 34020 register menu, and the test menu. Since significant effort has been made to make VCnVTST "user friendly", the features of the program are pretty self explanatory. However, instructions are included in Peritek's TMS34020 Software Manual.

VCnVTST thoroughly tests the registers and RAM sections on the board. Bad results in some tests will prevent other tests from being run. A bad LAR, for instance, will prevent execution of the RAM diagnostics.

Tests are run on the control register block for read/write bits. The read/write bits in each register are tested by writing and reading back an incrementing pattern. If the pattern read back does not match what was written, an error message is printed with the bad bits set in a binary word.

A register uniqueness test is run on the read/write registers within each on-board device. A unique pattern is written to each register, with an immediate readback performed after each write to verify proper access. After all of the registers have been written into, another readback is performed to see that each register still has its unique pattern, thereby verifying that the registers are being addressed correctly.

Two diagnostic tests are performed on each RAM and the color and cursor maps. The first test, the RAM uniqueness test, tests RAM addressing by verifying that each address references a unique location. In this test a unique value is stored at each address in the memory region under test. As each location is written into, an immediate readback is performed to rule out RAM and data line failures. After every address in the memory region under test has had a unique value written to it, readback of the whole memory is performed. If a non-unique location is found (i.e., an addressed location does not contain what was originally written to it) the last address to access the location can be determined by the new data in the location. The first and last addresses that referenced that location are printed in an error message. Three passes are required for the GRAM and System RAM, first testing byte addressing, then word addressing, and finally, line addressing.

The second test, the RAM pattern test, makes several passes through the memory using different data patterns in an effort to detect pattern-sensitive failures. In each pass, a pattern is written to every word and then its complement is written to every third word. When the whole memory is filled, a readback check is made.

When an error occurs, the bad bits are OR'ed into a word (initially all zero's) and AND'ed into another word (initially all ones). The 16-bit word address is formed and is OR'ed and AND'ed in a similar manner. Testing continues until the entire memory has been checked with the current pattern. If any errors have been detected, this test is terminated (after completing testing with the current pattern) and an error message is displayed on the console device.

VCnVTST has command line options similar to VCnVINT. Please refer to the Peritek TMS34020 Software Manual for a description.

3.8.5 VCnVLD

Description: VCnVLD is used to load premade 34020 tasks to be loaded into the graphics board and executed. Numerous programs of the form **file.RAM** are supplied. VCnVLD has the command line options which allow you to specify the name of the task to be loaded and to force the loader to wait (normally VCnVLD does not wait for the 34020 task to complete before exiting). You also have the same non-standard CSR and line buffer options available in VCnVINT.

3.8.6 VCnVWT

Purpose: Wait for a task started by VCnVLD to complete. This program duplicates the second half of VCnVLD. If you didn't run VCnVLD with the **-wait** option you can "wait" later by running this program.

3.8.7 VCnVHLT

Purpose: Halt a program running in the 34020. This program is used to debug on board code. It flushes cache and halts the 34020.

3.8.8 VCnVGO

Purpose: Restart a program previously loaded and running in the 34020. This program is used to debug on board code and can used in conjunction with VCnVHLT. It restarts the 34020 at the point at which it was halted (**not at the beginning of the 34020 program**).

3.8.9 VCnVOFF

Purpose: Clears the MEMON bit in the VCnV CSR register. Normally this is not necessary, since all host programs clear this bit on termination. However, if the program did not finish normally, it may be necessary to run this program (especially if you have boards which share the same line buffer addresses).

3.8.10 VCnVDMP

Purpose: Dumps formatted portions of board memory to the terminal. This program is useful for dumping the 34020 register block or video memory. Output is similar to the Unix **od** command. Command line options allow on to specify address type and format and data size and format.

3.8.11 TIDTSK

Purpose: Provides to the terminal a formatted dump of COFF 34020 download images.

Chapter 4
Theory of Operation

4.1 Introduction

This chapter contains a somewhat detailed look at the proprietary parts of the graphics board design. Standard devices, such as color map chips and DUARTs are not covered here. Chapter 5 has some relevant information about the devices. Otherwise, we depend on the manufacturer's data sheet to provide complete information.

Section 1.2 contains a complete Functional Description. Please refer to that section before continuing with this chapter.

This chapter has the following sections:

4.2 System Architecture

As noted before, this manual deals with all three of Peritek's 34020-based graphics boards. The awkward places which result are in having to delineate differences between boards, while maintaining a coherent flow in the material. In this chapter, the overall board architecture and feature set is the same. The most significant differences arise out of the variety of display output options: 24 bit true color for the VCT-V, 8 bit color for the VCU-V and VCD-V, and digital output for the VCD-V. Referring to the block diagram appended to this chapter, it is evident that the VCD-V, VCT-V, and VCU-V graphics boards are divided into two main sections: the VMEbus interface section and the TMS34020 section.

VMEbus Interface

Rather than use a commercial VMEbus interface controller such as the VIC064 or the SCV64, Peritek uses a proprietary chip set which is tailored to the interface requirements of the 34020. Implemented in 3 high density AMD MACH FPGA devices, the chip set includes control signals for the VMEbus bus drivers, address decoders for the VMEbus, Control/Status Register (CSR), Line Address Register (LAR), Line Buffer Address Register (DBRADR), Extended Address Register (XAR), Interrupt Vector Address Register (IVAR), an interrupt controller, VMEbus/34020 arbitrator and a byte swapper (see Section 5.3).

Control Registers

The 4 word CSR/LAR group and a 1 KB line buffer are all in the A16 space. A control bit can be used to enable A24 operation for the line buffer when the VMEbus host doesn't support A16/D32 transfers. D32 capability is important because long word data transfers will go twice as fast.

The CSR provides basic control over the board, including 34020 reset, line buffer response enable, A24 enable, A32 enable, hardware byte swapper enable, and interrupt enable. The LAR selects which 1 KB section of memory or block of device registers is accessed through the line buffer. The line buffer mechanism is used instead of direct addressing because the internal memory capacity of the graphics board is in excess of 48 MB, which is a substantial amount of address space, one that is outside the reach of both A16 and A24 bus masters. Alternatively, the graphics board can respond to a 64 MB section of A32 VMEbus address space, which might be convenient for a disk controller. Note that Peritek software supports only the A16 and A24 space addressing modes except for multiprocessor arbitration on Sun and Motorola 188 systems.

The on-board interrupt controller supports an interrupt from the 34020 to the VMEbus host. Interrupt level is jumper selectable.

34020/VMEbus Host Interface

The VMEbus/34020 arbitrator allows the VMEbus to access not only the CSR group but also the 34020-side devices. The 34020 participates in the termination of those cycles, since it must synchronize them to its own bus activity. The latching bus transceivers are actually controlled by the 34020, which reads or writes them in conjunction with the completion of the cycle requested by the arbitrator.

34020 Functional Unit

The 34020 section of the graphics board is a unit unto itself. It includes display and 34020 (system) memory, writemask register, color maps (which one is installed depends on configuration), 2681 DUART (serial I/O), 8242PC keyboard controller, and local memory and device decoding. Really, the two sections have little to do with one another except for passing data back and forth - once loaded with a program and started, the 34020 can run independently of the VMEbus host. Except for the CSR group, the 34020 has complete control over the functions of the board. The VMEbus, going through the 34020 "host interface", also has ready access to those functions. The 34020 provides a very fast and efficient interface to the host, supporting byte operations and translating VMEbus 8, 16, and 32-bit accesses into the 34020's native 32-bit environment.

34020 Data and Address Buses

The 34020 has a multiplexed address/data bus (MAD) which supplies 32 bit data to memory and devices. The 34020 address is actually a **bit** address, not the more customary byte address. Thus, 34020 address line 5 corresponds to VMEbus address 2. 34020 address lines 0-4 are not used to address memory. 4 CAS lines are used instead to select 1 to 4 bytes of the data bus. The VMEbus address line 1 and Upper and Lower Data Strobes are used for byte selection.

The 34020 also has a separate multiplexed row and column address bus which is used for the dynamic RAMs and video RAMs, both of which have multiplexed address inputs. In order to ensure retention of the data in the VRAMs, each row of the 512 rows of data in the memories must be refreshed every 8 ms. They are refreshed using the CAS before RAS refresh mode, which is controlled by the 34020's DRAM refresh logic. This mode utilizes a refresh counter internal to the memory chip. When CAS is asserted before RAS (the opposite of the normal order), the memory chip executes a self-refresh cycle.

In general though, devices themselves (e.g. DUART, color maps) are only eight bits wide, so the device registers, while located on 32-bit boundaries, have at most 8 valid bits.

4.3 Graphics Board Clock Sources

There are several clock sources on the graphics board: a 40 MHz clock for the 34020 and VMEbus/34020 arbitrator, a programmable phase locked loop (PLL) pixel clock, a 14.7456 MHz reference oscillator for the pixel clock PLL with a divide-by-4 to provide a 3.6864 MHz clock for the DUARTs, and for the VCD-V (only) one or two fixed frequency pixel clock oscillators (24.576 to 110 MHz).

Phase Locked Loop (PLL) Clock

The VCT-V, VCU-V, and some versions of the VCD-V incorporate a programmable pixel clock oscillator (ICS1562-201AM) which allows the user to program virtually any frequency pixel clock up to more than 200 MHz. In fact, only the VCU-V is capable of operating at such lofty frequencies, and even then requires a special order BT468. The 1562 uses the 14.7456 oscillator as its reference clock to drive an internal phase-locked loop (PLL).

While a PLL clock is being phased into the VCD-V at the time of this writing, most VCD-V configurations use a fixed frequency oscillator. The video clock is connected to a GAL16V8-7 high speed PAL which is used as a programmable synchronous divider controlled by the MACH110-based horizontal control register. The function of the register changes somewhat, depending on configuration. See Sections V.7 and V.8 for detailed information. The divider output drives a counter in the MACH110 to provide shift and load clocks for the Video RAMs and blanking circuits.

34020 Video and Processor Clock Synchronization

The 34020, among all of its nice features, has separate processor and video clocks. It has internal synchronizers which make this work. The 1562 supports a "genlock" feature, which allows the pixel clock to be synchronized (locked) to external horizontal and vertical signals. This works in conjunction with the 34020 to provide a completely genlocked system.

4.4 VMEbus Interface

The VMEbus is an asynchronous bus, consisting of 32-bit bidirectional address and data busses, a 6-bit address modifier code, and 5 primary control lines.

The following discussion assumes a working knowledge of these busses. For detailed information concerning operation of the VMEbus, please refer to the VMEbus Specification C.1 (available from VITA, see Section 1.2).

The graphics board has two devices connected to the VMEbus: a bus address register (BAR) and 32-bit bidirectional data transceiver. The BAR is actually part of a MACH230, which latches the address bits and provides A16, A24, and A32 address decode ranges and block transfer requests as determined by the CSR programmable address decoder registers.

Address Decoding

XMEMON is connected to the A32 main memory address decoder. XMEMON comes up **off** on power-up, and prevents the graphics board from responding to A32 addresses until it is turned on. This may be never, since Peritek software doesn't use extended addressing except in multiprocessing Sun and Motorola PX Windows systems.

The valid address decoder provides 3 signals: IOREQ, CSRREQ, and BFREQ. IOREQ and CSRREQ provide decoding for the A16 space addresses (CSR and I/O window). In the case of the I/O window, MEMON must be set to enable board response and the I/O window can be enabled to appear in A24 space instead of A16. BFREQ is the decode for A32 space addresses (see Section 6.2 for VMEbus address assignments).

When

[(IOREQ*MEMON)
+ CSRREQ
+ (BFREQ*XMEMON)*!IACK] * AS * DSn = 1

[where **+** means logical OR, and ***** means logical AND]

then, VREQ is set. It can be assumed that a valid address has been clocked into the address register and data is already set up on the VMEbus (for write) or the CPU is waiting to receive data from the board (read). VREQ is used to request control of the board by the VMEbus. The operation of the arbitrator in this case is described in the Section 4.5.

The low 10 bits of the BAR are always active on the graphics board's internal CMA bus. If the VMEbus address is an A16 space address, the LAR is gated onto CMA 10-25. If, instead, the address is an A32 address, bits 10-25 of the BAR are gated onto the CMA bus.

Data Bus Transceivers and the Byte Swapper

The 74BCT16652 BiCMOS high drive low power registered bidirectional bus transceivers provide a 32-bit path for data transfers between the VMEbus processor and the on-board devices during programmed I/O cycles. Since the board, as a VMEbus slave, must support D32, D16 and D8 bus transfers, the byte swapper, which sits in the data path, is used to pass 32 bit data straight to the 34020 side or to multiplex data from the high data bits to the low data bits for D16 and D8 transfers. Note that accessing the CSR group with D32 instead of D16 will result in bytes 0 and 1 undefined.

The MACH435 FPGA controls the state of the 74BCT16652s and the byte swapper when the VMEbus is master. The 74BCT16652 output register, which drives the VMEbus, is edge triggered. It is clocked at the end of a read cycle to hold data read from an on-board device. DTACK*READ allows the 74BCT16652s to drive the VMEbus. The 74BCT16652 inputs, which receive data from the VMEbus, latch the data at the end of the write cycle. The 34020 will read the data out of the transceivers later. The 34020 will delay its response if a second write occurs before the 34020 has finished the first one.

4.5 VMEbus Interrupt Controller

The interrupt controller FPLA is a D08 RORA (Release On Register Access) interrupter. It may be used with a D08 interrupt handler, which is the most common interrupter type, and includes CPUs that use the "VIC068" chip.

If the board is not requesting an interrupt and it receives an IAKI it will drive IAKO. As IAKI is internally synchronized it may take up to two 34020 clocks (50 ns) to drive IAKO after receipt of IAKI. IAKO is asynchronously reset (immediately negated) upon the negation of AS, as required by the VME specification. When the 34020 sets its HINT interrupt flag, and the DEVINTEN in the CSR is set, the board will drive one of IRQ1 through IRQ7 lines, depending on the IRQ jumper option selection. The VME interrupt handler will then drive (true) VIACK, IACKO, and AS, and drive (true or false) A01-A03, LWORD, DS1, and DS0, depending if it wants a D32, D16, or D08 Status ID. A01-A03 reflect the interrupt priority the interrupt handler is acknowledging. When the interrupt controller receives these signals it compares A01-A03 with the vector priority select jumpers (VPSEL0-2). If there is not a match it will drive IAKO as outlined above. If there is a match it will cause a read of the 8 bit Interrupt Vector Register (IVAR). The board will drive the contents of the IVAR into bits 0-7 of the transceivers. Bits D08-D31 are not driven by the board. They are pulled high by the VMEbus terminators. The interrupt cycle then terminates as a normal read cycle. The interrupt handler uses the vector number read from the board to point to an exception routine address. As the interrupter is a RORA device, the exception routine should negate (or toggle) the DEVINTEN bit in the CSR. The exception routine then executes its function and ends with an RTE (return from exception) instruction.

4.6 System Arbitration

One of the most important pieces of logic on the graphics board is the system arbitrator. The function of the arbitrator is to allow the VMEbus to access the non-34020 related functions on the graphics board and to provide handshaking between the 34020 and the VMEbus for 34020 related functions.

The graphics board has four addressable registers (CSR group) on the VMEbus side of the board and 8 devices on the 34020 side (34020, writemask register, color map/cursor controller, 2 DUART serial I/O chips, PC Keyboard controller, zoom control register, and for the VCD-V, the digital lookup table). Although the VMEbus can access the 34020 side devices, the 34020 cannot access the CSR group. Separate device address decoders are therefore required to select the 34020 devices. Note that because they are all on the 34020 "side" of the board, the arbitrator is not required for 34020 access to display or processor memory or any device (except the CSR group, which the 34020 **can't** access anyway). Arbitration is required for VMEbus access to any part of the board.

Control Register Decoding (CSRREQ)

CSRREQ is used to request non-34020 related functions (CSR group). When the arbitrator receives a valid request it grants access by asserting VSTRB and gating the address and data onto the board's internal busses. Once the operation is complete (about 100 ns) DTACK is set, terminating the VMEbus request.

Line Buffer Decoding (IOREQ)

IOREQ is used to request 34020 related functions. A valid LAR (see Section 4.2) and an offset into the line buffer address block will select a unique address in the 34020 address space. The following diagram illustrates the mapping.

CMA	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2
LAR	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
LAD	29/28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5

Notes: CMA is VMEbus address, LAR is Line Address Register and LAD is 34020 address bus - note: LAD 31 and 30 are tied high, 29 and 28 are wired together.

34020 Host Interface Arbitrator

When the arbitrator receives the request from the VMEbus, it asserts the 34020 control lines (HCS, HWR, HRD) and waits until the 34020 responds, typically within 100 ns. If data is being read from the 34020 side, then that data is loaded into the VMEbus/34020 32-bit bus transceivers when the 34020 responds. If data is being written to the 34020 side, the bus transceivers are latched into the 34020-side 32-bit address/bus, where it is loaded directly into memory or a device.

Once the reply phase is entered, the arbitrator sets VMEbus DTACK. When VMEbus signals DS0 and DS1 go false, (which indicates that read or write has become false), DTACK is negated.

VMEbus Block Mode

The VMEbus supports a high-speed data transfer method known as block mode. According to the specification, up to 64 contiguous long words may be transferred using the technique of implied addressing. **However, the graphics board can support as many transfers as you wish.**

Using the Address Modifier control lines, the bus master signals its intent to initiate block transfers. It supplies a memory starting address and the bus slave (i.e. the graphics board), using the 34020's block transfer function, supplies its own addresses, which increments after each memory cycle. This allows the bus master to skip the address output cycle, which can result in significantly higher data transfer rates.

4.7 Display Memory

The memory devices used in the display memories are expressly designed for high speed graphics applications. These devices are called video RAMs (VRAMs). They are like ordinary DRAMs, but they also contain an internal 512 x 8 line buffer. During a special data transfer cycle, an entire row address worth of data is loaded into this line buffer. The VRAMs have a mode control input (DT/OE), which is used to trigger a data transfer. When DT/OE is active when RAS is asserted, a data transfer cycle occurs. The row address selects a line of data, and the column address selects the starting position within that line. The data are then shifted out by a serial clock, 8 bits/clock appearing at the outputs.

Since the internal VRAM buffer needs to be loaded only once per line time, the VRAM is available for random access operations at all other times. There is a small additional overhead time for memory refresh, which occurs about once every 15 us. Thus, the availability of the VRAM to external access is about 95\% as compared with approximately 35\% for normal DRAM designs.

The VMEbus accesses the memory via the 34020 host interface which includes two 74BCT16652 bus transceivers. The 34020 accesses the memory via its 32 bit data bus and multiplexed (row and column) address bus. Access is controlled by the 34020.

The graphics board display memory size is a function of the board type and the display configuration.

VCD-V and VCU-V Pixel Size

On the VCD-V and VCU-V, pixel memory size is 8 bits for both primary and overlay. The overlay actually only uses the low 4 bits because that is all the color map will use. Separate address spaces are allocated for the primary and overlay memories.

VCT-V Pixel Size

On the VCT-V, pixel memory size is 24 bits for primary and 8 bits for overlay. The overlay actually only uses the low 4 bits because that is all the color map will use. The primary and overlay share the same address space: the primary uses the low 24 bits of each word, and the overlay uses the top 8 bits. The writemask is very useful on the VCT-V for this reason.

VCD-V Display Memory Size

For a VCD-V with a 640 x 480 display, the minimum video memory is 1 MB of byte-addressable memory, expandable to 4 MB. Each byte is a pixel, and there are 4 pixels to each 34020 long word. The overlay memory occupies a similar amount of address space, but only uses the low 4 bits of each byte. The other 4 bits have valid data, but are not used.

VCU-V Display Memory Size

For a VCU-V with a 1280 x 1024 display, the minimum video memory is a 2 MB of byte-addressable memory, expandable to 8 MB. Each byte is a pixel, and there are 4 pixels to each 34020 long word. The overlay memory occupies a similar amount of address space, but only uses the low 4 bits of each byte. The other 4 bits have valid data, but are not used.

VCT-V Display Memory Size

For a VCT-V with a 1280 x 1024 display, the minimum video memory is a 8 MB of byte-addressable memory, expandable to 16 MB. Each 32-bit 34020 longword is a pixel position, where bits 0-23 are primary and bits 24-31 are overlay. The overlay memory uses the low 4 bits of each byte. Bits 28 and 29 are used for window type table, and bits 30 and 31 are valid but not used.

4.8 System Memory

The 34020 has its own private 32-bit memory which is independent of the video RAM (VRAM) configuration and timing. Obviously, the writemask register is not used with system memory, because it would cause unpredictable operation of the board. The system memory resources for the graphics board are comprehensive, and consist of four components: SIMM sockets for field upgradability and Flash EEPROM. Section 5.3.4 has address ranges and memory maps for the DRAM and EEPROM memories.

SIMM Sockets

The graphics board is built with SIMM sockets because it allows considerable manufacturing flexibility. 1 MB, 4 MB, 8 MB, 16 MB, and 32 MB units can be fit into the same slot, with only jumper changes required to accommodate the different capacities. Peritek makes its own SIMMs because most commercial vendors do not make modules short enough to fit into the VMEbus form factor. The SIMM module must not exceed .95 inches in overall height.

Flash EEPROM

The Flash EEPROM uses four 150 ns 32-pin PLCC 8-bit wide parts, for a maximum capacity of 2 MB. The graphics board is designed to permit on-board reprogramming. However, as yet no software has been released (to customers) to support this.

4.9 Programmed Logic Devices

Virtually all logic on the graphics board is contained in commercial parts such the 34020, bus buffers, and programmed parts. The board uses numerous Programmed Logic Devices (PLDs). This section will briefly describe each part used and then outline the functions implemented by them.

AMD MACH Field Programmable Gate Arrays (FPGA)

The lion's share of the programmed parts are MACH pin parts, where a MACH210, MACH230, and MACH435 parts are used. The are characterized by a high degree of flexibility. All parts are EEPROM technology, which permits easy reprogrammability.

The MACH parts are essentially 26V16 building blocks linked by a partially implemented crossbar switch built into a single package. The MACH210 part has two blocks and the 230 has four blocks. Registered and asynchronous I/O pins abound, and I/O and buried register per pin capability, global clocks and resets are included. The MACH435 adds input registers.

PAL22V10 Field Programmable Gate Array (FPGA)

The 22V10 part contains a general AND-OR array, with 8 OR terms that can drive a particular output. Each output can be programmed for active high or low, and can function as a tristate output as well. The parts are EEPROM technology, which permits easy reprogrammability.

Table 4-1 VCT, VCU, VCD Common PLD Device Summary

Part Number	Device Type	Description
VCDTU1	MACH230	Provides LRDY and BUSFLT to 34020. Supplies address decoders for color maps, cursors, HSP, zoom register, PC Keyboard, serial I/O, SCSI (VCT/VCU only), and DLUT and pixel mux (VCD only). Has input jumpers for DRAM size and 8 bit/pixel (VCD, VCU) or 32-bit/pixel (VCT) VRAM. Supplies RAS lines for graphics and system memory. Decodes LAD0-3 for page mode writes, refresh, VRAM special functions (shift register load, writemask, block fill, color register,), and FPU select. Provides modified SF line to VRAMs for correct special function operation. Controls 74BCT16652 hidden writemask register. Provides chip select and special address decoding shift for autobooting PROMS.
VCDTU10	MACH230	Functions as a 32-bit registered bus transceiver with byte, word, and long data swapping. All data passing between VMEbus and 34020 side goes through this device.
VCDTU11	PALC22V10	Supplies CAS lines for display and system memory. Decodes LAD0-3 to support VRAM block fill. Different versions are required for VCD, VCT, and VCU.
VCDTU234	MACH435	A16, A24, A32 space VMEbus decoder, CSR bits 0-15, LAR bits 0-15, A16/A24 DBR (line buffer base address) bits 0-13, XAR (A32 space base address) bits 0-5, VEC (interrupt vector) bits 0-7. Outputs drive 34020 host address lines CMA 10-25. Data lines DA0-DA15 are I/O's for the registers and output bits 0-7 of the interrupt vector. Generates byte swap and bus control for VCDTU10. Enables A24 and A32 VME block transfers.
VCDTU78	MACH230	VMEbus arbitrator and bus control. Address modifier decoder. Generates 34020 autoboot/CRTCON, byte selects and chip select. Control VMEbus 74BCT16652 bus transceivers. Interrupt arbitrator. Transmits 34020 interrupt to VMEbus. Buffers VMEbus address lines 2-9.

Table 4-2 VCT, VCU, VCD Unique PLD Device Summary

Part Number	Device Type	Description
VCTUV19	MACH210	Used for VCT and VCU. Zoom control register. Provides ICS1562 3-wire control interface. Provides programmable polarity for HSYNC and VSYNC. Supports external sync/genlock function. Controls blanking, VRAM shift clocks, and VCLK. VCT version includes support for interlaced cursors.
VCTU9	MACH210	Used for VRAM block write operation. Takes the two low order 34020 low address lines and CAS lines and recodes them according to the VCT or VCU memory architecture to correctly select up to four pixels in a VRAM for simultaneous writing from the VRAM color register.

Table 4-3 VCD Unique PLD Device Summary

Part Number	Device Type	Description
VCDV19	MACH230	Used for VCD. Zoom control register. Provides GAL (crystal clock) and ICS1562 3-wire control interface. Provides programmable polarity for HSYNC and VSYNC. Supports external sync/genlock function. Controls blanking, VRAM shift clocks, and VCLK. Includes support for interlaced cursors. Contains multiplexer and latches for DLUT output for 1, 4, and 8 bit/pixel digital output.
VCDV5	MACH130	Pixel multiplexer. Latches 4 pixels (32 primary and 16 overlay data bits) and pipes the pixels out one at a time to the SRAM and BT482. Latches six 34020 multiplexed address (RCA) lines and presents a 12-bit address to the SRAM for read/write by the 34020 or VMEbus host CPU.

Figure 4-1 VCD-V Block Diagram

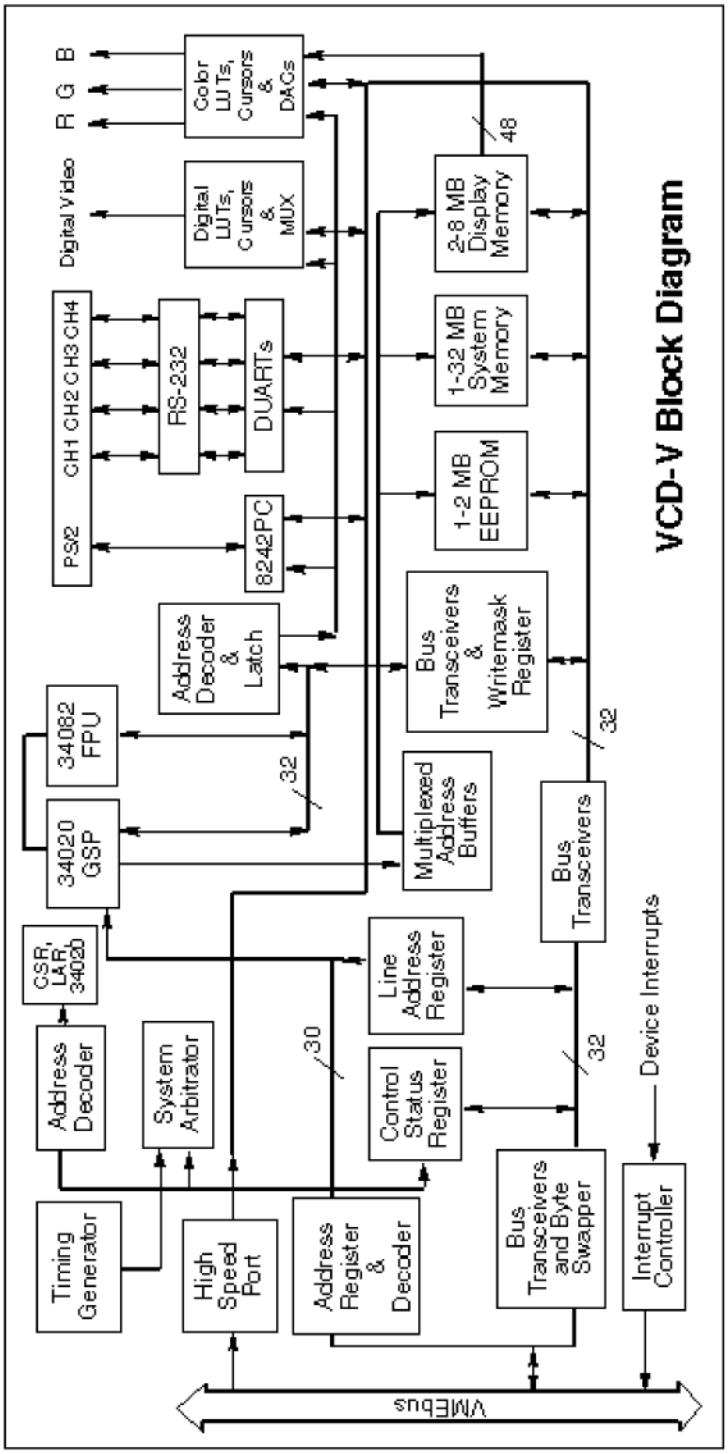


Figure 4-2 VCU-V Block Diagram

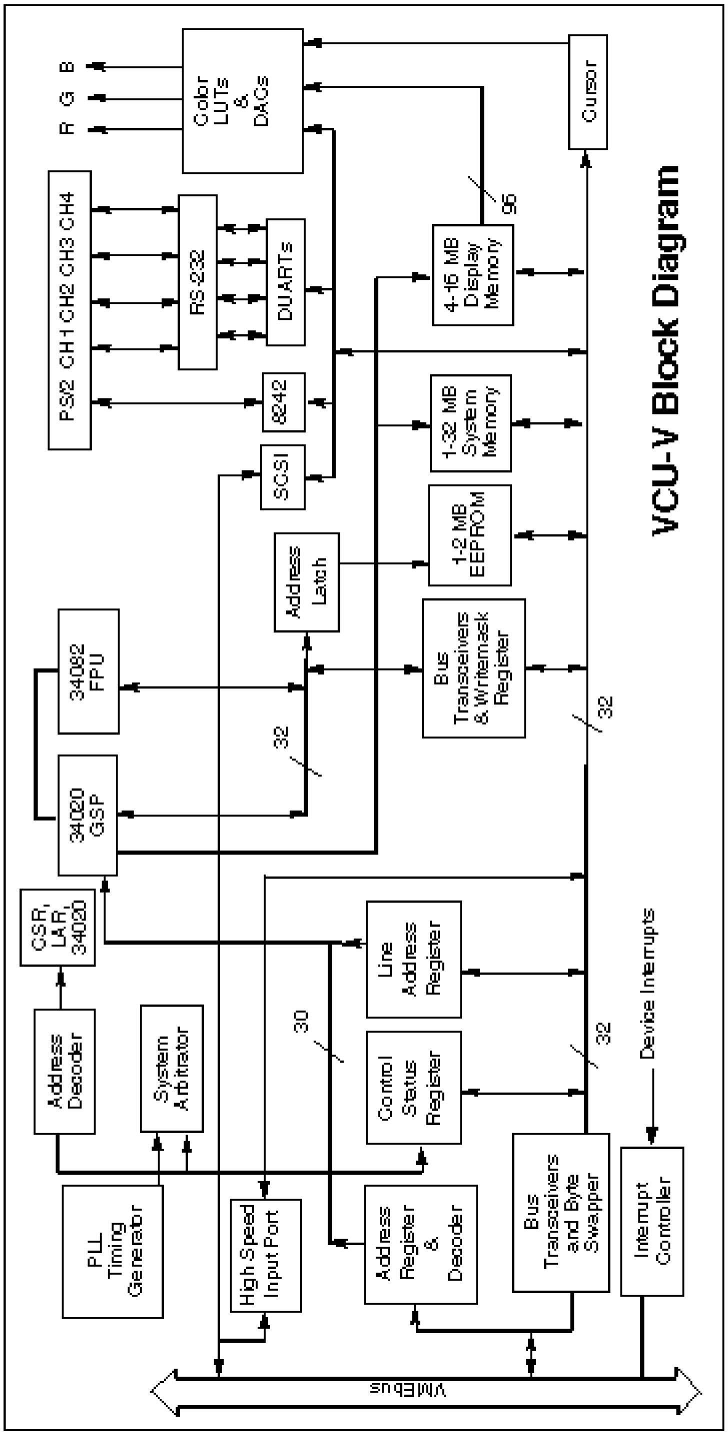
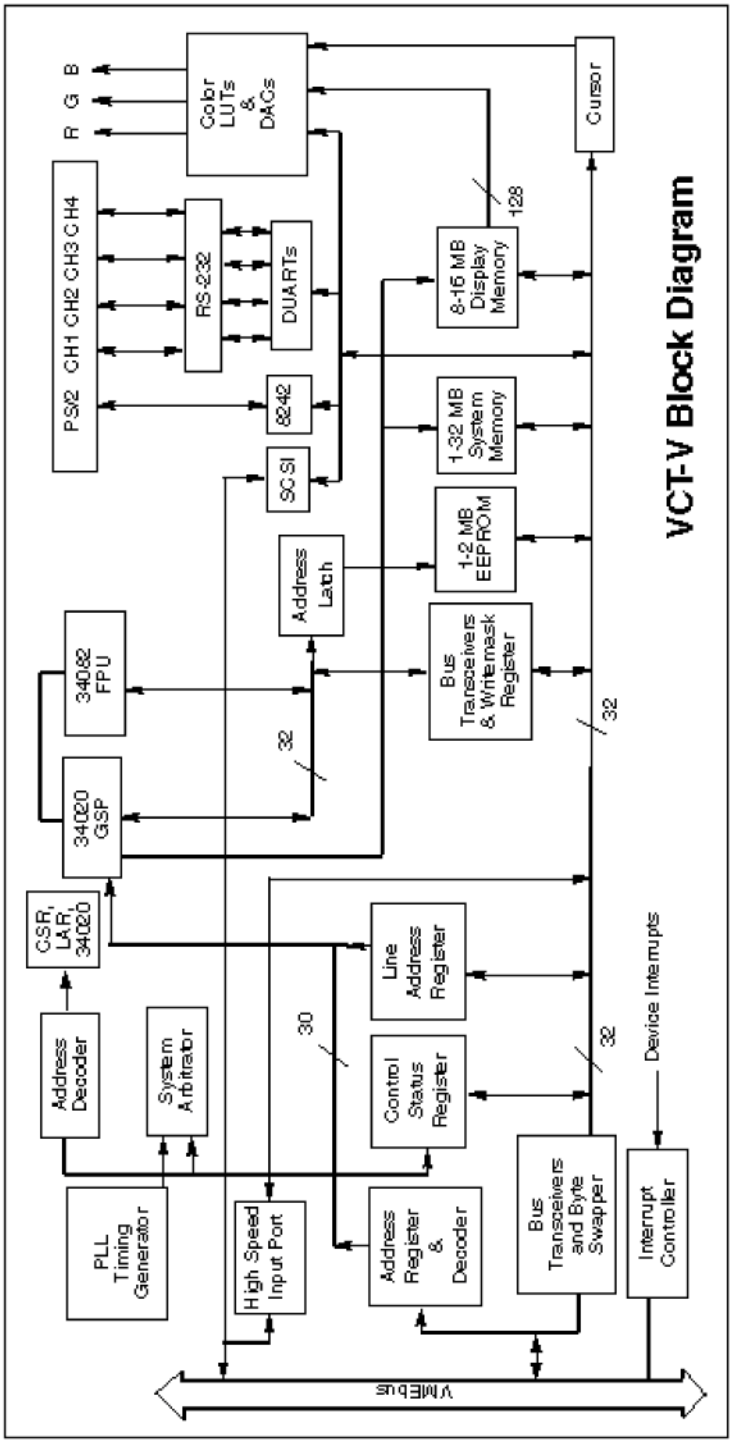


Figure 4-3 VCT-V Block Diagram



Chapter 5 Programming On-board Devices and Memories

5.1 Introduction

As with the other chapters, this one covers the VCT-V, VCU-V, and VCD-V. Most of the features are common, so the number of exceptions does not get out of control. In the interest of reducing superfluous information, and operating on the assumption that most users have either PX Windows or CnP, this chapter has undergone some heavy duty editing. In addition, the data sheet extracts which have heretofore been appended to the manual have been removed, as they are little used. They are still available upon request.

This chapter covers the special programming features of the individual devices used on the graphics board. It is intended to supply information unique to the board or to the application of a particular chip. Section 1.2 provides a list of appropriate publications which include manufacturer's data sheets and manuals.

Peritek offers a variety of software to support the VCD-V, VCT-V, and VCU-V in both Unix and real-time environments. Software includes demo, test and initialization programs, a very comprehensive Graphics Subroutine Package (generically CnP), and an X Windows X11R6 server (PX Windows). These offerings are covered in detail in Chapter 3.

Note

Please read these sections **before** starting on this chapter:

Section 1.2	Functional description of the VCD-V, VCU-V, and VCT-V graphics boards.
Sections 2.2-5	Installation.
Section 2.4	Jumper options.
Chapter 3	Summary of software support from Peritek.
Chapter 4	Theory of operation

This chapter includes the following sections:

5.2 VMEbus and Control Registers

The control registers and I/O window lie in a region of the VMEbus address space reserved for peripheral devices and is usually to be found in A16 space. The I/O window can also be placed in A24 space. Multiuser operating systems (i.e. Unix) do not automatically allow a user to access the VMEbus or physical memory. **All Peritek software (including CnP and PX Windows) include special mapping calls to give access to that part of the VMEbus where the board addresses are located.**

The VMEbus interface is implemented as a 1024 byte byte-addressable raster line buffer (DBR) and a 4 word CSR group (16-bit word, long word boundaries, VMEbus bits 0-15) in A16 space (the DBR can also be in A24 space - see Sections 5.2.1 and 5.2.4). For the sake of compatibility over numerous CPUs and OS's (or as one might say, the lowest common denominator) A16 space is the most general, and there is a minimal performance impact for using the line buffer. Thus Peritek software supports A16/A24 space addressing only. The standard addresses are shown in Section 2.4.1. For linear address access to the entire board, it can also respond to a 64 MB byte-addressable section of the 32-bit VMEbus memory map, Contact Peritek for assistance in determining when it is appropriate to use A32 space.

High speed VMEbus block transfers are supported for accesses to the A24 and A32 (64 MB block). Using the 34020's implied address mode, this permits up to 64 long words to be transferred over the VMEbus with only one address cycle. See 4.5 for more information.

Hardware byte swapping is now included in the graphics board. This can be used to advantage in transferring large blocks of data between the big-endian VMEbus and the little-endian graphics board (see Section 5.3 for more information.

Note

When byte swapping is enabled the 1KB line buffer expands to 4KB and the 64 MB block expands to 256 MB.

The CSR group consists of 4 registers:

Relative Offset	Register Mnemonic	Description	Section Reference
0	CSR	General Control	5.2.1
4	LAR	Line Address Register	5.2.2
8	XARADR	A32 Address Match Register	5.2.3
	DBRADR	A16/A24 Address Match Register	5.2.4
C	VECADR	Interrupt Vector Address Register	5.2.5

The XAR/DBRADR and VECADR registers allow all VMEbus address areas to which the board responds to be programmable except for the CSR base address, which is (necessarily) jumper selected from 16 different combinations (see Section 2.4.1).

Note

CSR group registers should be accessed as words, not long words, because the high word will not read back useful data.

5.2.1 Control/Status Register (CSR)

Table 5-1 CSR Bit Summary

Bit	Mnemonic	Function	R/W	Reset
15	spare	was BIGEND, now reads back 0	no	no
14	REVFLAG	was r/w, now read back set	no	no
8-13	spare	not used, reads back 0	no	no

7	A24EN	clear = A16 DBR access,
		set = A24 DBR access
			yes	sysreset

6	CRTCON	turns on the 34020	yes	sysreset
5	MEMON	enables the DBR addresses	yes	sysreset
4	XMEMON	Enables 32-bit address response.	yes	sysreset
3	A1624SWAPEN	Enables A16 and A24 swap mode	yes	sysreset
2	VINTEN	VMEbus interrupt enable	yes	sysreset
1	A32SWAP	Enables A32 swap mode	yes	sysreset
0	XARSEL	Select XAR register access	yes	sysreset

Table 5-2 CSR Bit Definitions

REVFLAG used by software to determine that this is a board with PC Keyboard and byte swapper capability. On older boards this bit was r/w. From VCT/VCU Rev 4 and VCD Rev 2 it reads back set.

A24EN allows the 1K DBR address range to respond in A24 space. The CSR is still only addressable in A16 space.

CRTCON turns on the 34020. It must be set **before** the 34020 has been loaded with its timing parameters (See Section 5.4). When CRTCON is reset, the 34020 internal register set is held cleared.

MEMON allows the board to respond to the 1KB DBR (line buffer) address range.

XMEMON allows the board to respond to the 32-bit address range.

A1624SWAPEN Enables byte swapping in the DBR A16 or A24 address range. **Note that when this bit is set, the DBR address range expands to 4 KB (from 1 KB).**

VINTEN allows the 34020 on the board to interrupt the VMEbus. When entering the interrupt service routine, VINTEN should be cleared. To avoid spurious interrupts, be sure that the 34020's interrupt request flag has dropped before reenabling VINTEN.

Interrupts are level sensitive, i.e. if VINTEN is enabled after a device interrupt has been asserted, the VMEbus will be interrupted immediately. The VMEbus interrupt request goes away only if the VINTEN is cleared. Device interrupts are latched if VINTEN is on.

A32SWAPEN Enables byte swapping in the 64 MB window's A32 address range. **Note that when this bit is set, the address range expands to 256 MB (from 64 MB).**

XARSEL selects the XAR register for access through the dual function XAR/DBRADR address match registers (CSR group, offset 8). See Section 5.2.4.

5.2.2 Line Address Register (LAR)

The LAR selects a memory or device register group on the board and permits a 1K Byte segment to be accessed through the DBR. Fully configured, the board contains a number of programmable devices, each of which has between 2 and 16 control registers. The board uses spare LAR values (0 and 400) to permit access to all of these devices through the DBR line buffer. **See Section 5.2.3 for A32 addressing information.**

Note

The 34020 must be initialized prior to accessing memories or devices.

Table 5-3 LAR Bit Definitions

LAR	Memory Selected
0	34020 internal device registers. See Section 5.3 and 5.4 for more information.
400	Device Buffer - Color map chip(s), DUARTs, PC Keyboard controller, zoom register, cursors, and SCSI (**VCT/VCU**).
800	High Speed Data input port (HSP) - special order only
C00-CFF	Digital Lookup Table (**VCD-V** only)
2000-2FFF	Up to 2 MB of Flash EEPROM, using four 512 KB devices.
Video RAM	***VCD-V and VCU-V***
4000-5FFF	Primary (8-bit) graphics RAM - 1024 pixels for each LAR value. 1K x 1K pixels is 1 MB or 1024 LARs. Memory is byte addressable and writemask applies.
6000-7FFF	Overlay (4-bit) graphics RAM - 1024 pixels for each LAR value. The data is valid in each byte, but only the 4 low bits are used. Memory is byte addressable and writemask applies.

Table 5-3 LAR Bit Definitions (continued)

Video RAM *VCT-V*

LAR	Memory Selected
4000-7FFF	Graphics RAM - bits 0-23 are primary (RBG) true color memory, bits 24-29 are overlay and window type table, bits 30-31 are r/w but not used. 256 pixels for each LAR value. 1K x 1K pixels is 4 MB or 4096 LARs. Memory is byte addressable and writemask applies.

Movie Memory RAM *VCD-V, special order*

1K bytes for each LAR value. Used only on the VCD-V for storing off-screen images. Special routines in the Subroutine Package support its use. Memory is not displayable. Byte addressable. Memory capacity is 8 MB or 16 MB.

Memory Installed	LAR ranges
8 MB	8000-9FFF
16 MB	8000-BFFF

System RAM

1K bytes for each LAR value. Used for the 34020 program store. Memory is not displayable. Byte addressable. 1 MB of memory is 1024 LARs. Memory capacity is expandable to 32 MB. See Section 5.3.4 for information on 34020 memory access.

Memory Installed	LAR ranges
1 MB	FC00-FFFF
4 MB	F000-FFFF
8 MB	E000-FFFF
16 MB	C000-FFFF
32 MB	8000-FFFF

5.2.3 A32 Address Map and the XARADR Address Match Register

The graphics board has an extended addressing function which allows a VMEbus host to linearly access the board through a 64 MB block in A32 space. You can access all on-board devices and memory through this block. Except for PX Windows multiprocessor drivers, Peritek software does not use A32 access.

In order for the A32 space on the board to respond the XARADR register must be set up with a valid address, the 34020 must be initialized, and CSR bit **XMEMON** must be set.

Caution

Do not set XMEMON until ***after*** the XARADR has been set up.

The hardware byte-swapping function can be enabled for A32 space. You must first set the **A32SWAPEN** bit in the CSR. **Note that the A32 space usage increases to 256 MB when A32 byte swapping is enabled.** The reason for this is revealed in Section 5.3.5, which contains detailed information on the byte swapping.

The XARADR/DBRADR Address Match Registers are programmed through a single register in the CSR block. Which register is accessed through the register slot is controlled by the **XARSEL** bit in the CSR.. When XARSEL = 1, the XARADR address match register is selected.

Table 5-4 XARADR Address Match Register

A32 Space Address Bit	XARADR Data Bit	R/W	Reset
26	0	yes	SYSRESET
27	1	yes	SYSRESET
28	2	yes	SYSRESET
29	3	yes	SYSRESET
30	4	yes	SYSRESET
31	5	yes	SYSRESET
--	6-31	reads 0	no

Note

When A32SWAPEN is set, XARADR 0 and 1 are don't care.

5.2.4 A16/A24 Address Map and DBRADR Address Match Register

The graphics board's A16/A24 (I/O/standard) addressing function allows the board to respond for the line buffer (DBR) in A16 (or A24) space. In order for the A16/A24 space on the board to respond the DBRADR register must be set up with a valid address, the 34020 must be initialized, and CSR bit **MEMON** must be set.

The only reason for selecting A24 space is if your CPU doesn't support D32 (long word) accesses in A16 space. You will pay a small (10-20%) performance for running D16 only. Peritek software must be explicity told to use A24 space, because the A24EN bit in the CSR must be set. Note that the 4 register CSR block can only respond in A16 space. To enable A24 response, turn off MEMON, set the DBRADR to match the upper 14 bits of the A24 address, then set both MEMON and A24EN. The hardware byte-swapping function can be enabled for A16/A24 space. You must first set the **A1624SWAPEN** bit in the CSR. **Note that the A16/A24 space usage increases to 4 KB when byte swapping is enabled.** The reason for this is revealed in Section 5.3.5, which contains detailed information on the byte swapping.

The XARDBR/DBRADR Address Match Registers are programmed through a single register in the CSR block. Which register is accessed through the register slot is controlled by the **XARSEL** bit in the CSR.. When XARSEL = 0, the DBRADR address match register is selected. Remember that MEMON in CSR also has to be set.

The size of the DBRADR register varies is a function of A24 and byte-swapping (see table).

Table 5-5 DBRADR Address Match Register

A16/A24 VMEbus Address Bit	DBRADR Data Bit	R/W	Reset
10	0	yes	SYSRESET
11	1	yes	SYSRESET
12	2	yes	SYSRESET
13	3	yes	SYSRESET
14	4	yes	SYSRESET
15	5	yes	SYSRESET
16	6	yes	SYSRESET
17	7	yes	SYSRESET
18	8	yes	SYSRESET
19	9	yes	SYSRESET
20	10	yes	SYSRESET
21	11	yes	SYSRESET
22	12	yes	SYSRESET
23	13	yes	SYSRESET
--	14-31	reads 0	no

Note

When A1624SWAPEN is set, DBRADR 0, 1 are don't care.
When A24EN is clear, DBRADR 6-13 are don't care.

5.2.5 VECADR Interrupt Vector Address Register

The interrupt vector address which the board supplies to the VMEbus during an interrupt acknowledge cycle is also programmable. Now, it should be pointed out that this is not literally the address to which the CPU will vector, but a 1 of 256 index, which is left shifted 2 bits by the CPU and **then** used as the vector address. Bits 0-7 in this register correspond to vector addresses in the range 0 to 1020. Remember that VINTEN in CSR also has to be set before you will get an interrupt from the board. Also, some processors require that the low 2 bits of VECADR be 0 to vector correctly. Bits 8-15 in VECADR read zero. Sections 2.4.2, 2.4.3, and 5.16 have more information on interrupts.

Table 5-6 Interrupt Vector Address Register

VECADR Data Bit	VMEbus Address Bit
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

5.2.7 VMEbus Block Transfers (BLT)

As of VCD-V FAB REV 2 and VCT-V/VCU-V FAB REV 4, the BLT function has been changed and software which uses it must also be changed to continue to use it. The BLT data transfer now supports **D32 only**. BLT is now supported for both A24 and A32 accesses (A16 BLTs are not defined). The 34020 itself generates the addresses, so the HINC bit (12) in the TI HSTCTLH register must be set, otherwise the same location is accessed repetitively. This bit can be left on all the time. If HINC is always on then non-BLT reads will be faster if sequential, possibly slower if random. If HINC is always on then non-BLT writes will be unaffected if HPFW[HSTCTLH] is clear (default), possibly slower if it is set.

Note

HINC must not be set on pre-Rev 4 VCT/VCU or Pre-Rev 2 VCDs.

5.2.8 Device Register Access

The board devices (and memory) are accessed through the 1 KB line buffer in the A16/A24 space. The 34020 registers are available when LAR = 0. See Section 5.3.7 for 34020 side addressing of devices and memory.

The color maps, zoom register, PC Keyboard controller, HSP interface, SCSI (VCT/VCU) and 2681 DUARTs are accessible when LAR = 400. Although most of the devices are just 8 bits, they are placed on long word (4 byte) boundaries, which simplifies addressing by the 34020, which controls their addressing. Having initialized the 34020 and set the CSR MEMON bit, these registers appear as shown on the next page:

Table 5-7 VMEbus Side Device Buffer

Selected	Relative		See	
by LAR	**Address**	**Device**	**Section**	**Function**
0000	0-7F	34020	5.3	The 34020 provides control and video timing registers.
0000	2C-2F	WRITEMASK	5.3.2	The writemask register supports display memory bit plane write protection.
400	3-F	BT463 (VCT-V) BT468 (VCU-V) BT459 (VCD-V)	5.6 5.7 5.8	The high resolution color map processes the primary overlay, and cursor pixel data into 8-bit Red, Green with Sync, and Blue analog video outputs. The BT459 and BT468 include on-chip 2-bit cursor.
400	3-F	BT482 (VCD-V)	5.9	The BT482 color map is used instead of the BT459 in cases where a digital output is supplied and medium resolution (640 x 480) analog is also needed. Includes 2-bit cursor.
400	20-2F	8242PC	5.14	The 8242PC is an Intel 8042 programmed with the Phoenix MultiKey keyboard BIOS.
400	43-4F 83-8F C3-CF	both BT431's BT431 #1 BT431 #2	5.11	Two BT431's are used on the VCD and VCT. It has both 64 x 64 bitmap cursor and full screen crosshair cursor functions.
	83-8F	BT482		The BT482 is used ***only as a cursor*** on some digital VCDs. It has a 32 x 32 bitmap cursor.

Table 5-7 VMEbus Side Device Buffer (continued)

Selected by LAR	Relative Address	Device	See Section	Function
400 400	103-13F 143-17F	DUART A DUART B	5.12	The 2681 Dual Asynchronous Receiver Transmitters (DUARTs) provide RS-232 ports for LK401-type keyboard, mouse or trackball, and console, etc.
400	183-19F	SCSI	5.13	Used on the VCT and VCU (optional). The 5380 Small Computer Systems Interface (SCSI) supports intelligent 8-bit parallel I/O devices.
400	1C2	ZOOM	5.5	**16-bit register access only.** The zoom control register is used to set the dot clock frequency, and in some configurations, front and back porch horizontal sync adjust, and sync polarity.
800-BFF	0-3FC	HSP	5.15	**32-bit access only.** High Speed Port (optional) allows the 34020 to directly copy data into memory via a 32-bit port connected to VMEbus P2 connector.

Note

Except where noted, LAR=400 devices are byte (D8) addresses. Convert to D16 address by subtracting 1. Convert to D32 address by subtracting 3. This is due to the fact that the 34020 is little endian and the VMEbus is big endian.

5.3 TMS 34020 Graphics Systems Processor

The Texas Instruments 34020 Graphics System Processor (GSP) is a general-purpose, 32-bit programmable processor with specialized graphics instructions and a 512 byte LRU instruction cache. It includes a full set of video timing control registers. The 34020 has a 32-bit processor data (LAD) bus which is connected directly to the 34082 FPU and to a set of two 74BCT16652 16-bit bus transceivers which act as buffers between the low drive capability LAD bus and the high load memory/device (MAD) bus.

A 34020's 32-bit host address bus and 34020 to VMEbus data multiplexers support a low latency interface between the VMEbus and the 34020, memory, and devices. The 34020 takes care of arbitration and data and address bus control for the host interface. Commands, status, display parameters, graphics drawing, refresh, and display update address data are all passed over these common busses.

The 34020 operates on memory in byte, word or long word segments. It also supports page-mode read and write memory accesses for maximum memory performance. For graphics memory, color register, block fill and writemask functions are supported. The writemask operation is write enable per bit enable function which allows direct writes instead of read-modify-writes).

The 34020 derives its timing from a clock which is **independent** of the video clock. In fact, the standard clock is 40 MHz, while the typical video clock is 110 MHz. The 34020 has internal synchronizers which take care of VRAM memory accesses (CPU clock synchronous) and VRAM shift, load, and blank functions (video clock synchronous). The 34020 has inputs for the VRAM shift and load clocks so that it can keep track of blanking even with horizontal zoom enabled. Section 4.2 covers the high speed clock and zoom generation.

An interesting consequence of the dual clock nature of the 34020 is that if you read a register driven by the pixel clock (e.g. VCOUNT), you will get erratic results. You have to read the comparison flag or use interrupts to get correct results. The reason for this is simple: the VCOUNT register can change state in the middle of a 34020 read cycle. Its operations are totally asynchronous to the 34020 CPU clock.

The purpose of this section is to supplement the well written and already complete information provided in the 34020 User's Guide, especially in the areas which relate to display timing and memory interface. The 34020

User's Guide is a **necessary** adjunct to a comprehensive understanding of this device and can be ordered from TI or Peritek (see Section 1.2). Peritek offers in-depth software support, which is covered in Chapter 3.

Note that the 34020 registers can be accessed as long words in D32 mode by the VMEbus (when the LAR=0), consistent with the register organization shown in the 34020 User's Guide. When accessing 34020 registers as 16-bit words, the order of the registers is **reversed** from what is shown in the 34020 manual.

34020 side address	**VME word address**
0	2
2	0
4	6
6	4
etc.	etc

5.3.1 34082 Floating Point Coprocessor

The 34082 Floating Point Unit (FPU) chip can be installed in the socket provided on the board. When properly utilized, it can significantly enhance performance. TI compiler/assembler switches must be set to use it. Since the chip select for the FPU always responds, you must run an FPU instruction. An uncompleted operation will reveal its absence. The Coprocessor ID# for the 34082 is **0**.

The 34082 conforms to the IEEE floating point standard P754 R10.0 for high level math functions. In addition, it offers transcendental functions (trigonometrics, hyperbolics, exponentials, logarithmics, etc.) and non-transcendental functions (absolute values, square roots, negate, etc.) performed to 32-bit, 64-bit, or 80 bit precision.

Typical instructions sequences also include copying data between the 34020 and the FPU, accessing memory, and executing internal instuction microcode. See Section 1.1 for a complete description of the functions of the FPU.

5.3.2 Writemask Register

The 34020 writemask register is a 32-bit read/write register located in the TI register buffer (LAR = 0). It is used to enable any or all of the bit planes in graphics memory (VRAM). This memory has a write-per-bit feature which allows bit planes to be selectively write enabled. The **complement** of the bits to be write-enabled is loaded into this register (i.e. bit set = write **disabled**). The writemask register has no effect on system memory.

The 34020 actually maintains a writemask register internally which is intended to be passed to VRAMs which support a "persistent writemask" function. It depends on a special function of the VRAMs which involves passing the contents of the 34020 writemask register into the VRAMs, where the data is stored until changed by the 34020. Since regular VRAMs only hold the writemask data for the current cycle, the standard 34020 writemask function won't work. Therefore, the graphics board has an auxiliary writemask register. It is buried inside the 74BCT16652 registered bus transceivers which pass data between the 34020 local data bus (LAD) and the 34020 common memory and device data bus (MAD bus). The 74BCT16652s hold the writemask data (essentially acting as the VRAM writemask holding register) and gate it onto the MAD bus at the beginning of every VRAM cycle.

Writing the 74BCT16652 writemask register is a bit of a trick: When a program writes into the 34020 writemask register, the 34020 executes a special cycle to update the VRAM writemask holding register. A PLD detects this cycle and stores this data in the 74BCT16652s.

5.3.3 VRAM Color Register and Block Fill Special Function

There are a few instructions (VBLT, VFILL, and VLCOL) which can only be used with Video RAMs (VRAMs) which support so-called VRAM special functions, which include block write and color register support. Peritek PX Windows software automatically detects and uses special functions if the board's VRAM can support them.

The Color Register is used in conjunction with the VRAM block fill mode which allows up to 4 adjacent locations in the VRAM to be written in one cycle. In this way, large areas with repetitive patterns can be written at a considerable higher rate that normal.

5.3.4 Memory Types and Sizes

The board has a maximum of 8 MB (VCD-V) or 16 MB (VCT-V and VCU-V) of display memory and 32 MB of 34020 system memory capacity contained on field replaceable Single Inline Memory Modules (SIMMs). In addtion, four 32-pin PLCC sockets are provided for 8-bit Flash EEPROM devices (see Sections 2.4.4 and 5.17 for more information). The memory possible combinations are manifold and are designed to provide the experienced user with memory tailored to the application. Please contact Peritek to discuss your requirements.

The 34020 shares access to the display and system memories with the VMEbus by means of the 34020's host interface. When accessed by the VMEbus or 34020, the display and 34020 (non-EPROM) memories are byte addressable.

The board memories can be accessed by the VMEbus in one of two ways: A 1 KB "window" in A16 or A24 space which gives access to memory selected by the LAR (see Section 5.2) or a 64 MB block in A32 space. When used, the hardware byte-swapper quadruples these memory sizes.

The 34020 addresses long (32-bit) words, so masking must be done to limit the operation to a single byte or less (if desired). The masking may be done by setting the proper bit field size in the 34020, or, if the operation is to be performed on the display memory, by using the writemask register (see Section 5.3.3).

5.3.5 Byte Ordering and the Hardware Byte Swapper

The infamous big endian/little endian dilemma must be dealt with here. Big endian means that the least significant byte (Byte 0) is assigned to the high order (DA24-DA31) data lines. Conversely, little endian means that the least significant byte (Byte 0) is assigned to the low order (DA0-DA7) data lines. The merits of one way or the other is truly religious, and therefore we won't talk about that. But we do have deal with the consequences.

The 34020 is intrinsically a little endian device. The VMEbus is big endian. Although the 34020 ***does have*** a big endian mode there is a performance penalty for using it. Therefore, the graphics board is designed to run little endian. A programmable byte swapper is included to alleviate some of the CPU overhead which is otherwise incurred.

This section contains three parts:

a) a discussion of the VMEbus and 34020 byte ordering issues
b) example code for software byte swapping
c) a discussion of the new hardware byte swapper

5.3.5a VMEbus and 34020 Byte Order Mapping

The VMEbus observes reversed byte polarity relative to the 34020 and the board internal bus. It considers its byte 0 (A0/A1 = 00) to be the high byte, data bits 24-31, while its byte 3 (A0/A1 = 11) is the low byte, data bits 0-7. Since the VMEbus can do byte and word operations on board memories, care must be taken that bytes are not inadvertently swapped.

Long word operands are stored by the VMEbus with the high order word (bytes 0 and 1) preceding the low order word (bytes 2 and 3) in memory, the reverse of the 34020. The TI cross assembler/compiler tools will supply the object code in big or little endian. The Peritek downloader and software are written for little endian. The VMEbus must use an **odd** address for byte access of 8-bit devices (color maps, cursors, DUART, DLUT).

Table 5-8 Byte/Word/Longword Mapping

	VME Address	**34020 Byte Address**	**Data Lines**
Little Endian Byte Addressing	0	3	24 - 31
	1	2	16 - 23
	2	1	8 - 15
	3	0	0 - 7
Big Endian Byte Addressing	0	0	24 - 31
	1	1	16 - 23
	2	2	8 - 15
	3	3	0 - 7
Little Endian Word Addressing	0	2	16 - 31
	2	0	0 - 15
Big Endian Word Addressing	0	0	16 - 31
	2	2	0 - 15
Big or Little Endian Long Addressing	0	0	0 - 31

5.3.5b Example Code for Software Byte Swapping

If you consider the graphics memory as a 2-D array of bytes, and index into it as a byte matrix, the byte order will be reversed relative to the 34020 byte sense. This can be dealt with easily by Exclusive-ORing the low two bits of the index pointer. In "C" this would be:

```
primem [y][x^3] = pv    ;where primem is the memory.
```

5.3.5c The Hardware Byte Swapper

Starting with the VCD-V FAB REV 2 and the VCT-V and VCU-V FAB REV 4 a hardware byte swapper has been included. Use of the swapper under PX Windows can give between 5 and 15% performance boost. The swap mode expands the address space and allow multiple mappings to the same physical memory. This permits swap modes to be changed on-the-fly without changing control bits. The only penalty is memory space requirements on the VMEbus, which jumps by 4 times.

A16/A24 Swap Modes

If A1624SWAPEN is set then the 1 KB DBR expands to 4 contiguous 1 KB DBRs and the lower 2 bits of the DBRADR are don't care. The 4 KB block must start on a 4 KB boundary. Each DBR points to the same memory on the board, but with a different swap mode:

offset	swap mode
0x000	none
0x400	byte
0x800	word
0xC00	both

A32 Swap Modes

If A32SWAPEN is set then the 64 MB space expands to 4 contiguous 64 MB spaces and the lower 2 bits of the XARADR are don't care. The 256 MB block must start on a 256 MB boundary. Each 64 MB space points to the same memory on the board, but with a different swap mode:

offset	swap mode
0x00000000	none
0x04000000	byte
0x08000000	word
0x0C000000	both

Swap Mode examples

ABCD refers to 4 bytes 0-3 big endian or 3-0 little endian. Swap mode examples may be verified by writing in each mode and reading back in none mode or by writing in none mode and reading back in each mode. The examples reflect D32 accesses, however the swapper works correctly for D8 and D16 accesses as well. UATs also work but are not recommended because they may not give the results that you expect. In particular, a string of bytes written to the board using UATs will end up non-contiguous.

mode	write	data on board
none	ABCD	ABCD
byte	ABCD	BADC
word	ABCD	CDAB
both	ABCD	DCBA

5.3.6 Virtual Memory, Page Faults, and Autoincrement Registers

When copying data into host memory from the color map auto-incrementing registers (color palettes) one must be careful about page faulting on a virtual memory machine. Before reading the color map, you should "touch" the variable(s) you are copying into to ensure that they are in CPU memory. If you don't do this, you may get a page fault which would force a retry of the instruction. Since the color map has already been read when the page fault occurs, you will end up reading the color palette too many times.

5.3.7 34020 Memory and Device Addresses

This section covers the 34020 address equivalents for the board on-board memory and devices. The 34020 address is a **bit** address, not a byte address. In the case of 34020 internal registers, this means that the smallest increment in the 34020 space is 10 (hex). The 34020 uses the concept of field size (**FE**) to control the amount of data which is transferred at one time (this applies to memory and registers). This parameter is set up with special 34020 instructions. The field size for accessing 34020 internal registers should be 16 (bits). If it is 32, the register following the one meant to be accessed will also get changed (this is actually desirable with the timing registers). Table 5-12 lists all of the board device registers in the 34020 address space.

To ease program development, the 34020 system memory appears in two places in the 34020 memory map - at the top and the bottom. This is done because while the VMEbus/LAR addressing can only access the 34020 memory at the top of 34020 address space, it can, in fact, be desirable to have 34020 programs execute starting at 34020 address 0. Therefore, the board address decoder permits system memory to appear in two places. As long as you write relocatable (position independent) code, you could even switch between the spaces dynamically.

If you have 1 MB system memory, this means that the both the top and the bottom of the 34020 address spaces have the same 1 MB of memory. However, for larger system memory configurations, this is not true. Thinking of a memory block starting from the bottom of the address space, the **top** of that memory block will map to the top of the address space.

Table 5-9 LAR/34020 Starting Address Table

			TMS34020 address	
		LAR	**High**	**Low**
34020 Registers		0	C000	0000
Device Registers		400	C080	0000
High Speed Port		0800	C100	0000
Digital Lookup Table (VCD only)		C00	C180	0000
EEPROM		2000	C400	0000
Primary Display Memory	1st 2 MB	4000	C800	0000
	2nd 2 MB	4800	C900	0000
	3rd 2 MB (VCT/VCU)	5000	CA00	0000
	4th 2 MB	5800	CB00	0000
Overlay Display Memory	1st 2 MB	6000	CC00	0000
	2nd 2 MB	6800	CD00	0000
	3rd 2 MB	7000	CE00	0000
	4th 2 MB	7800	CF00	0000

Table 5-10 LAR/34020 Starting Address Table (continued)

System Memory, referenced to the *top* ***of the 34020 address space***

			Last Valid TMS34020 address	
		LAR	**high**	**low**
System Memory	1 MB	FC00	FF80	0000
	4 MB	F000	FE00	0000
	8 MB	E000	FC00	0000
	16 MB	C000	F800	0000
	32 MB	8000	F000	0000

System Memory, referenced to the *bottom* ***of the 34020 address space***

System Memory	(1 MB)	n/a	007F	FFFF
	(4 MB)	n/a	01FF	FFFF
	(8 MB)	n/a	03FF	FFFF
	(16 MB)	n/a	07FF	FFFF
	(32 MB)	n/a	0FFF	FFFF

n/a - these addresses are not accessible to the VMEbus.

5.3.8 Sample Address Calculations

This section is intended to help you figure out what VMEbus addresses the board appears at. All addresses are calculated by adding a host CPU's (A16 or A32) VMEbus base address **+** an offset. In this example, **A16 VMEbus** is the base address which the host CPU has mapped to the VMEbus A16 space (see Section 6.2) and is of the form **A16 VMEbus = abcd 0000**. In many systems, A16 space is mapped to high memory, so let's assume that **A16 VMEbus** = FFFF0000. **A32 VMEbus** is the base address which the host CPU has mapped to the VMEbus A32 space (see Section 6.2) and will be **A32 VMEbus = xy00 0000**. Fot this example, let's assume **xy = 10**.

Calculating the CSR and DBR Base Addresses

The graphics board's CSR group A16 address has a jumper selectable base address which we assume here to be **C000** (set by jumpers in Section 2.4.1). Therefore, the board appears at the 32-bit address: **A16 VMEbus** + **C000**, so we would get **FFFFC000**.

The DBRADR is a register containing the high 6 bits of the DBR A16 address (see Section 5.2.4). The actual VMEbus offset value = **D offset = DBRADR*400h**. Let us assume D offset = 8000 (DBRADR = 20). The board appears at the 32-bit address: **A16 VMEbus + D offset**, so we would get **FFFF8000**.

Calculating the A32 Base Address

The XARADR is a register containing the high 6 bits of the A32 extended address (see Section 5.2.3). The actual VMEbus offset value = **X offset = XAR*400000h**. Let us assume X offset = 8000 0000 (XARADR = 20). The board appears at the 32-bit address: **A32 VMEbus + X offset**, so we would get **1000 0000 + 8000 0000 = 9000 0000**.

Table 5-11 Graphics Board Local Memory Map

A16 VMEbus				
abcd C000- abcd C00A				CSR, LAR, XARADR, DBRADR, VECADR

A32 VMEbus+ X offset	**A16 VMEbus+ D offset**	**LAR**	**34020 address**	**34020 zero-based addresses for system RAM**	
0	abcd 0000- abcd 007E	000	C000 0000- C000 03E0		34020 Internal Registers
10 0000	abcd 0003- abcd 03BF	400	C080 0000- C080 0DE0		Device Buffer color maps, BT431s, PC Keyboard controller, DUARTs, Zoom Register
20 0000	abcd 0000	0800	C100 0000		High Speed Port
30 0000	abcd 0000- abcd 03FF	0C00- 0CFF	C180 0000- C18F FFE0		Digital Lookup Table
80 0000	abcd 0000- abcd 03FF	2000- 2FFF	C400 0000- C5FF FFE0		Flash EEPROM
100 0000	abcd 0000- abcd 03FF	4000- 7FFF	C800 0000- CFFF FFE0		**VCT-V** 1K x 1K x 32 Display Plane Four Pages
100 0000	abcd 0000- abcd 03FF	4000- 4FFF 5FFF	C800 0000- C9FF FFE0 CBFF FFE0		1K x 1K x 8 Primary Display Plane Four Pages (**VCD-V**) Eight Pages (**VCU-V**)
180 0000	abcd 0000- abcd 03FF	6000- 6FFF 7FFF	CC00 0000- CDFF FFE0 CFFF FFE0		1K x 1K x 4 Overlay Display Plane Four Pages (**VCD-V**) Eight Pages (**VCU-V**)
3C0 0000	abcd 0000- abcd 03FF	F000- F3FF	FE00 0000- FE7F FFE0	0000 0000- 007F FFFF	34020 System Memory Bottom 1 MB (4 MB Config.)
3D0 0000	abcd 0000- abcd 03FF	F400- F7FF	FE80 0000- FEFF FFE0	0080 0000- 00FF FFFF	Low-mid 1 MB (4 MB Config.)
3E0 0000	abcd 0000- abcd 03FF	F800- FBFF	FF00 0000- FF7F FFE0	0100 0000- 017FF FFFF	High-mid 1 MB (4 MB Config.)
3F0 0000	abcd 0000- abcd 03FF	FC00- FFFF	FF80 0000- FFFF FFE0	0180 0000- 01FF FFFF	Top 1 MB (all versions)

Table 5-12   34020 and VMEbus Register Offsets(VCTLAR = 400)

	TMS34020 Address		**VMEbus Byte Offset from DBRBase**
	high	**low**	**Address**
Principal analog color map			
(BT459 - VCD-V, BT468 - VCU-V, BT463 - VCT-V)			
Address Register (low byte)	C080	0	3
Address Register (high byte)	C080	20	7
Data Buffer for control, cursor, overlay	C080	40	B
Data Buffer for primary color palette	C080	60	F
BT482 Alternate Color Map (low resolution VCD-V only)			
Write Address Register (primary cursor RAM)	C080	0	3
Data Buffer (primary)	C080	20	7
Primary Palette Mask Register	C080	40	B
Read Address Register (primary cursor RAM)	C080	60	F
Write Address Register (overlay cursor color)	C080	80	3
Data Buffer (overlay)	C080	A0	7
Command Register A	C080	C0	B
Read Address Register (overlay, cursor color)	C080	E0	F
8242PC Keyboard Controller			
Data Buffer	C080	100	23
Control/Status Register	C080	120	27
BT431 cursors A and B, (VCT-V and VCD-V Digital)			
Address Register Low	C080	y00	n3
Address Register High	C080	y20	n7
Bitmap Data Buffer	C080	y40	nB
Control Data Buffer	C080	y60	nF

n = 4 for cursors A,B; 8 for cursor A; C for cursor B
y = 2 for cursors A,B; 4 for cursor A; 6 for cursor B

Table 5-12   34020 and VMEbus Register Offsets(continued)

			VMEbus Byte
	TMS34020		**Offset from**
	Address		**DBRBase**
	high	**low**	**Address**
2681 (DUART) serial port controllers			
Mode Registers 1A and 2A	C080	w00	m+03
Status/Clock Register A	C080	w20	m+07
Command Register A	C080	w40	m+0B
Receive/Transmit A Buffers	C080	w60	m+0F
Input Port/Auxiliary Control	C080	w80	m+13
Interrupt Status/Mask Registers	C080	wA0	m+17
Counter/Timer Upper Registers	C080	wC0	m+1B
Counter/Timer Lower Registers	C080	wE0	m+1F
Mode Registers 1B and 2B	C080	x00	m+23
Status/Clock Register B	C080	x20	m+27
Command Register B	C080	x40	m+2B
Receive/Transmit B Buffers	C080	x60	m+2F
Reserved	C080	x80	m+33
Not used	C080	xA0	m+37
Stop Counter	C080	xC0	m+3B
Start Counter	C080	xE0	m+3F
For DUART A: m = 100, w = 8, x = 9			
For DUART B: m = 140, w = A, x = B			
5380 (SCSI, VCT-V and VCU-V option)			
SCSI Data In/Out	C080	C00	183
Initiator Command Register	C080	C20	187
Mode Register	C080	C40	18B
Target Command Register	C080	C60	18F
SCSI Bus Status/Select Enable	C080	C80	193
Bus and Status Register	C080	CA0	197
Latched Input Data Register	C080	CC0	19B
Reset Parity/Interupt Register	C080	CE0	19F
Zoom Register (use D16 access)	C080	E00	1C2

5.4 Initialization Tables

The 34020 must be programmed to generate the proper video timing for the hardware configuration and display format. This section includes a list of precalculated timing tables and their applicability. You can check pin 13 (horizontal) and pin 14 (vertical) on the video connector for correct timing intervals. If you think that the table you are using is incorrect or you don't know which table to use please call Peritek. The following table summarizes the notation for board features.

Table 5-13 Graphics Board Option Description

Option Type	Description
/X6	Analog only, 640 x 480, X Compatible,
/X10	Analog only, 1024 x 768, X Compatible
/X12	Analog only, 1280 x 1024, X Compatible
/A6, /A10, /A12	Analog only, as above, but CnP compatible
/D8	Digital only, VCD-V only, 640 x 480, 8-bit/pixel
X6/D8, X10/D8, X12/D8	Analog and Digital, as above, VCD-V only, X Compatible
A6/D8, A10/D8, A12/D8	Analog and Digital, as above, VCD-V only, CDP compatible
/K1	VCD-V only, BT431 cursors
/K2	VCD-V only, BT482 cursors

The table on the following lists some common initialization tables by board and oscillator type. Crystal means a fixed frequency pixel clock is used. PLL means that the ICS1562 programmable pixel clock is used. This listing is accurate as of the time of manual publication.

Table 5-14   Summary of Initialization Tables

Filename	Board Model	Configuration	Frequency	Type
D04VFP08.27	VCD/D8/K1	Flat Panel/VGA	27 Mhz	Crystal
D04VFP08.80	VCD/D8/K1	Flat Panel/VGA	80 Mhz	Crystal
D07VFP08.27	VCD/D8/K2	Flat Panel/VGA	27 Mhz	Crystal
D20V1708.80	VCD/X6	NTSC	80 Mhz	Crystal
D20V1708.98	VCD/X6	NTSC	98 Mhz	Crystal
D20VVGA8.27	VCD/X6	VGA	27 Mhz	Crystal
D20VVGA8.80	VCD/X6	VGA	80 Mhz	Crystal
D20VVGA8.98	VCD/X6	VGA	98 Mhz	Crystal
D24VFP08.27	VCD/X6/D8/K1	Flat Panel/VGA	27 Mhz	Crystal
D24VFP08.80	VCD/X6/D8/K1	Flat Panel/VGA	80 Mhz	Crystal
D90V1708.80	VCD/X10	NTSC	80 Mhz	Crystal
D90V1708.98	VCD/X10	NTSC	98 Mhz	Crystal
D90V1x18.80	VCD/X10	1024x1024	80 Mhz	Crystal
D90VVGA8.80	VCD/X10	VGA	80 Mhz	Crystal
D90VVGA8.98	VCD/X10	VGA	98 Mhz	Crystal
D90VX108.80	VCD/X10	1024x768	80 Mhz	Crystal
D90VX128.110	VCD/X10	1280x1024	110 Mhz	Crystal
TV1709.ics	VCT	NTSC	1562	PLL
TV1x19.ics	VCT	1024x1024	1562	PLL
TVVGA9.ics	VCT	VGA	1562	PLL
TVX109.ics	VCT	1024x768	1562	PLL
TVX129.ics	VCT	1280x1024	1562	PLL
TVsn39.ics	VCT	SunMode3	1562	PLL
UV1x18.ics	VCU	1024x1024	1562	PLL
UVVGA8.ics	VCU	VGA	1562	PLL
UVX108.ics	VCU	1024x768	1562	PLL
UVX128.ics	VCU	1280x1024	1562	PLL
UVX168.ics	VCU	1600x1280	1562	PLL
UVsn38.ics	VCU	SunMode3	1562	PLL

Display Format	Vertical Refresh	Horizontal Refresh	Pixel Clock
640 x 480	60 Hz	31.5 KHz	27 MHz
1024 x 768	70 Hz	60 KHz	55 MHz
1024 x 1024	57 Hz	60 KHz	80 MHz
1024 x 1024	60 Hz	64 KHz	100 MHz
1280 x 1024	67 Hz	64 KHz	110 Mhz
1280 x 1024	72 Hz	72 KHz	125 MHz
1600 x 1280	60 Hz	79 KHz	170 MHz

The following two pages contain an actual Peritek timing table as generated by Peritek's in-house timing table program, VIDP. This table applies to the VCT-V/X12. It is included only for illustrative purposes.

Table 5-15   Example Initialization Table (1280 x 1024, 110 MHz)

```
!Peritek-Initialization
!Generated with the following command line options:
! bt VCT canned U/T-X12-1562 isvme y duarts 2 scsi y overlay y oldStyle n halt
y overlay y scsi n
! 1024 lines * 1280 pixels, non-interlaced
!1562 derived frequency of: 109.921745 MHz
!actual frequencies: horizontal: 63.907992 KHz, vertical: 60.518931 Hz
! SCCS Version of generator: 1.45 4/8/94
TI34020:
0x36   0x110a   ! config     row/column addressing, refresh rate, VRAM modes
0x12   0x7      ! dpyctl     sync and display refresh
0x14   0x0000   ! controla   transparency, window checking, cache, PIXBLT
0x5e   0x03fc   ! dpymsk     Used for midline reload functions
0x1c   0x0000   ! hstctll    messages, emulator interface, busfault and retry
0x22   0x8000   ! hstctlh    halt, cache flush, reset, nonmaskable interrupt
0x42   0x0000   ! dpystl     Display starting address (low 16 bits)
0x40   0xc800   ! dpysth     Display starting address (high 16 bits)
0x20   0x0000   ! intenb     external, display, window violation interrupts
0x26   0x0000   ! intpend    external, display, window violation flags
0x24   0x0000   ! convsp     Source pitch conversion for XY to linear
0x2a   0x0000   ! convdp     Destination pitch conversion for XY to linear
0x2e   0x0000   ! pmaskl     Bit plane write disable (low 16 bits)
0x2c   0x0000   ! pmaskh     Bit plane write disable (high 16 bits)
0x32   0x0000   ! convmp     Mask pitch conversion for XY to linear
0x16   0x01ff   ! dpyint     display line count interrupt register
0x5a   0x0000   ! scount     shift clock count - used for midline reload
0x4a   0x0000   ! dincl      memory width (low 16 bits), Vertical Zoom
0x48   0x0001   ! dinch      memory width (high 16 bits)
! The following are the vertical timing parameters.
0x2    5        ! vesync     time at which vertical sync ends
0x6    28       ! veblnk     time at which vertical blanking ends
0xa    1052     ! vsblnk     time at which vertical blanking starts
0xe    055      ! vtotal     time at which vertical sync starts
! The following are the horizontal timing parameters.
0x0    23       ! hesync     time at which horizontal sync ends
0x4    48       ! heblnk     time at which horizontal blanking ends
0x8    208      ! hsblnk     time at which horizontal blanking starts
0xc    214      ! htotal     duration of the horizontal scan line in VCLKS
0x4c   190      ! heserr     Defines the point at which serration ends
0x28   0x20     ! psize      Set pixel size (1, 2, 4, 8, 16, or 32)

PCC1562:
0x10a021a 0xa6001201
```

Table 5-15   Example Initialization Table (continued)

```
! Next block is cursor control register block
BT431:
0x000           0x04       ! command register
BT463:
0x201           0x48       ! command register 0
0x202           0x48       ! command register 1
0x203           0xc0       ! command register 2
0x205           0xff       ! P00-07 readmask register
0x206           0xff       ! P08-15 readmask register
0x207           0xff       ! P16-23 readmask register
0x208           0xff       ! P24-27 readmask register
0x209           0x00       ! P00-07 blinkmask register
0x20a           0x00       ! P08-15 blinkmask register
0x20b           0x00       ! P16-23 blinkmask register
0x20c           0x00       ! P24-27 blinkmask register

TI34020:
0x12            0xd007     ! dpyctl - turn on display after initializing everything
! Information region
INFO:
"CMMType"       0x0002     ! Bt463
"CURSType"      0x0002     ! BT431
"Overlay"       1

"VideoWide"     2048       ! width of video memory
"VideoHigh"     1024       ! height of video memory

"DisplayWide"   1280       ! width of displayed part of video memory
"DisplayHigh"   1024       ! height of displayed part of video memory
"NumDuarts"     2
"DlutSize"      0          ! used on the VCD-V only
"DlutCurs"      0          ! used on the VCD-V only
"HasScsi"       0          !  used with VCT-V and VCU-V SCSI option
"MovieMeg"      0          ! used on the VCD-V only
"MovieHigh"     0          ! used on the VCD-V only
"MovieWide"     0          ! used on the VCD-V only
```

Locations 12, 18-1C, 36, 3C, 4C, 50-5A, 5E, 68-7E reserved

5.4.1 Application Note: Tweaking 34020 Initialization Parameters

Ordinarily, you should be able to use one of the initialization tables shown in the list on the previous pages. However, it may be that small adjustments are required. This section gives you some advice on how to do this. You can also supply Peritek with a filled-in copy of the monitor parameters sheet which follow this section. We can then provide you with a complete, correct calculated version.

Most monitors have adjustments for Horizontal Frequency, Horizontal Position, Horizontal Size, Vertical Frequency, Vertical Position and Vertical Size. It is recommended that the monitor adjustments be tried before changing values in the initialization table.

To change the horizontal frequency:

Indications that the horizontal frequency needs to be changed are an unviewable picture with diagonal lines. Some monitors display no picture when the horizontal frequency is out of its bandwidth. The same symptoms can be caused by no sync at all, so make sure that the cables are connected correctly and that the monitor is configured correctly. The graphics board default output is sync-on-green.

The horizontal frequency is controlled by HTOTAL. The number of diagonal lines is an indication of how close you are, fewer lines are closer, more lines are farther. Changing the horizontal frequency will also affect the vertical frequency.

Decrease the horizontal frequency by making HTOTAL larger. This will generally result in a wider picture. Increase the horizontal frequency by making HTOTAL smaller. HTOTAL must be larger than HSBLNK. HSERR must be smaller than HTOTAL. Once the correct value of HTOTAL is found, reduce or increase HSERR by one-half of what the amount you change HTOTAL.

To change the horizontal position:

To shift the image **left** subtract an equal amount from HEBLNK and HSBLNK. HEBLNK must be larger than HESYNC. To shift the image **right** add an equal amount to HEBLNK and HSBLNK. HSBLNK must be less than HTOTAL.

To change the number of pixels:

To display less pixels make the difference between HEBLNK and HSBLNK smaller. To display more pixels make the difference between HEBLNK and HSBLNK larger. HEBLNK must be larger than HESYNC. HSBLNK must be less than HTOTAL.

To change the width of the image:

There are 3 ways to change the width (horizontal size) of the image.

1) Display more pixels. The aspect ratio remains the same.

2) Change the oscillator frequency. You will need to contact Peritek for advice and assistance. All the timing parameters will need to be recalculated.

3) Change the horizontal frequency. Increasing the horizontal frequency will result in a wider image decreasing it will result in a narrower image. Changing the horizontal frequency will also affect the vertical frequency.

To change the vertical frequency:

Indications that the vertical frequency needs to be changed are a picture which rolls up or down. Sometimes the appearance is of multiple pictures, one on top of another, with multiple horizontal lines. An excessively slow vertical frequency will cause the image to flicker. Some monitors display no picture when the vertical frequency is out of it's bandwidth. The same symptoms can be caused by no sync at all, make sure that the cables are connected correctly and that the monitor is configured correctly. The default output is sync-on-green.

It is best to calculate VTOTAL based on the monitors specified vertical scan rate. If that is not possible you can try adjusting it by this method.

There are two ways to change the vertical frequency.

1) Change the horizontal scan rate. As vertical timings are in units of horizontal lines, the vertical rate will change proportionately.

2) Change VTOTAL. To decrease the vertical frequency make VTOTAL larger. To increase the vertical frequency make VTOTAL smaller. VTOTAL must be larger than VSBLNK.

To change the vertical position:

To shift the image up subtract an equal amount from VEBLNK and VSBLNK. VEBLNK must be larger than VESYNC. To shift the image down add an equal amount to VEBLNK and VSBLNK. VSBLNK must be less than VTOTAL.

To change the number of lines:

To display fewer lines make the difference between VEBLNK and VSBLNK smaller. To display more lines make the difference between VEBLNK and VSBLNK larger. VEBLNK must be larger than VESYNC. VSBLNK must be less than VTOTAL.

To change the height of the image:

There are 2 ways to change the width (vertical size) of the image.

1) Display more lines. The aspect ratio remains the same.

2) Change the vertical frequency. Increasing the vertical frequency will result in a shorter image, decreasing it will result in a taller image.

Declaration

Peritek remains dedicated to making your application work. We can assist in creating special initialization tables for specific monitors and other output devices. If you need help it would be very useful if you can gather the data requested in the following form before calling us.

Request for Timing Table

Submit to: Peritek Corporation
Attn: Ivor Bowden
5550 Redwood Road
Oakland, CA 94619 USA
TEL: (510) 531-6500
FAX: (510) 530-8563
email: ivor@peritek.com

Company Information

Company Name________________________________
Contact_______________________________________
Phone Number_____________________
Fax Number_________________________
email_______________________________

Monitor Information

Monitor Brand_____________________ Model Number___________________

Horizontal Timing Information

Note: Horizontal timings may be given in pixel units (if given) or time units.
Horizontal Pixels per Line Displayed_____________________
Pixel Time or Frequency (optional)_____________________
Horizontal Total Line Time or Frequency_____________________
Horizontal Front Porch_____________________
Horizontal Sync Width_____________________
Horizontal Back Porch_____________________

Vertical Timing Information

Note: Vertical timings may be given in line units or time units.
Vertical Lines Displayed___________ Interlaced? (Yes/No)________
Vertical Lines Total or Frequency (Field Rate)_________
Vertical Lines Total or Frequency (Frame Rate)________
(same as Field Rate unless interlaced)
Vertical Front Porch_____________________
Vertical Sync Width_____________________
Vertical Back Porch_____________________

Sync Information

Composite Sync on Green (Yes/No)________
If Not Composite Sync on Green:_________
Sync Format (Composite or Separate Horizontal and Vertical)________________
Sync Polarity (+ or -): Composite:_____ Horizontal:_____ Vertical:_____

Additional Notes

____
____
____
____

5.5 Vertical and Horizontal Zoom

Note that while CnP (Graphics Subroutine Package) supports zoom, PX Windows doesn't.

Using VGA Cables and Timing Modes

In order to use a standard VGA cable, you have to set the sync polarities correctly for the monitor to select the right line mode.

Table 5-16 VGA Timing Modes

	vertical line count		
Zoom Register Bit	**480 lines**	**400 lines**	**350 lines**
HSYNCPOL	set	set	clear
VSYNCPOL	set	clear	set
VCMODE	clear	clear	clear

Vertical Zoom

The primary and overlay graphics screens are zoomed together vertically through bits 0-4 of 34020 register DINCL. This is a binary zoom, factors 1, 2, 4, 8, 16, 32. **Due to a bug in the 34020, vertical zoom does not work properly in interlaced displays**.

Horizontal Zoom

The zoom control register is used not only for horizontal zoom but also to program the ICS1562 programmable phase locked loop (PLL) pixel clock which is used in many configurations. For compatibility reasons, the bit functions can vary considerably on the VCD-V (non-ICS1562) versions - see the following page. A configuration tag accompanies the VCD-V which will tell you if your board uses the 1562. If you are unsure, contact Peritek for assistance.

5.5.1 ICS1562 Type Horizontal Zoom Register

All VCT-V and VCU-V and some versions of the VCD-V use the ICS1562 programmable phase-locked loop (PLL) pixel clock oscillator which is programmable to any frequency. This allows the hardware horizontal zoom circuit to be eliminated. To effect hardware horizontal zoom just reinitialize the timing registers and pixel clock. **Note that the ZCR is programmed by the PCC1562 word in the initialization table. As the 1562 only has 56 bits, bits 56-63 of the PCC1562 word are used for bits 3-10 of the ZCR.** Contact Peritek for assistance.

5.5.1a Horizontal Zoom Register (VCT-V and VCU-V)

Table 5-17 Zoom Control Register for VCT-V and VCU-V

Bit	Mnemonic	Function
31-9	- spare -	not used, not defined, reads 0 or 1
8	ICUR	**VCT-V only**. Enable interlaced cursors. **clear** = non-interlaced **set** = interlaced
7	EMODE	Enable genlock. **Contact Peritek before using** **clear** = internal sync **set** = external sync (genlock)
6	VCMODE	Vertical/Composite Mode for J3 pin 14 **clear** = vertical sync **set** = composite sync
5	VCSYNCPOL	Vertical/Composite Sync Polarity for J3 pin 14 **clear** = output is active low **set** = output is active high
4	HSYNCPOL	Horizontal Sync Polarity for J3 pin 13 **clear** = sync is active low **set** = sync is active high
3	VCLK	pixels per VCLK. Horizontal timing registers are programmed in units of 1 VCLK. **VCU-V**: **clear** = 16 **set** = 8 **VCT-V**: **clear** = 4 **set** = 8
2	1562HOLD	Bits 0-2 are used to program the ICS1562.
1	1562DATA	Peritek's PX Windows and CnP Graphics
0	1562DCLK	Subroutine Package support the 1562.

5.5.2 VCD-V/A6 Type ICS1562 Version Horizontal Zoom Register

This is the first of two styles of zoom control register for VCD-Vs which use the ICS1562 (see 5.5.1 for more information).

Table 5-18 VCD-V/A6 type ICS1562 Zoom Control Register

Bit	**Mnemonic**	**Function**
31-14	spare	not used, not defined, reads 0 or 1
13	FP	Front porch adjust (see /MF for definition)
12	BP	Back porch adjust (see /MF for definition)
11	GL	Genlock mode. **Contact Peritek before using.** **clear** = internal sync **set** = external sync (genlock)
10	SM	Sync Mode **clear** = composite sync **set** = block sync (no serrations during VSYNC)
9	SU	Setup - 7.5 IRE step between blank and sync **clear** = no step **set** = step enabled (normal)
8	IN	Interlaced mode **clear** = non-interlaced **set** = interlaced
7	JP	Reads jumper
6	V/CMODE	Vertical/Composite Mode for J3 pin 14 **clear** = vertical sync **set** = composite sync
5	V/CSYNCPOL	Sync Polarity for signal on J3 pins 14 **clear** = output is active low **set** = output is active high
4	HSYNCPOL	Horizontal Sync Polarity for on J3 pin 13 **clear** = sync is active low **set** = sync is active high
3	HI	reads high
2	1562HOLD	Bits 0-2 are used to program the ICS1562.
1	1562DATA	Peritek's PX Windows and CnP Graphics
0	1562DCLK	Subroutine Package support the 1562.

5.5.3 VCD-V/A6/D8 Type ICS1562 Version Horizontal Zoom Register

This is the second of two styles of zoom control register for VCD-Vs which use the ICS1562 (see 5.5.1 for more information).

Table 5-19 VCD-V/A6/D8 Type ICS1562 Version Horizontal Zoom Register

Bit	Mnemonic	Function
31-8	spare	not used, not defined, reads 0 or 1
7	JP	Reads jumper
6	V/CMODE	Vertical/Composite Mode for J3 pin 14 **clear** = vertical sync **set** = composite sync
5	V/CSYNCPOL	Sync Polarity for J3 pin 14 **clear** = output is active low **set** = output is active high
4	HSYNCPOL	Horizontal Sync Polarity for J3 pin 13 **clear** = sync is active low **set** = sync is active high
3	DAMODE	Digital/Analog mode **clear** = digital display **set** = analog display
2	1562HOLD	Bits 0-2 are used to program the ICS1562.
1	1562DATA	Peritek's PX Windows and CnP Graphics
0	1562DCLK	Subroutine Package support the 1562.

5.5.4 Horizontal Zoom Control Register (Non-ICS1562 VCD-V's)

The Zoom Control Register for non-ICS1562 VCD-V's controls the zoom factor for both primary and overlay screens. It is located at VMEbus address offset 1C3 in the device buffer (LAR=$400) and controls the master dot clock frequency. Depending on configuration, additional bits may be used to control sync modes on the BT482 and digital outputs and to provide for pixel increment control of the horizontal front and back porches. In some cases, additional bits control the polarity of the Horizontal and Vertical/Composite Sync pins on J3 and JP77 video connectors. The tables on the following pages define the register functions.

Normally, the oscillator installed in the board is 80.64 MHz, although other frequencies, such as 49.34 (for NTSC) or 110 MHz (for 1280 x 1024) can be used. The zoom register controls a 2 or 4 bit prescaler which gives integer divide down from the oscillator. This register is used either to expand the standard image width (i.e. zoom) or to prescale the master oscillator for a required pixel clock. An example of the prescale function would be to divide an NTSC master oscillator (49.34 MHz) to 12.33 MHz, which is what the NTSC color pixel clock is.

Under some circumstances the analog and digital ports can be used simultaneously (Sharp LQ10DH11 and LQ10DH15 panels use standard VGA timing), but oftentimes the timing requirements of the typical flat panel are slower than the analog. This is usually most apparent in the retrace specifications, where flat panels have minimal retrace times.

The VCD-V supports hardware integer horizontal zoom factors as shown below. These zoom factors, combined with the programmable 34020 timing parameters, give the VCD-V the ability to display almost any format from 1280 x 1024 non-interlaced to 64 x 64 interlaced. They are also useful for panning the display around an expanded region of interest.

Because the various configurations of analog and digital outputs affect the zoom capabilities, a summary table is provided. Please note that this states the situation as of this writing, and additional configurations will be added as time goes on. Please contact Peritek if you find that the existing versions do not meet your need.

Table 5-20 VCD-V Configuration Summary (non-ICS1562)

VCD-V Model Option	Display Type	Lookup Table Type	Typical Master Oscillator(s)	Valid Zoom Factors
/A12, /X12	Analog only	BT459	110	integer, 1-16
/A10, /X10	Analog only	BT459	80.64, 100	integer, 1-16
/X6, /A6	Analog only	BT482	24.576, 27.00	1 (fixed)
/X12/D8	Analog port	BT459	100, 110	1 (fixed)
	Digital port	-	100, 110	4 (fixed)
/X6/D8	Analog port	BT482	24.576, 27.00	1 (fixed)
	Digital port	-	24.576, 27.00	1 (fixed)
/D4, /D8	Digital	-	24, 27, 32 (max)	1
/MF	Digital	BT482	67.2, 80.64	3,4,5,6,7,8,16 **(1 not valid)**

Table 5-21 VCD-V Analog Only Zoom Register (BT459)

Bit	Mnemonic	Function
31-8	- spare -	not used, not defined, reads 0 or 1
7	RJP12	Reads the state of jumper JP12.
6	VCMODE	Vertical/Composite Mode for J3 pin 14 **clear** = vertical sync **set** = composite sync
5	VCSYNCPOL	Vertical/Composite Sync Polarity for J3 pin 14 **clear** = sync is active low **set** = sync is active high
4	HSYNCPOL	Horizontal Sync Polarity for J3 pin 13 **clear** = sync is active low **set** = sync is active high
3-0	ZOOM0-3	Zoom control bits. See Table 5-22.

Table 5-22 VCD-V/X12/D8 Analog/Digital Zoom Register

Bit	Mnemonic	Function
31-8	spare	not used, not defined, reads 0 or 1
7	RJP12	Reads the state of jumper JP12. Reads 0 when jumper is installed. When used with EPROM-based VCDTE Terminal Emulator, JP12 **installed** causes VCD-V to be initialized for BT459/1280 x 1024 analog output. When JP12 is open, VCD-V is initialized for flat panel/640 x 480 output.
6-1	xxx	Reads 0
0	Mode	Display mode enable. Mode = 0 then Analog output is selected. Mode = 1 selects digital output.

Table 5-23 VGA style Zoom Control Register

Bit	Mnemonic	Function
7	VCMODE	Vertical/Composite Mode for J3 pin 14 **clear** = vertical sync **set** = composite sync
6	VCSYNCPOL	Vertical/Composite Sync Polarity for J3 pin 14 **clear** = sync is active low **set** = sync is active high
5	HSYNCPOL	Horizontal Sync Polarity for J3 pin 13 **clear** = sync is active low **set** = sync is active high
4-0	------>	See /MF Zoom Control Register definitions on next page.

Table 5-24 VCD-V /MF Zoom Control Register (BT482 Analog and Digital)

<table>
<tr><th>Bit</th><th>Mnemonic</th><th>Function</th></tr>
<tr><td>31-8</td><td>spare</td><td>not used, not defined, reads 0 or 1</td></tr>
<tr><td>5-7</td><td>CR0-CR2</td><td>Zoom control bits 0-2
<table>
<tr><th>CR2</th><th>CR1</th><th>CR0</th><th>Zoom Factor</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>16</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>8</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>7</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>6</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>5</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>4</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>3</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>4</td><td>BP</td><td>Back porch adjust</td></tr>
<tr><td>3</td><td>FP</td><td>Front porch adjust
<table>
<tr><th>BP</th><th>FP</th><th>Function</th></tr>
<tr><td>x</td><td>0</td><td>Horizontal Front Porch is not delayed</td></tr>
<tr><td>x</td><td>1</td><td>Horizontal Front Porch is delayed</td></tr>
<tr><td>0</td><td>x</td><td>Horizontal Back Porch is not delayed</td></tr>
<tr><td>1</td><td>x</td><td>Horizontal Back Porch is delayed</td></tr>
</table>
It is not necessary to provide more than 1 bit for FB and BP because VCLK is in increments of 2 pixel clocks. Additional changes can be made directly to the HSBLNK and HESYNC.</td></tr>
<tr><td>2</td><td>SM</td><td>Sync mode. Affects Sync on VGA connector.
<b>SM = 0</b> selects <b>Composite Sync</b>. Horizontal serrations are present during Vertical Sync.
<b>SM = 1</b> selects <b>Block Sync</b>. Horizontal serrations are absent during Vertical Sync.</td></tr>
<tr><td>1</td><td>INTER</td><td><b>Set</b> to provide field sync to BT482 during interlaced operation. Must be <b>clear</b> for non-interlaced operation. You also have to set or clear Bit 4 in the BT482 cursor register.</td></tr>
<tr><td>0</td><td>SETUP</td><td><b>Works only when INTER is clear</b>. <b>Set</b> to select 7.5 IRE level blanking step. <b>Clear</b> for no blanking step. BT482 Command Register B bit 5 controls step in interlaced mode.</td></tr>
</table>

5.6 BT463 - Color Map Controller for the VCT-V

Note

This section describes the features of the BT463. Some of these features are not directly used or supported in the Peritek PX Windows or CnP Graphics Subroutine Package.

PX Windows does not support overlays (but this will change with X11R6), window type tables, multiple color maps, or pseudo-color.

The **CnP Graphics Subroutine Package** supports overlays and zoom and has the **potential** to support the other functions by virtue of special subroutine calls which allow you to access any BT463 register.

On the VCT-V, the composite video output is generated by the Brooktree BT463. It provides individual color maps for the red, green, and blue planes in 24 bit mode, and supports a pseudo color translation of 8 bits into a full 24 bits, 8 bits each for red, green, and blue. It has a 512 entry primary lookup table plus 16 entries for overlay and 2 entries for cursor color. The BT463 converts the pixel data coming from the display memory into analog voltages which drive the display monitor. There are two ways to make this conversion: true color and pseudo color.

The BT463 latches four pixels of primary screen data plus four bits for overlay. The data is synchronized internally through another register, and then fed pixel-by-pixel through the chip. Two additional mode control bits (window type) are also latched at each pixel time which supports dynamic true color/pseudo color switching. The BT431 cursors use the top two window type inputs.

As described in Section 4.2, an ICS1562 programmable pixel clock generator supplies pipeline reset, differential ECL level dot clock and load clock for the BT463 and the VRAM shift clock.

True color, which is what the VCT-V was really designed for, gives a full range (8 bits each Red, Green, and Blue) of color selection for each pixel, in other words, 16.7 million colors. Every pixel on a 1280 x 1024 display can be a unique value. 24 bits of display memory, divided into 8-bit Red, Green, and Blue sections, are required for true color displays. In order to allow modification of color balance (e.g. gamma correction) each 8-bit section is coupled to its own 8-bit in/8-bit out lookup table (inside the

BT463). Pseudo color is supported by the BT463, but not in Peritek's software. See the BT463 data sheet for more information.

As mentioned above, the BT463 supports a 4-bit overlay. Pixel intersections between any of the planes results in a unique color, so that the pixels will still be visible. This is because the overlay has a higher priority than the primary input in selecting an output color value. In other words, as long as the data bits going into the overlay inputs are ***NON-ZERO***, **they** will select a color for a particular pixel position.

Setting control bit CR12 in command register 1 splits the 4-bit overlay plane into overlay/underlay mode. When CR12 is set and for any given pixel bit 3 of the overlay plane is clear, then the lower three bits of the overlay plane actually define an underlay color.

There are additional control inputs to the BT463 which, when used in conjunction with bits in the BT463 command register, support switching between true and pseudo color modes on pixel boundaries. These bits are called the window type bits. They permit the simultaneous existence of several windows on the screen, each with its own color map, either true or pseudo color. For a complete description of how these bits work, see the BT463 data sheet.

The BT463 has twenty internal registers plus the window type table and the color palette RAM. They are accessed via a four single byte register/data buffer group programmed through the device buffer (VCTLAR = 400). Each byte is located on a **long-word** boundary. The first two bytes make a 12 bit address register. Byte 3 is the data buffer for register and window type table (REGDBR) and byte 4 is the data buffer for the color map (PALET)

The BT463 is controlled through an 8 bit I/O port. It is addressed indirectly, with the use of an address register, followed by three data registers. In order to make loading of the BT463 as fast as possible, the address register autoincrements after each access to one of the data buffer bytes. In the case of access to control registers, the address register increments after each access. When accessing the window type table, cursor colors, or color palette RAM, two additional low order address bits are used. They are internal only and are cleared by a read or write to the address register. The low two bits count mod-3, and thus cycle through the R, G, and B parts of each of the 256 LUT entries. Only after these bits have cycled will the main address register increment. Access to the window-type table should be restricted to vertical and horizontal blanking times. Note that the new values to be loaded into the color map location are actually loaded when the blue value is loaded. The red and green values are held in a temporary register inside the BT463 until the blue value is entered.

BT463 registers include bit-mask (or read mask), allowing any combination of R, G, B and overlay bit(s) to be unconditionally masked off. Additonal registers include blink mask, allowing any of these same bit(s) to be blinked. The details of the internal control registers are documented in the BT463 data sheet.

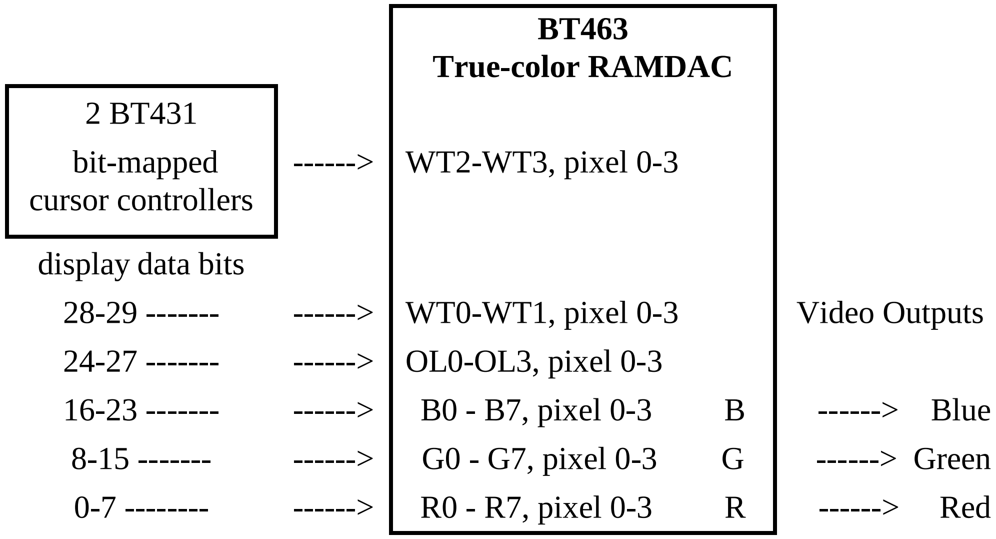
Table 5-25 BT463 registers

VMEbus Offset	Mnemonic	Function
3	ADRLO	Low 8 bits of the address register. Autoincrements after access to REGDBR or PALET. (See description on previous page).
7	ADRHI	High 4 bits of the address register. Autoincrements after access to REGDBR or PALET. (See description on previous page).
B	REGDBR	Data Buffer for control registers, window type table, and cursor colors - 2 x 3 (R,G,B). R,G,B locations cycled by mod-3 counter
F	PALET	Color map palette (R or W). Primary 256 x 3 (R,G,B) and overlay 15 x 3 (R,G,B). R,G,B locations cycled by mod-3 counter.

Note

Window Type Table (WTT) must not be updated during active display. Also, WTT must be programmed with multiple copies of cursor color.

Figure 5-2 BT463 Display Memory Bit Assignments



(BT463 latches 4 pixels at a time)

The following table and block diagram show what color value you get depending on the various inputs to the color map.

Table 5-26 BT463 Color Map Input Conversion

Window Type Inputs	**Cursor Input**	**Overlay Input**	**Primary Input R**	**G**	**B**	**Color Value**
0-3	0	n	a	b	c	Color map section controlled by window type RAM - see BT463 data sheet
0-3	0	0	0	0	0	R, G, B palettes
0-3	0	0	a	b	c	R, G, B palettes
.	.	.	.	.	.	
0-3	0	0	FF	FF	FF	R, G, B palettes
0-3	0	1	xx	xx	xx	overlay palette entry 1
.	.	.	.	.	.	
0-3	0	F	xx	xx	xx	overlay palette entry F
4-7	1	xx	xx	xx	xx	window type 4-7 (cursor color 0)
8-B	2	xx	xx	xx	xx	window type 8-B (cursor color 1)
C-F	3	xx	xx	xx	xx	window type C-F (cursor color 2)

5.7 BT468 - Color Map Controller for the VCU-V

Note

This section describes the features of the BT468. Some of these features are not directly used or supported in the Peritek **PX Windows** including overlays (but this will change with X11R6).

On the VCU-V, the composite video output is generated by the Brooktree BT468 high performance monolithic color map/cursor controller. The BT468 latches eight pixels of 8-bit primary screen data plus four bits for overlay. The data is synchronized internally through another register, and then fed pixel-by-pixel through the chip.

The BT468 provides composite sync generation, the color map lookup tables, cursor bit map and position control, primary and overlay plane blinking and blanking, and video output DACs. In addition to the customary 256 word translation map for the 8-bit primary graphics plane, it contains 15 additional table entries for the 4-bit graphics overlay planes and 3 for the 2-bit cursor. Each table entry is a 24 bit value (8 bits each for R, G, and B). It occupies four register locations in the VCU Device Buffer. The registers use only the low 8 bits of each word.

The BT468 is controlled through an 8 bit I/O port. It functions exactly the same way as the BT463, described in the previous section. The same goes for the ICS1562 programmable pixel clock generator.

The BT468 has a two bit graphics cursor. It contains a 64 x 64 x 2 bit map, position match registers, and counters triggered by the dot clock and referenced to horizontal and vertical sync. The match registers compare a programmed value (corresponding to an X-Y position on the screen) to the counters. When coincidence occurs, the cursor output become active and select one of the three cursor colors. By setting the correct internal control bits, the BT468 can supply a bit-map cursor and/or a cross-hair cursor. The display window for the cross-hair is programmable.

The details of the internal control registers are documented in the BT468 data sheet.

Table 5-27 BT468 registers

VMEbus Offset	Mnemonic	Function
3	CMAPARL	Low byte of BT468 address register. Two bit low order internal counter addresses individual R, G, and B locations when color maps are accessed. (See description above).
7	CMAPARH	High byte of BT468 address register. See description above.
B	CSRMAP	BT468 control/status buffer, cursor position control and bit map, zoom control, and cursor and overlay color maps.
F	PRIPAL	Primary plane color map palette. 256 x 3 (R,G,B) locations. R,G,B locations cycled by mod-3 counter.

The following table and block diagram show what color value you get depending on the various inputs to the color map (inputs are called Cursor Overlay, Graphics Overlay and Primary Screen).

Table 5-28 BT468 Color Map Input Conversion

Cursor Input	Overlay Input	Primary Input	Color Value
0	0	00	primary palette entry 0
.	.	.	.
0	0	FF	primary palette entry FF
0	1	xx	overlay palette entry 1
.	.	.	.
0	F	xx	overlay palette entry F
1	xx	xx	cursor palette entry 1
2	xx	xx	cursor palette entry 2
3	xx	xx	cursor palette entry 3

Figure 5-3 BT468 Display Memory Bit Assignments

```
display data              BT468 RAMDAC
    bits
  overlay
   56-59----    ----->  Overlay Pixel 7
   48-51----    ----->  Overlay Pixel 6
   40-43----    ----->  Overlay Pixel 5
   32-35----    ----->  Overlay Pixel 4
   24-27----    ----->  Overlay Pixel 3
   16-19----    ----->  Overlay Pixel 2
    8-11----    ----->  Overlay Pixel 1
     0-3 ----   ----->  Overlay Pixel 0
  primary       ----->
   56-63----    ----->  Primary Pixel 7
   48-55----    ----->  Primary Pixel 6
   40-47----    ----->  Primary Pixel 5
   32-39----   ------>  Primary Pixel 4               Video Outputs
   24-31----   ------>  Primary Pixel 3
   16-23----   ------>  Primary Pixel 2    B    ------>  Blue
    8-15----   ------>  Primary Pixel 1    G    ----->  Green
     0-7 ----  ------>  Primary Pixel 0    R    ----->    Red
```

(BT468 latches 8 pixels at a time)

5.8 BT459 - Color Map Controller for the VCD-V

Note

This section describes the features of the BT459. Some of these features are not directly used or supported in the Peritek **PX Windows** including overlays (but this will change with X11R6).

On the high resolution VCD-V (e.g. VCD-V/X12), the composite video output is generated by the Brooktree BT459 high performance monolithic color map/cursor controller. The BT459 latches four pixels of 8-bit primary screen data plus four bits for overlay. The data is synchronized internally through another register, and then fed pixel-by-pixel through the chip.

The BT459 provides composite sync generation, the color map lookup tables, cursor bit map and position control, primary and overlay plane blinking and blanking, and video output DACs. In addition to the customary 256 word translation map for the 8-bit primary graphics plane, it contains 15 additional table entries for the 4-bit graphics overlay planes and 3 for the 2-bit cursor. Each table entry is a 24 bit value (8 bits each for R, G, and B). It occupies four register locations in the VCD Device Buffer. The registers use only the low 8 bits of each word. The BT459 is controlled through an 8 bit I/O port. It functions exactly the same way as the BT463, described in the Section 5.6.

The BT459 has a two bit graphics cursor. It contains a 64 x 64 x 2 bit map, position match registers, and counters triggered by the dot clock and referenced to horizontal and vertical sync. The match registers compare a programmed value (corresponding to an X-Y position on the screen) to the counters. When coincidence occurs, the cursor output become active and select one of the three cursor colors. By setting the correct internal control bits, the BT459 can supply a bit-map cursor and/or a cross-hair cursor. The display window for the cross-hair is programmable.

A four bit horizontal zoom control is taken from the MACH110 VCDV6 and applied to the load inputs of a GAL16V8-7 (VCDV9) which functions as a high speed counter to yield an integer zoom function at 125 MHz. VCDV9 also generates a differential TTL clock and load for the BT459. Using the clock from the GAL, VCDV6 generates VCLK for the 34020 and VRAM shift clocks.

As described in Section 4.2, on some versions of the VCD-V, instead of the GAL16V8 circuit described in the previous paragraph, an ICS1562 programmable pixel clock generator supplies pipeline reset, differential ECL level dot clock and load clock for the BT459 and the VRAM shift clock.

The details of the internal control registers are documented in the BT459 data sheet.

Table 5-29 BT459 registers

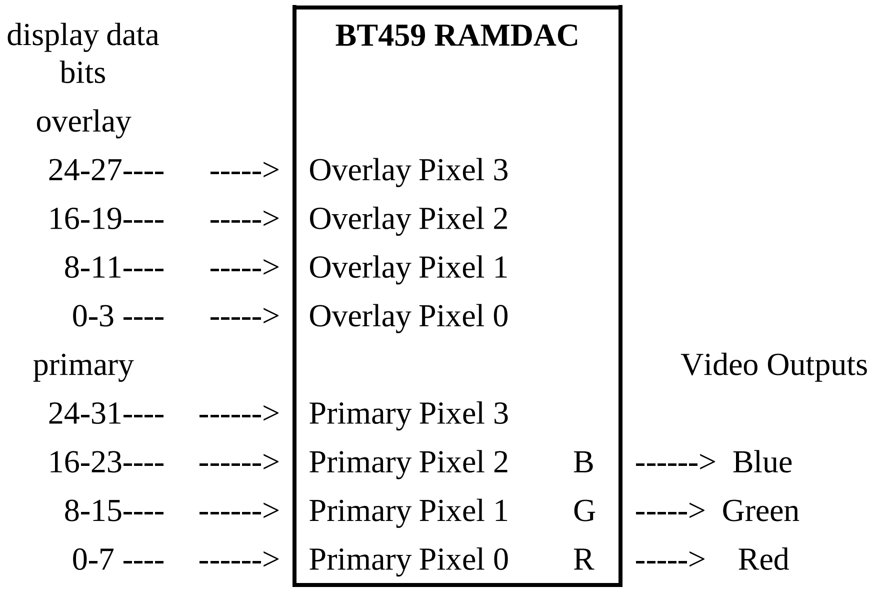
VMEbus **Offset**	**Mnemonic**	**Function**
3	CMAPARL	Low byte of BT459 address register. Two bit low order internal counter addresses individual R, G, and B locations when color maps are accessed. (See description above).
7	CMAPARH	High byte of BT459 address register. See description above.
B	CSRMAP	BT459 control/status buffer, cursor position control and bit map, zoom control, and cursor and overlay color maps.
F	PRIPAL	Primary plane color map palette. 256 x 3 (R,G,B) locations. R,G,B locations cycled by mod-3 counter.

The following table and block diagram show what color value you get depending on the various inputs to the color map (inputs are called Cursor Overlay, Graphics Overlay and Primary Screen).

Table 5-30 BT459 Color Map Input Conversion

Cursor Input	Overlay Input	Primary Input	Color Value
0	0	00	primary palette entry 0
.	.	.	.
0	0	FF	primary palette entry FF
0	1	xx	overlay palette entry 1
.	.	.	.
0	F	xx	overlay palette entry F
1	xx	xx	cursor palette entry 1
2	xx	xx	cursor palette entry 2
3	xx	xx	cursor palette entry 3

Figure 5-4 BT459 Display Memory Bit Assignments


display data bits
BT459 RAMDAC
overlay
24-27---- -----> Overlay Pixel 3
16-19---- -----> Overlay Pixel 2
8-11---- -----> Overlay Pixel 1
0-3 ---- -----> Overlay Pixel 0
primary
Video Outputs
24-31---- ------> Primary Pixel 3
16-23---- ------> Primary Pixel 2
B ------> Blue
8-15---- ------> Primary Pixel 1
G -----> Green
0-7 ---- ------> Primary Pixel 0
R -----> Red

(BT459 latches 4 pixels at a time)

5.9 BT482 - Color Map Controller for the VCD-V

Note

This section describes the features of the BT482. Some of these features are not directly used or supported in the Peritek **PX Windows** including overlays (but this will change with X11R6).

The BT482 color map/cursor controller is used in VCD-V configurations (e.g. VCD-V/X6/D8) which include the digital lookup table (DLUT) and a desire for medium resolution analog output. In most other cases the BT459 will be used (see Section 5.8). Since the BT482 only latches one pixel at a time and the VRAM array is arranged 4 pixels wide, it is fed by the DLUT pixel multiplexer. Once the pixel is clocked into the BT482, it is processed like the BT459, except there is no blink function.

The BT482 provides composite sync generation, the color map lookup tables, cursor bit map and position control, primary and overlay plane blinking and blanking, and video output DACs. In addition to the customary 256 word translation map for the 8-bit primary graphics plane, it contains 15 additional table entries for the 4-bit graphics overlay planes and 3 for the 2-bit cursor. Each table entry is a 24 bit value (8 bits each for R, G, and B).

The BT482 has a two bit graphics cursor. It contains a 32 x 32 x 2 bit map, position match registers, and counters triggered by the dot clock and referenced to horizontal and vertical sync. The match registers compare a programmed value (corresponding to an X-Y position on the screen) to the counters. When coincidence occurs, the cursor output become active and select one of the three cursor colors. By setting the correct internal control bits, the BT482 can supply a bit-map cursor and/or a cross-hair cursor. The display window for the cross-hair is programmable.

A four bit horizontal zoom control is taken from the MACH110 VCDV6 and applied to the load inputs of a GAL16V8-7 (VCDV9) which functions as a high speed counter to yield an integer zoom function at 125 MHz. VCDV9 also generates a differential TTL clock and load for the BT482. Using the clock from the GAL, VCDV6 generates VCLK for the 34020 and VRAM shift clocks.

As described in section 4.2, on some versions of the VCD-V, instead of the GAL16V8 circuit described in the previous paragraph, an ICS1562 programmable pixel clock generator supplies pipeline reset, TTL level dot clock and load clock for the BT482 and the VRAM shift clock.

The BT482 can be used as a three channel monochrome driver. In this situation, the programmer would load the R, G, and B lookup tables with identical data. Each of R, G, and B can independently drive a terminated monitor. Control bits in the BT482 Command Register A control which of the R, G, and B have sync present on the output. Bits in the zoom register control (see Section 5.5) to enable either composite sync (horizontal sync present during vertical sync) or block sync (no horizontal sync during vertical sync) to be driven on to them.

The BT482 occupies eight register locations in the VCD register block and use only the low 8 bits of each long word. The first register is a write address register, and is used to access the second register, which is the primary Look-Up Table (LUT) or the cursor bitmap RAM. The address register is actually a 10 bit counter, the high 8 bits of which appear in the register location. The low 2 bits are internal only and are cleared by a read or write to the address register. The register will autoincrement whenever a LUT location is addressed. The low two bits count mod-3, and thus cycle through the R, G, and B parts of each of the 256 LUT entries.

Since the address register is autoincrementing, it needs to be initialized only with the base address of the color map to read or write the entire map. The red and green values are held in a temporary register inside the BT482 until the blue value is entered, at which time all three are loaded.

The third register is a bit-mask, allowing any bit(s) going into the BT482 primary pixel input to be unconditionally masked off.

The fourth register is a read address register. When a value is loaded into it, the corresponding R, G, and B values are immediately read into an internal data buffer (for reading out through the LUT buffer) and the read address register is incremented. Thus, if you write a 5 into the read address register, it will actually read back 6. The cursor bitmap can also be read.

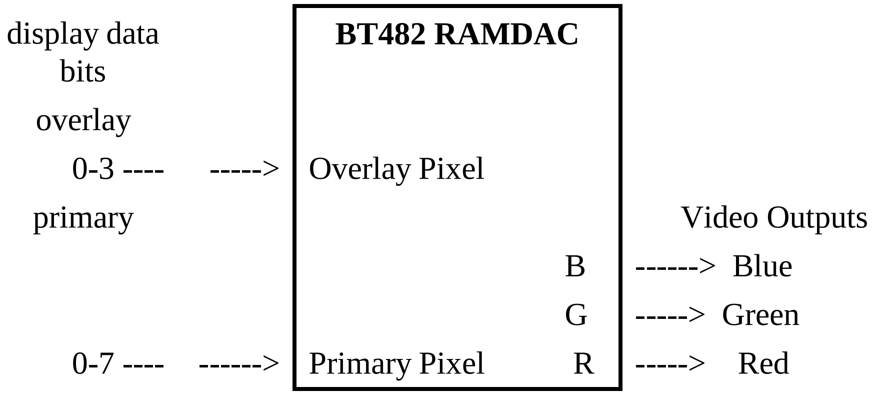
The high block of four BT482 registers applies to the overlay inputs and the cursor color pallette, and works identically to the low register set. There is no overlay bit-mask register, instead there is a command register.

The following table summarizes the BT482 registers and their functions. Addresses shown are for VMEbus. Subtract 3 for 34020 addresses. The details of the internal control registers are documented in the BT482 data sheet.

Table 5-31 BT482 registers

VMEbus Offset	Mnemonic	Function
3	PRWADR	Primary RGB triplet (0-255) write address register and cursor bitmap write address register.
7	PRIPAL	Primary color map palette. 256 x 3 (R,G,B) locations. R,G,B locations cycled by mod-3 counter. Cursor bitmap data buffer.
B	PRIMSK	Read mask register. Setting a bit in this register enables the corresponding primary plane bit to drive the color palette.
F	PRRADR	Primary RGB triplet (0-255) read address register and cursor bitmap read address register.
13	OLWADR	Overlay RGB triplet (0-255) write address register and cursor palette write address register.
17	OLYPAL	Overlay color map palette. 256 x 3 (R,G,B) locations. R,G,B locations cycled by mod-3 counter. Cursor palette data buffer.
1B	---	Command Register A
1F	OLRADR	Overlay RGB triplet (0-255) read address register and cursor palette read address register.

Figure 5-5 BT482 Display Memory Bit Assignments



The following table and block diagram show what color value you get depending on the various inputs to the color map (inputs are called Cursor Overlay, Graphics Overlay and Primary Screen).

Table 5-32 BT482 Color Map Input Conversion

Cursor Input	Overlay Input	Primary Input	Color Value
0	0	00	primary palette entry 0
.	.	.	.
0	0	FF	primary palette entry FF
0	1	xx	overlay palette entry 1
.	.	.	.
0	F	xx	overlay palette entry F
1	xx	xx	cursor palette entry 1
2	xx	xx	cursor palette entry 2
3	xx	xx	cursor palette entry 3

5.10 VCD-V Digital Lookup Table (DLUT)

The Digital Look Up Table (DLUT) maps the 8-bit primary, 4-bit overlay, and 2-bit cursor data into unified 8-bit pixels. The data is made available on a 26-pin header along with sync, clock, and blanking (see Section 2.5 for connection information).

The digital video output path for the VCD-V uses a MACH130 PLD to simulate the front end of a BT459. It latches and multiplexes 4 pixels (4 bits overlay and 8 bits primary) into a sequential stream, 1 pixel at a time. The 12-bit output of the MACH130 is coupled to a 32 KB SRAM, which has 15 address inputs and 8 data lines. The other 3 address lines are controlled by a 2-bit hardware cursor and blanking (1 bit). The output of the multiplexer is also connected to the BT482 (see Section 5.9).

Since it is necessary to program the SRAM, address and data from the host or 34020 must also be allowed to control the SRAM. UCASO and six 34020 CMA multiplexed address lines, normally intended for DRAMs, are also connected to the MACH130. When UCASO is high, the high order lines, (in this case LAR bits 0-5) are active and are latched into a register buried in the 130. When UCASO goes low, address lines 1-6 are selected and latched. Thus, a 12 bit address is formed inside the 130. Three additional address lines are latched inside the VCDV4 MACH130, and are multiplexed onto the cursor and blanking lines. When a read or write to the SRAM occurs, this 15-bit address drives the MACH130 output lines, not pixel data. An 8 bit bus transceiver connected to the SRAM data lines allows data to be read or written.

The data output from the lookup table constitutes a pixel, up to 8 bits wide. A programmable output mux controls the data format. It is synchronized with the pixel clock via a 74F273 and then goes to the digital video connector (26 pin dual row header) along with synchronized HSYNC, VSYNC, and CBLNK (composite blanking).

Lookup RAM Size	**Primary**	**Overlay**	**Cursors**	**Configuration**
32K RAM	8-bit primary	4-bit overlay	two 1-bit cursors	A
8K RAM	8-bit primary	2-bit overlay	two 1-bit cursors	S
8K RAM	8-bit primary	no overlay	two 1-bit cursors	C
8K RAM	8-bit primary	4-bit overlay	no cursors	T

An additional RAM address bit is allocated for blanking. The configuration determines how the DLUT is accessed. The base address is at LAR = C00 for the VME host. The base address is at C180 0000 for the 34020.

Remember that the VMEbus line buffer is 1 KB wide. The basic LUT block is 2 KB, of which the high 1 KB is reserved for blanking and thus loaded with 0. Only the low 1 KB of each LUT block is used by the primary, overlay and cursor resources. In an effort to make the mapping as clear as possible, please refer to the tables for each configuration provided below.

The DLUT is only 8 bits wide, and as with the device registers, appears in byte 3 in each VMEbus long word. It appears, of course, in byte 0, for the 34020.

It is implicit in these tables that some entries have to be loaded many times. For instance, in the **Case A** table, the Cursor 0 value has to be loaded (256*16) times. We have found that a very effective way to quickly load the table is to use the 34020 graphics FILL instruction. The CDP and PX Windows software available from Peritek both take advantage of this trick.

In the tables shown on the next page, the term **invalid** indicates that the entry is not defined. In some cases, an invalid location actually duplicates valid entries; in other cases, there is no duplication, and no effect when the location is accessed.

Case A Bytes 3-255 will be Primary or Overlay, bytes 259-511 will be cursor color 0, bytes 515-767 will be cursor color 1, and bytes 771-1023 will be the overlap of cursor 0 and cursor 1 (VME addressing).

Case S Bytes 3-255 will be Primary or Overlay, bytes 259-511 will be cursor color 0, bytes 515-767 will be cursor color 1, and bytes 771-1023 will be the overlap of cursor 0 and cursor 1 (VME addressing). Only overlays 4, 8, and 12 are valid.

Case C Bytes 3-255 will be Primary, bytes 259-511 will be cursor color 0, bytes 515-767 will be cursor color 1, and bytes 771-1023 will be the overlap of cursor 0 and cursor 1 (VME addressing).

Case T Only bytes 259-511 are valid (VME addressing).

Table 5-33 Case A DLUT Memory Map

LAR Even Values				**LAR Odd Values**	**2 KB Ram Block**
3-255	259-511	515-767	771-1023	3-1023	
Primary 0-255	Cursor 0	Cursor 1	Cursor 0+1	Blanking	0
Overlay 1	Cursor 0	Cursor 1	Cursor 0+1	Blanking	1
Overlay 2	Cursor 0	Cursor 1	Cursor 0+1	Blanking	2
Overlay 3	Cursor 0	Cursor 1	Cursor 0+1	Blanking	3
Overlay 4	Cursor 0	Cursor 1	Cursor 0+1	Blanking	4
Overlay 5	Cursor 0	Cursor 1	Cursor 0+1	Blanking	5
Overlay 6	Cursor 0	Cursor 1	Cursor 0+1	Blanking	6
Overlay 7	Cursor 0	Cursor 1	Cursor 0+1	Blanking	7
Overlay 8	Cursor 0	Cursor 1	Cursor 0+1	Blanking	8
Overlay 9	Cursor 0	Cursor 1	Cursor 0+1	Blanking	9
Overlay 10	Cursor 0	Cursor 1	Cursor 0+1	Blanking	10
Overlay 11	Cursor 0	Cursor 1	Cursor 0+1	Blanking	11
Overlay 12	Cursor 0	Cursor 1	Cursor 0+1	Blanking	12
Overlay 13	Cursor 0	Cursor 1	Cursor 0+1	Blanking	13
Overlay 14	Cursor 0	Cursor 1	Cursor 0+1	Blanking	14
Overlay 15	Cursor 0	Cursor 1	Cursor 0+1	Blanking	15

Table 5-34 Case S DLUT Memory Map

LAR Even Values				**LAR Odd Values**	**2 KB Ram Block**
3-255	259-511	515-767	771-1023	3-1023	
Primary 0-255	Cursor 0	Cursor 1	Cursor 0+1	Blanking	0
Overlay 1	Cursor 0	Cursor 1	Cursor 0+1	Blanking	4
Overlay 2	Cursor 0	Cursor 1	Cursor 0+1	Blanking	8
Overlay 3	Cursor 0	Cursor 1	Cursor 0+1	Blanking	12

Table 5-35 Case C DLUT Memory Map

LAR Even Values				LAR Odd Values	2 KB Ram Block
3-255	259-511	515-767	771-1023	3-1023	
Primary 0-255	Cursor 0	Cursor 1	Cursor 0+1	Blanking	0

Table 5-36 Case T DLUT Memory Map

LAR Even Values				LAR Odd Values	2 KB Ram Block
3-255	259-511	515-767	771-1023	3-1023	
invalid	Primary 0-255	invalid	invalid	Blanking	0
invalid	Overlay 1	invalid	invalid	Blanking	1
invalid	Overlay 2	invalid	invalid	Blanking	2
invalid	Overlay 3	invalid	invalid	Blanking	3
invalid	Overlay 4	invalid	invalid	Blanking	4
invalid	Overlay 5	invalid	invalid	Blanking	5
invalid	Overlay 6	invalid	invalid	Blanking	6
invalid	Overlay 7	invalid	invalid	Blanking	7
invalid	Overlay 8	invalid	invalid	Blanking	8
invalid	Overlay 9	invalid	invalid	Blanking	9
invalid	Overlay 10	invalid	invalid	Blanking	10
invalid	Overlay 11	invalid	invalid	Blanking	11
invalid	Overlay 12	invalid	invalid	Blanking	12
invalid	Overlay 13	invalid	invalid	Blanking	13
invalid	Overlay 14	invalid	invalid	Blanking	14
invalid	Overlay 15	invalid	invalid	Blanking	15

5.11 Hardware Cursors

A hardware implemented cursor is highly desirable because it allows the cursor to be moved on the screen simply by changing the cursor coordinates. The Peritek graphics boards have cursors implemented in one of three Cases:

Case 1 as a part of the color map chip (in the case of the BT459 and BT482 on the VCD-V and the BT468 on the VCU-V).

Case 2 as a single modified BT482 color map chip used as a cursor controller (in the case of some versions of the digital port on the VCD-V).

Case 3 as a pair of BT431 cursor control chips (in the case of the VCT-V and some versions of the digital port on the VCD-V).

The underlying principle in all cases is that the cursor is a bitmap whose contents and X-Y position can be programmed by the user. The bitmap array size is 64 x 64 for all but the BT482, which is 32 x 32. This is not a serious limitation since many cursors in X are only 16 x 16. The cursor is synchronized to the rest of the video timing with HSYNC and VSYNC. The cursor coordinates are a function of the **color map load clock** (which makes a transition every 4 or 8 pixel times) and horizontal and vertical syncs (which clear the cursor's X and Y position counters. It then counts load pulses to position itself along the horizontal axis and counts horizontal sync pulses to determine vertical position. Since the cursor clock is referenced to HSYNC and VSYNC, you can derive the relative 0,0 position from the 34020 timing initialization table (see Section 5.4). Since all video timing changes with zoom factor, the cursors will maintain their postions.

One complication of the hardware cursor is that interlaced displays confuse the cursor's vertical timing. The result is that, unless special measures are taken, it is "zoomed" vertically (by a factor of two) for interlace displays. The BT482 has an internal control bit which can be set for interlaced mode which then makes it work correctly. External logic does the same thing for the BT431 cursors when used on the VCT-V. Unfortunately, nothing can be done for the BT459 and BT468 cursors, and so they really are unsuitable for interlaced displays.

The **Case 1** cursor function is described in the BT459, BT468, and BT482 data sheets.

The **Case 2** cursor is a little bit unusual. Two of the three analog outputs from a BT482 color map/cursor controller are converted to digital. These outputs are then used as a 2-bit cursor for the DLUT (see Section 5.10). Needless to say the programming must be done carefully and the cursor is not well suited to a variety of video clock rates because of the analog signal output conversion process requires some "tuning" of the A-D circuit. The BT482 cursor is mapped to the cursor 0 address.

The **Case 3** cursor, which is used on the VCT-V, implements a two-bit hardware cursor using two BT431 cursor controllers chips (see Section 5.11 for a functional description). They can be accessed through the register area with decoding which allows addressing both chips together or as individual chips. The dual addressing allows both chips to be loaded with position data at the same time.

The cursor colors on the BT463 are not used, because to do so requires special gating to force the high four bits in the BT463 to be $E or $F (this method was used, however, on VCT-V revisons 0-2). Instead, the cursors use the high two bits of the 4-bit window type table inputs (see BT463 data sheet). This requires some unusual programming of the window type table inputs and means you effectively only get four unique tables.

Each BT431 has four internal registers, and occupies four register locations in the graphics board (VCD-V or VCT-V) register block. The registers use only the low 8 bits of each word. In order to provide the greatest flexibility, the BT431s can be individually addressed. Multiple addressing is provided so that if all you need to do is to change position registers, both chips can be changed at one time. See the chart below which shows how this works.

By setting the correct internal control bits, the BT431 can supply a bit-map cursor and/or a cross-hair cursor. The display window for the cross-hair is programmable. The BT431 cursor does not have an internal blink function.

The BT431 is accessed in much the same way as the BT459. For functional details, please refer to the BT431 data sheet.

Table 5-37 BT431 registers

VMEbus Offset	Cursors Selected	Description
43	BT431 cursors 0 & 1	Low byte address register (write only)
47	BT431 cursors 0 & 1	High byte address register (write only)
4B	BT431 cursors 0 & 1	Bitmap RAM data buffer (write only)
4F	BT431 cursors 0 & 1	Control register data buffer (write only)
83	BT431 cursor 0	Low byte address register (read/write)
87	BT431 cursor 0	High byte address register (read/write)
8B	BT431 cursor 0	Bitmap RAM data buffer (read/write)
8F	BT431 cursor 0	Control register data buffer (read/write)
C3	BT431 cursor 1	Low byte address register (read/write)
C7	BT431 cursor 1	High byte address register (read/write)
CB	BT431 cursor 1	Bitmap RAM data buffer (read/write)
CF	BT431 cursor 1	Control register data buffer (read/write)

5.12 Serial I/O Ports (DUART)

The DUART contains four independent asynchronous serial I/O ports. Each port can be programmed separately for transmit and receive baud rates, with a maximum baud rate of 38 Kb. The receive buffers are quadruply buffered, to minimize the possibility of data overrun. The serial interface provides data-leads only RS-232 as well as internal loopback (for testing). To obtain more understanding of the DUART and its many programmable functions, please refer to the 2681 data sheet available from the manufacturer (see Section 1.2 or contact Peritek)

The DUART uses a 3.6864 MHz oscillator for its master clock. Each DUART has internal divider chains provide a full range of software programmable baud rates and timer periods.

44-pin DUARTs are used on the graphics boards. Some control lines are reserved for HSP operations (see Section 5.15) and for programming the serial EEPROM (see Section 5.17). Unused inputs are connected to pullups and jumper pin blocks for use as user inputs. Three control outputs are used to drive red, yellow, and green LEDs.'

The RS-232 interface is provided by a MAX238 CMOS quad EIA RS-232 receiver/transmitter. This chip provides four complete channels as well as built in slew rate control. The chip also includes +/- 10 volt charge pump generators to supply the necessary RS-232 voltage swings and clamping diodes for protection against static charges on both inputs and outputs.

The DUART contains 16 register locations, several of which are either read or write only. The register number shown in the table below is added to the DUART base address (in the device buffer) to obtain the actual location. The values enable TX, RX, no parity, 8 bits/char, no loop, no RTS or CTS, 1 stop bit, 38.4 KB based on external clock (3.6864 MHz), and no interrupts.

The DUART has a single interrupt request line which is connected to the 34020 X2P interrupt pin. Internally, an interrupt can be caused by break, receiver full, and transmit ready for either channel . Externally, an interrupt can be caused by one of the two DUART input lines which function as change of state interrupts from the PC Keyboard controller. The interrupt mask register (IMR) filters out unwanted interrupts. A complementary interrupt status register holds the ANDs of the request(s) and the corresponding IMR bit. The following table lists the registers and some sample values.

While there are general purpose serial I/O routines in the CnP Subroutine Package, CnP has no knowledge of a mouse or keyboard. For the mouse, you will have to take the 5 byte packets and convert them into cursor motion on the screen. If you use an LK401 type keyboard with CnP, you will have to generate a scan-code to ASCII translation table because the LK401 is not ASCII (neither is a PC Keyboard).

CnP "test programs" exist in the /Board/progs directory which can be used as examples for these tasks. Contact Peritek for more information.

Table 5-38 DUART Initialization Table

VMEbus Relative Address	Typical Value(s)	Read/ Write	Mnemonic(s)	Function
m+3	13,7	yes	MR1A,MR2A	Mode Register A
m+7	----	Read	SRA	Status Register A
	CC	Write	CSRA	Clock Select Register A (38.4 Kbaud)
m+B	----	Read	----	Reserved-Read inhibited by hardware
	15	Write	CRA	Command Register A
m+F	Recv Data	Read	RHRA	Receive Holding Register A
	Xmit Data	Write	THRA	Transmit Holding Register A
m+13	----	Read	IPCR	Input Port Change Register
	70	Write	ACR	Auxiliary Control Register
m+17	----	Read	ISR	Interrupt Status Register
	0	Write	IMR	Interrupt Mask Register
m+1B	0	yes	CTUR	Counter/Timer Register High
m+1F	2	yes	CTLR	Counter/Timer Register Low
m+23	13,7	yes	MR1B,MR2B	Mode Register B
m+27	----	Read	SRB	Status Register B
	CC	Write	CSRB	Clock Select Register B (38.4 Kbaud)
m+2B	----	Read	----	Reserved-inhibited by hardware
	15	Write	CRB	Command Register B
m+2F	Recv Data	Read	RHRB	Receive Holding Register B
	Xmit Data	Write	THRB	Transmit Holding Register B
m+33	----	no	----	Reserved-inhibited by hardware
m+37	----	NU	----	Not used on board
m+3B	----	Read	----	Read to start counter
m+3F	----	Read	----	Read to stop counter

Note: m = 100 for DUART A, m = 140 for DUART B

5.13 SCSI Port

The Small Computer Systems Interface (SCSI) port is an optionally installed eight bit parallel interface designed to allow the VCT-V or VCU-V to communicate with other intelligent devices. The standard SCSI supports up to eight devices, which can be processors, disk controllers, etc. The 5380 chip, which is as the interface to the SCSI, contains registers and bus interface logic. It itself is not intelligent, and it requires a fair amount of handholding by the CPU.

The 5380 controller is designed to accommodate the standard 8-bit parallel SCSI as defined by ANSI X3T9.2. It operates in both initiator and target roles, and can therefore be used in host adapter, host port, and formatter environments. The 5380 supports arbitration, including reselection. It has special high-current open collector output drivers, capable of sinking 48 mA at .5 V, for direct connection to the SCSI bus. No other external logic is required to support the chip except for the on-board terminators.

Note

As mentioned in Chapter 2, the SCSI and the High Speed Port (HSP) share some signals on the P2 connector, thus precluding simultaneous installation of both options on a given board.

The 5380 contains several registers which facilitate control of the SCSI bus and as input buffers which allow the direct readback of all SCSI signals. The 5380 provides the actual electrical interface, but doesn't have any intelligence built in. The 34020 or the VMEbus host must do the actual bit manipulation and handshaking required to implement the SCSI protocol.

The NCR 5380 design manual, MAXTOR XT8000S OEM Manual, and the Adaptec ACB-4000A OEM manual are useful references. For the sake of clarity, relative to the ACB-4000A, the 5380 functions as the initiator, which means that the ACB-4000A is the target and controls the SCSI bus. Also, in the 5380 manual, the signals are referred to in the register tables as negative true signals, but in fact are programmed as normal positive true. Using the ACB-4000A as an example, we shall go through the SCSI protocol. While there certainly are other ways to use the SCSI port, this is a common application.

As mentioned above, the VCT/VCU SCSI port functions as the SCSI initiator, and the disk controller functions as the target. The 5380 has a register called the Target Command Register, which is used to select the SCSI control lines when the SCSI port is used in "target" mode. This register can also be used in our application: the expected phase (command or data, input or output, message or not) can be loaded into this register by the SCSI driver. If at any time the phase does not match, the 5380 will generate a **non-maskable** interrupt and clear the phase match bit in the Bus Status Register. This can be used to determine the end of a SCSI transaction, or to indicate an error (if the phase changes unexpectedly). One should be careful to never enter a protocol state only on the basis of the C/D, I/O and Message lines. REQ must be set prior to entering the next state.

Pseudo DMA Data Transfers can be used to significantly improve data transfer times. Ordinarily, the 34020 has to test the REQ* and ACK* bits to transfer data. By putting the 5380 into DMA mode when transferring data, some of the bit toggling will be assumed by the 5380 itself. Set bit 1 in the MDR, poll bit 6 (DRQ) in the Bus & Status Register and when active, read or write data as required to the Pseudo DMA Data Buffer (LAR 800, offset 0 or 34020 address C100 0000). DMA transfers are initiated by writing (anything) to the correct DMA start register.

While the SCSI chip is intended to control SCSI devices, it makes a fine parallel I/O port too. In this case, the CSD/ODR register would be used for passing 8-bits of data. Depending on whether you have the 5380 programmed as a target or initiator, other control bits can be used as handshaking bits. Consult the 5380 data sheet for detailed information.

Table 5-39 SCSI Register Initialization

Offset	**Read/Write**	**Value**	**Mnemonic**	**Function**
183	Read	--	CSD	Current SCSI Data
	Write	data out	ODR	Output Data Register
187	Yes	0	ICR	Initiator Command
18B	Yes	0	MDR	Mode Register
18F	Yes	0	TCR	Target Command Register
193	Read	--	CSB	Current SCSI Bus Status
	Write	1	SER	Select Enable Register
197	Read	--	BSR	Bus & Status Register
19B	Read	data in	IDR	Input Data Register
19F	Read	--	RIR	Reset Interrupt Register

Table 5-40 SCSI Protocol

The basic order of the SCSI protocol is arbitration (optional - required only for multiple initiators), selection, command transfer, data transfer (optional), and termination.

Selection is achieved as follows:

Monitor the SCSI signals BSY, RST, and SEL.
When they are clear, load SCSI ID#, SCSI data bus on, and SEL.
Test for BSY.
When BSY is detected, clear SEL and procede to command phase.

The Command transfers either 6 bytes or 10 bytes of command data:

Set the TCR for command phase and verify phase match.
Load data byte in output register.
Wait for ACK. When it is received, check REQ and phase.
If phase is the same, load more data, check ACK, etc.
Keep sending until phase changes. The ACB-4000A knows when to stop.

Now, the controller will want to transfer data or send termination status:

Check the CSBS register to find out what is needed. If data is to be transferred, call the 34020 routine to transfer data. Using pseudo DMA mode will speed data transfers. If a message is to be sent, read it in.

The Termination consists of a status byte and a message byte:

The Status byte indicates to the driver whether any problems have been encountered. The bits are decoded as follows:

Bit 0,4,5,6,7	reserved
Bit 1	ERROR - If set, execute Request Sense command
Bit 2	Equal - Search condition satisfied. (not used).
Bit 3	Controller Busy - re-run the current command.

The ACB-4000A always sends a zero for the message byte, but the data must accepted anyway, using the REQ/ACK interaction. Once the message phase has been terminated, the controller is ready for another command.

5.14 PC Keyboard Controller (8242PC)

A new feature of the graphics boards is the PC Keyboard controller chip. This is an Intel 8042 preprogrammed with the Phoenix Multikey keyboard BIOS. It supports both AT and XT keyboards and the PS/2 mouse. Peritek supports the AT keyboard function **only** in its PX Windows software. The PC Keyboard is not (officially) supported in the CnP Graphics Subroutine Package. However, by using the peek and poke commands resourcefully, you could program the 8242PC. It is not too hard to do. However, just as with the LK401, the PC Keyboard is a scan code device, not ASCII, so you will have to do some translation.

8242PC Keyboard interrupts are processed through DUART0 change of state inputs. This is convenient because the DUARTs drive the 34020 device interrupt pin directly. Care must be taken when accessing these bits, as they clear on read.

The keyboard connection is via a mini-DIN PS/2 style connector. Bidirectional clock and data lines communicate with an intelligent keyboard scan controller. Both chips must arbitrate and re-send if there is a collision on the clock/data lines.

5.15 High Speed Data Port (HSP)

The HSP is a 32-bit input port which allows the 34020 to transfer data from User Equipment (UE) into on-board memory. The HSP is available only by special order and its use must be qualified by the factory to ensure that the application is appropriate for the HSP hardware and software design. The Graphics Subroutine Package (CnP) includes subroutines to support the HSP functions. Refer to the CnP manual for detailed information. Section 2.5.3 has the connector pinout listing and suggests some appropriate bus receiver types.

The HSP uses the VMEbus P2 connector and follows the VSB data bus assignments. However, no attempt has been made to be compatible with the VSB. Also, the SCSI port option available on the VCT-V and VCU-V shares some of the same pins with the HSP. Therfore, you can't have both options on the same board. In the following description and in Section 2.5.3, signals names ending with L (e.g. VSL) are low active. Signals ending with H (e.g. DATA_00H) are high active.

A two port memory, located in the UE, will store the image. In the case of the VCD-V and VCU-V, DATA_00H - DATA_31H carry four bytes of pixel data, where pixel 0, the leftmost pixel on the display, is carried on DATA_00H - DATA_07H. DATA_00H is the LSB. In the VCT-V, DATA_00H - DATA_31H carry one pixel. Four additional control signals are defined (all low active): outputs: VSL, HSL, and REL; and input: PRDYL.

Under control of the 34020, a line of data is written into graphics board display memory at a position determined by software. Frames are transferred at a rate determined by software but can be as often as 30 times per second. A write-mask register protects any or all bits in the display memory from being written during the transfer process. Experiment has shown that data can be transferred at a sustainable rate of one 640 8-bit pixel line in about 67 us. Thus, there is ample margin to satisfy the data transfer requirements and still perform routine graphic functions. The control lines use four spare control lines lines in DUART0. All bits are read or write only. The lines are as follows:

Table 5-41 DUART Control Bit Usage for HSP

VMEbus Relative Address	Read/ Write	Mnemonic	Function
113	Read	IPCR	DUART0 Input Port Change Register
		Bit 0	PRDYL is intended to allow the UE to tell the graphics board when it has the first long word of pixel data available to be read. PRDYL must be driven high within 150 ns of HSL going low if data is not ready. PRDYL low indicates that data is ready and that REL may be activated. Software will poll PRDYL before initiating the transfer sequence, and will not activate REL until PRDYL goes low. Once the software starts reading the data, PRDYL is ignored until the next HSL cycle is started.
	Write	ACR	DUART0 Auxiliary Control Register
		Bit 0	VSL going low indicates preparation to read a new frame. VSL is a static bit controlled by software. No transitions will occur on VSL unless frames are being transferred to the graphics board. No set duty cycle is ascribed to VSL. Its period will vary depending on the size of the frames being transferred. Only the falling edge is important.
		Bit 1	HSL going low indicates preparation to transfer a line of data. The first data of a line should be presented to the HSP data bus in response. When frames of data are not being transferred, a programmable timer on the board will ensure a reliable stream of HSL pulses which the UE can use for refresh. When frames are being acquired, HSL will be activated before each line by the software. The period is a function of the time it takes to transfer a line of data. HSL may have a varied duty cycle and only the falling edge is important.
		Bit 2	REL is low indicates that data is being read from the buffer. New data may be presented as soon as REL goes low: no hold time is required. Data should be valid 20 ns before REL goes low. REL will be low for 62.5 ns (it shouldn't matter to the UE what this is). REL is inactive when frames are not being transferred. Only the falling edge of REL is important.

5.16 Graphics Board Interrupts

The board provides prioritized interrupts for use with the 34020 and the VMEbus host CPU. The VMEbus can be interrupted by the 34020. Interrupt sources for the 34020 are the DUARTs, VMEbus, and vertical line count. Vertical Sync is not supported because the 34020 has internal registers which support interrupt on **any** line.

Some early board revisions (VCD Rev 0, VCT/VCU Rev 0, 1) did support DUART interrupts to the VMEbus, but subsequent revisions do not.

Interrupt enables are handled at the device level. The VMEbus receives a single interrupt from the 34020. A VMEbus interrupt enable (in CSR - VINTEN) is used to disable interrupts to the VMEbus. See note about VINTEN in Table 5-2. See Section 5.2.5 for information about the VMEbus Interrupt Vector Address. The following table provides information regarding the interrupts.

For the 34020 side interrupts, all device interrupts are OR'd into X2P interrupt (see 34020 internal registers INTENB and INTPEND). This means that you have to poll the DUARTs to find out who interrupted. Figure 6-1 in the 34020 manual has a complete Vector Address Map.

Table 5-42 VMEbus Interrupt Functions

Interrupt Source	**VMEbus Standard Addresses (assumes VECADR = 40)**
34020	100

Table 5-43 34020 Interrupt Functions

Interrupt Source	**34020 autovector address**	**34020 INTPEND register name**
DUART	FFFF FFA0	X2P
34082 FPU	FFFF FFC0	X1P

5.17 Flash EEPROM and Serial EEPROM

Flash EEPROM

The board has four sockets for installing 32-pin PLCC EEPROMs. The devices can range in capacity form 16K x 8 to 512K x 8. The EEPROMs are wired as a full 32-bit wide memory to the 34020. This results in performance equal to running in normal system RAM. Well, almost: since page mode is not supported, the memory cycle time is always 200 ns. Page mode would give 100 ns, but can't be used with EEPROMs. The sockets can accommodate any of the sizes mentioned without jumpers. Smaller devices simply repeat multiple times.

Peritek has developed procedures for generating PROM-based software and loading EEPROMs with the code, using a PC and a BP Microsystems Programmer. Please contact Peritek for more information. As part of Peritek's software offerings, a simple terminal emulator (PTERM), PX Windows X11R6 server, and CnP Graphics Subroutine Package can now be supplied in EEPROM.

As an added bonus, a control bit, buried in the 34020 address decoder MACH130, allows the EPROM memory to respond to the highest 1 MB section of 34020 memory, blocking out the DRAM. This bit is initialized on SYSRESET (from the VMEbus). The graphics board can "autoboot" on power-up or reset when enabling jumpers are installed (see Section 2.4.5).

Although the program boots from highest memory, it is not desirable for it to continue running there. Therefore, using the magic of relocatable code, the program must "jump" to normal EEPROM address space. Once this happens, the control bit changes state so as to allow DRAM to takes its normal place. DRAM can be set up in advance of this since it also responds at the bottom of 34020 address space. Thus trap vectors and other code can be set up prior to making the jump.

Autoboot capability is highly desirable when a "single tube" system is required. In this case, the board can boot up to its terminal emulator and print out messages from the console port (when the graphics board console serial port is connected to the host CPU console port). Simple editing can also be done.

Serial EEPROM

The graphics board includes an IC position for an Atmel AT93C66 (or equivalent) 4 Kb (512 bytes) Serial Electrically Erasable Programmable Read Only Memory (EEPROM). This device is useful in an application where the EPROM is used to run an application program and some data needs to be stored during power-down. Taking into account that the EEPROM has about 100,000 read/write cycles, this data cannot be changed an infinite number of times.

No software is available at the time of writing which allows you to use this device (sigh).

The programming of the Serial EEPROM is done through the spare control lines OP0, OP1, and IP2 on DUART0 and also uses the chip select of DUART1. The pin functions are as follows:

DUART	**EEPROM**	
Signal Name	**Mnemonic**	**Description**
OP0	ECLK	Serial EEPROM clock
OP1	EDIN	Serial EEPROM data input
IP2	EDOUT	Serial EEPROM data output
DUACS1*	ECS	Serial EEPROM chip select

The EEPROM is programmed using a four wire protocol, although DUACS1* is actually only used during write and erase to trigger a busy/ready cycle. The protocol for programming the EEPROM is delineated in the data sheet. In general, the method for accessing the EEPROM is to use OP0 to transfer commands and data into or out of the EEPROM, with data from the EEPROM being read in on IP2 or written into the EEPROM on OP1. DUACS1* is toggled during a write or erase cycle so that the EEPROM will put status on OP1. This is necessary because the EEPROM takes about 1 ms to write or erase data.

Chapter 6
Troubleshooting

6.1 Introduction

This chapter contains information which should assist you in tracking down installation and functional problems with your board. There are several sections to this chapter:

6.2 Selecting an Address Range for your Board

6.3 Memory Map Address Example

6.4 Does this board talk at all?

6.5 General procedures

6.6 Maintenance, Warranty, and Service

6.2 Selecting an Address Range for your Board

Note

It is necessary to determine the correct address ranges of your CPU **before** you attempt to run the Peritek software. If you are unable to do so, even after reading this section, please contact Peritek for assistance.

Most CPU boards used on the VMEbus have a 32-bit physical address space, even if the CPU chip itself only puts 24 bits (i.e. 68000 or 68010). Normally a bus controller chip (such as the VIC068 or SCV64), located on the CPU board, converts the CPU chip's physical addresses to VMEbus addresses. Certain blocks of VMEbus address space are assigned to A16, A24, and A32 VMEbus address types. Unfortunately, since the VMEbus specification does not dictate a memory map for the address types, each manufacturer's board has its own assignments.

Clearly, knowledge of the details of the processor board memory map are important: you need to know the map in order to test out the board. Normally, the CPU board's boot PROM will set up the A16, A24, and A32 address space assignments in its local bus controller. If a controller chip isn't used, then the map will be hardwired into the CPU board design.

For A16 space, the high 16 bits of the CPU chip's address space is determined by the memory map (and thus the bus controller). What this means is that although the CSR block in the graphics board itself only sees the VMEbus address bits A1-A15, you actually use a 32-bit address in the CPU chip to address the board. The high 16 bits of the address is predetermined by the bus controller as corresponding to an A16 segment.

Given the broad range of computers which support the VMEbus, it is impossible to cover installation on all of them. However, we have provided in the following table a representative sampling. The table below summarizes the relevant A16, A24, and A32 spaces. All boards support A24 spaces too, and the graphics board can use it. This would only be necessary if your CPU doesn't support A16/D32 accesses. D32 accesses give the best data transfer performance. If you encounter difficulty, please do not hesitate to contact the factory for assistance.

Table 6-1 *Common CPU board addresses*

Manufacturer	Address Ranges	Resource	Comments
Force CPU-30 and CPU-40	00400000-F9FFFFFF	VSB/VME	A32, D32
	FBFF0000-FBFF7FFF	VME Short I/O	A16, D32
Force SPARC CPU's	00400000-F9FFFFFF	VSB/VME	A32, D32
	FBFF0000-FBFF7FFF	VME Short I/O	A16, D32
GMS V36 and V46	80000000-EFFFFFFF	VSB/VME	A32, D32
	F1000000-F100FFFF	VME Short I/O	A16, D32
Motorola MVME147S	DRAMsize-01000000	VME Standard	A24, D32
	01000000-EFFFFFFF	VME Extended	A32, D32
	FFFF0000-FFFFFFFF	VME Short I/O	A16, D16
Motorola MVME162/167/187/188/197	F0000000-FEFFFFFF	VME	A24/A32, D32
	FFFF0000-FFFFFFFF	VME Short I/O	A16, D32
Themis SPARC CPU's	04000000-FCFFFFFF	VME Extended	A32, D32
	FFFF0000-FFFFFFFF	VME Short I/O	A16, D32

Memory Map Example

In the following paragraphs the addresses given are based on the Motorola 167. This high performance Single Board Computer (SBC) has a 68040, serial and parallel I/O, memory, SCSI, and Ethernet. Its specifications are similar to SBCs made by Themis, Force, and Heurikon.

The Motorola 142/167/187/188/197 memory maps are programmed in the debugger (using the ENV command) for A24, A32, D16, and D32 functions. Contact Peritek for more information. The CPU's have an undocumented register which responds at offset 10 in A16 space, so Peritek boards cannot use the bottom of A16 space. Also, the 167/187 may be configured by operating system software to support only A16/D16 transfers. The Peritek device driver writes the correct value in VMEchip2.

The graphics board CSR block and 1 KB Line Buffer both reside in A16 space. The CSR block base address is controlled by 4 jumpers (see Section II.4.1), which give 16 possible address offsets within the A16 space. Normally, the offset is C000, giving a MVME CPU address of FFFFC000 for the CSR block. Now, the Line Buffer has an address offset determined by a programmable register in the CSR block. These bits correspond to address lines 10-15 for A16 and lines 10-23 for A24. On the MVME CPU's, Peritek uses is 8000, giving a CPU address of FFFF8000 for A16.

The graphics board also can respond to a 64 MB window in A32 space, giving access to the entire on-board memory. In this case, a programmable register in the CSR block allows you program the address of that window. The 6 bits in this register correspond to the high 6 bits of the A32 address. In the MVME CPU, A32 space extends up to MVME CPU address F0000000. Normally, we place the board at A0000000.

It is important to remember that the addresses which the graphics board responds to are a function of the VMEbus address and the address modifier codes. The address modifier codes must be asserted such that the board's CSR group and Line Buffer spaces are in A16 space and the extended memory block appears in A32 space. Not surprisingly, the address modifier codes are set up by the CPU board's bus controller.

The graphics board responds to D8, D16 and D32 accesses in A16 space and D8, D16, and D32 accesses in A32 space. However, bytes 0 and 1 (bits 16-31) are undefined for D32 accesses to the CSR block. Byte (D8) writes to the CSR block have no effect; only D16 and D32 writes will change these registers. Byte reads, however, do return the correct data.

6.3 Does this board talk at all?

Note

Most 680xx-based CPU boards use some variation of the Motorola debugger for base level communications and diagnostics. The installation/debug section uses the Motorola version of the debugger in the examples.

It is a great help in determining the cause of a board problem if you have either the PTERM or autobooting CnP PROM, because if you get a display on power up it is a reassurance that the board probably works. If you don't have a PROM, then you are running blind, especially if this is a new installation.

Typical equipment required for the test is a suitable monitor (analog RGB with sync on green) and a MVME167-based computer with Unix System V. Peritek supports QIC-150 cartridge tape distribution media.

If you have problems with the board responding, you may have an address conflict. This section tells you how to check to be sure there are no other devices which respond to its addresses. The board in this example follows MVME167 VMEbus address assignments with respect to A16, A24, and A32 areas. If you are not using a 167, see Chapter ? for other CPU board addresses. The graphics board responds to A16 and A32 bus masters (but not A24 unless specially enabled). The standard addresses for the graphics board are:

MVME167	Physical Address	Address Type	Data Type
Control Registers	FFFFC000-FFFFC00F	A16	D16, D32
Line Buffer	FFFF8000-FFFF83FF	A16/A24*	D8, D16, D32
Full Memory	A0000000-A3FFFFFF	A32	D8, D16, D32

* optional - set by control bit in CSR

The MVME167 debugger can be used to determine whether there are address conflicts. Except for the graphics board CSR base address itself, all address areas are software programmable. If you want to use the console debugger to examine the physical address areas that the board will occupy, do not allow the computer to boot. Using the following procedure:

Note

When starting this procedure, make sure that the graphics board ***is not*** plugged in.

```
enter   MM FFFFC000;L        <CR>   ;examines one CSR location

enter   MM FFFF8000;L        <CR>   ;examines the line buffer base address
```

You will get an "**Exception: Access Fault (Local Off Board)**" if there is no device already installed which uses the board's addresses. This is what should happen for both of these attempted memory examine operations.

Note

Now install the board following the procedure in Section 2.3.

Power up and don't let the debugger boot the operating system. Use the debugger to verify that board registers can now be read (see below). The Line Buffer cannot be read until the MEMON enable bit in the graphics board CSR is set. Since we can't use the bottom of A16 space in a 167 (or 187), it is also necessary to load the Line Buffer Address Register (LBAR, at FFFFC008). This is because the register comes up zero.

```
enter    MM FFFFC000;W       <CR>
receive  FFFFC000 4000?
enter    60                  <CR>;sets MEMON and CRTCON
receive  **WARNING:NO MATCH**    ;get an error because the 4000 bit is  lways set
enter    .                   <CR>;end the dialog

enter    MM FFFFC008;W       <CR>
receive  FFFFC008 0000?
enter    20                  <CR>;set line buffer address to FFFF8000
receive  FFFFC00A 0020?
enter    .                   <CR>;end the dialog

enter    MD FFFF8000:400;B   <CR>;dumps 1024 bytes starting at FFFF8000
receive  a bunch of bytes

enter    MD FFFF8022:1;W     <CR>;dumps the 34020 HSTCTLH register
receive  FFFFC008 8010           ;this is the right value for a halted 34020
enter    .                   <CR>;end the dialog
```

You will get an error message (which is not what you want) if there is no response. If indeed the board appears to be dead, call Peritek for further assistance.

6.4 General Procedures

The VCT-V, VCU-V, and VCD-V were designed with reliability and durability in mind. Nevertheless, it may happen that a problem will occur. This section is devoted to aiding the user in tracking down the problem efficiently and quickly.

You may be able to locate minor problems without technical assistance. Before placing a service call, try to solve the problem by following the directions given below, in Table 6-2. If the problem can not be remedied, Peritek can then issue a Return Material Authorization (RMA) so that the board can be returned to the factory for quick repair.

It can happen that installing a new board will overload the computer's power supply if the power supply margins are exceeded. The first step in ascertaining if this is the problem is to calculate a power supply budget. This involves adding up the power requirements of each board in the system to see if you are within specification. Consult your computer's technical manual for information on how to correctly determine this. A typical VCT-V, VCU-V, or VCD-V will draw about 2 amps at +5 volts.

When attempting to verify that the power supply is working properly, it is not unusual to unplug everything and measure the supply without a load. While this practice is acceptable for linear supplies, switching supplies (which are very commonly used in computers) require a certain load before proper regulation is achieved. Typically, at least 5 Amps must be drawn from the +5 volt supply before the +12 volt supplies will give the proper readings.

It can also happen that if you build your own cables and you short +12 or +5 to ground on the connector you may trigger the auto-resetting fuse which protect power supply pins when an overload occurs. These fuses are actually PTC elements which reset automatically when an overload is removed.

You may also wish to refer to the following sections:

2.2, 2.3	Installation and Checkout
2.4	Jumper Changes
3	Software Summary
5.4	Initialization Tables
6.6	Maintenance, Warranty, and Service

Table 6-2 *Common CPU board addresses*

Fault	**Possible Cause**	**Corrective Action**
Control Panel dead - On/Off switch unlit	No AC power	Check power cord. It may have been dislodged when installing board.
On/Off Switch lit	No DC power	Check for correct +5 and +12 volts.
Cannot Boot	Cable(s) dislodged	During installation an unrelated cable can get dislodged.
Cannot read Peritek distribution media	Improperly inserted, damaged, or incorrect media.	Check insertion and position. Take care that media is "mounted" properly. Unix distribution uses TAR format.
No message on console terminal or messages are garbled	Terminal disconnected or not configured properly.	Make sure cable between terminal and computer is plugged into proper terminal port. Put terminal into Local mode and verify operation.
System crashes or you get a "Trap" message	Software not installed correctly	Check installation procedures. See Software Release Notes.
No image on Monitor	COAX cables not connected properly or monitor is not on.	Check BNC cables, replace if necessary. Be sure to initialize board with correct initialization table.
Image is smeared or doing flip-flops	Sync signals missing or monitor sync failure.	Make sure monitor accepts sync on green, that monitor is terminated, and the hold controls are adjusted properly. Make sure that R,G,and B cables do go to R,G and B inputs. Check initialization parameters.
PX Windows Server is very slow to start up. Mouse movement is fast but windows are slow to open.	Graphics board to Host CPU interrupts are not being serviced.	Check interrupt pass/grant jumpers. Check operating system for correct interrupt configuration.
No response to mouse motion and/or keyboard entry.	Keyboard or mouse cable not plugged in. PX Windows board side server is crashed.	Check cabling. Reload software.

6.6 Maintenance, Warranty, and Service

Maintenance

The VCD-V, VCT-V, and VCU-V require no regular service, but if used in a particularly dirty environment, periodic cleaning with dry compressed air is recommended.

Because of the heat generated by normal operation of the graphics board and other boards in the system, forced crossflow ventilation is required. If forced ventilation is not used IC temperatures can rise to 60 degrees C or higher. Such high temperature operation causes IC failures and reduced MTBF. With proper forced air cooling IC temperatures will be less than 35 degrees C.

Warranty

The VCD-V, VCT-V, and VCU-V graphics boards are warranted to be free from defects in material or manufacture for a period of 6 months from date of shipment from the factory. Peritek's obligation under this warranty is limited to replacing or repairing (at its option) any board which is returned to the factory within this warranty period and is found by Peritek to be defective in proper usage. This warranty does not apply to modules which have been subjected to mechanical abuse, electrical abuse, overheating, or other improper usage. This warranty is made in lieu of all other warranties expressed or implied. **All warranty repair work will be done at the Peritek factory.**

Return Policy

Before returning a module the customer must first request a Return Material Authorization (RMA) number from the factory. The RMA number must be enclosed with the module when it is packed for shipment. A written description of the trouble should also be included.

Customer should prepay shipping charges to the factory. Peritek will prepay return shipping charges to the customer. Repair work is normally done within ten working days from receipt of module.

Out of Warranty Service

Factory service is available for modules which are out of warranty or which have sustained damage making them ineligible for warranty repair. A flat fee will be charged for normal repairs and must be covered by a valid purchase order. If extensive repairs are required, Peritek will request authorization for an estimated time and materials charge. If replacement is required, additional authorization will be requested.

All repair work will be done at the Peritek factory in Oakland, California, unless otherwise designated by Peritek.

Index