

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

CHAPTER 4
THEORY OF OPERATION

The PCVISION*plus* Frame Grabber is plug–compatible with the IBM Personal Computer (PC). It is the heart of a digital image processing system. As described in Chapter 1, the PCVISION*plus* Frame Grabber digitizes an analog video signal, stores the digitized data, and converts the stored image back to an analog format for display on an external video monitor.

This chapter describes the basic functional areas of of the PCVISION*plus* Frame Grabber (as shown in Figure 4–1). These basic blocks are:

- Video input interface
- Frame memory
- Display interface
- Host interface

In addition, this chapter contains information on using look–up tables for point processing, true color image processing, and operation of the board at the CCIR 50 Hz standard.

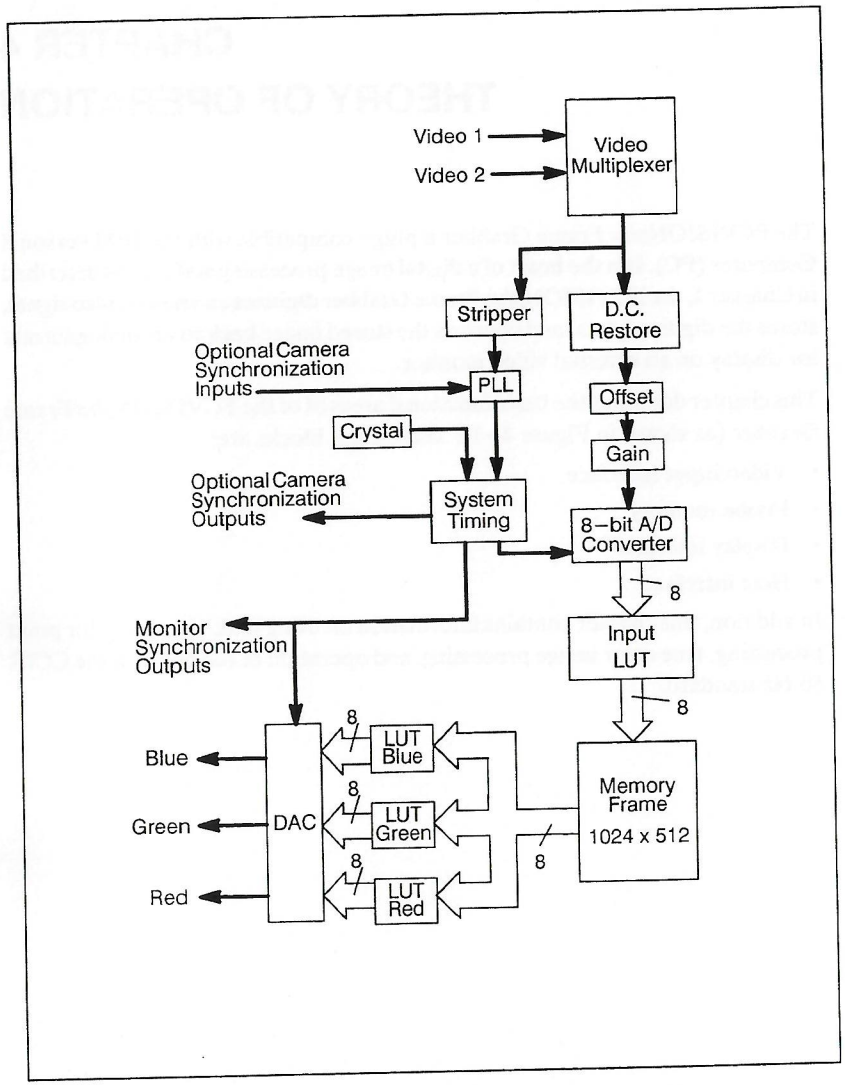


Figure 4–1. PCVISION*plus* Block Diagram

4.1 VIDEO INPUT INTERFACE

The video input interface performs timing, synchronization, signal conditioning, digitization, and pixel transformation.

4.1.1 System Timing

Two timing options are available for synchronizing the video source and the PCVISION*plus* Frame Grabber. The first option, called crystal mode, allows the Frame Grabber to generate all system timing internally, based on an internal crystal oscillator. In this mode, the video source synchronizes to the timing signals provided by the PCVISION*plus* Frame Grabber. The second synchronization option, called phase–locked loop (PLL) mode, extracts the synchronization information from the video signal by a sync stripper. The output of the sync stripper is then passed to a PLL. This process is illustrated in Figure 4–2. The PLL locks the internal timing of the PCVISION*plus* Frame Grabber to noisy video sources , such as video cassette recorders. If the video signal is lost, the PCVISION*plus* switches timing automatically to crystal mode to avoid the loss of any image data.

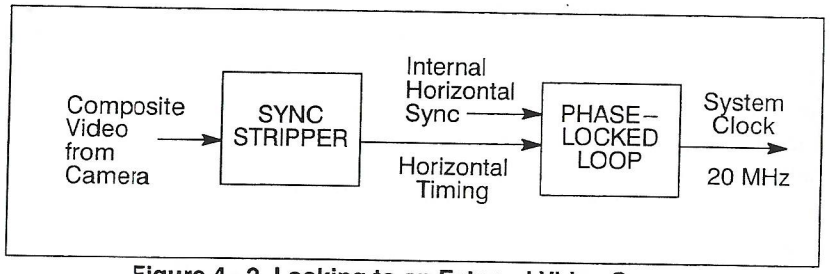


Figure 4–2. Locking to an External Video Source

Use of the PLL or Crystal mode is determined by the CLOCK bit in the Acquire/Status register. A detailed discussion of how to configure the PCVISION*plus* Frame Grabber for different synchronization modes is found in Chapter 2.

4.1.2 Signal Conditioning

The PCVISION*plus* Frame Grabber conditions the analog signal, to ensure that picture quality is maintained at all times, using a process known as DC restora-

tion. This process compensates for any drift in the voltage level of the incoming video. The horizontal blank portion of the video signal (shown in Figure 4-3) contains a section called the "back porch." This back porch is a voltage level representing the intensity of black. During each horizontal blank, the back porch is sampled. If an offset from black is present, the analog video signal for the subsequent horizontal line is adjusted by this offset.

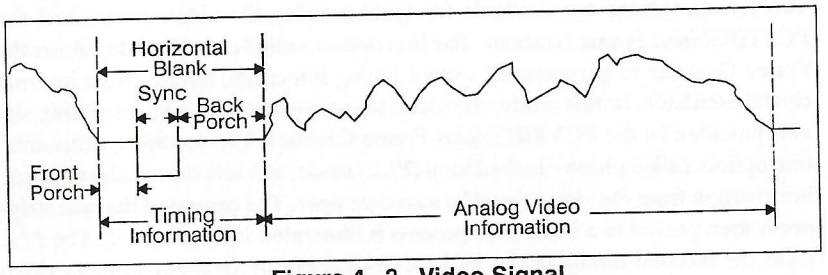


Figure 4-3. Video Signal

4.1.3 Programmable Offset and Gain

Programmable offset voltage and gain circuits are provided for increased flexibility when digitizing video sources that do not conform to the RS-170/330 or CCIR specifications or when smaller sections of video signal must be digitized over the full gray scale range of 256 bits. Gain boosts or attenuates the incoming signal. The gain is adjustable from 0.67 to 1.33 in 100 steps (.0066 per step). Offset adds a constant voltage to the signal, to shift it up or down. The offset is adjustable from -0.854 to +0.266 Volts in 100 steps (1.12 mV per step).

The circuit controlling programmable gain and offset has a 7 microsecond time-out. Some personal computers can write data faster than this. Your program must allow a delay of 10 microseconds between write accesses to the gain and offset counter.

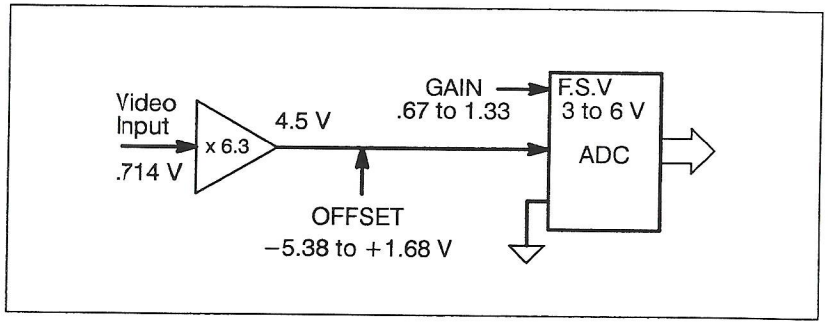


Figure 4−4. Signal Conditioning

The gain counter adjusts the ADC full−scale voltage from 3 to 6 Volts. The minimum gain of .67 corresponds to counter position 99 or ADC full−scale 6 Volts. The maximum gain of 1.33 corresponds to counter position 0 or ADC full−scale 3 Volts.

The offset counter adjusts the added offset after the 6.3 gain amplifier from −5.38 to +1.68 Volts. The offset is −5.38 Volts at counter position 0. The offset is +1.68 at counter position 99. Dividing these voltage values by 6.3 gives the offset adjustment range of −0.854 to +0.266 Volts at the video input.

4.1.4 Digitization

Digitization is the process that converts the analog video signal into a series of digital values or pixels. This is performed by sampling the analog signal at discrete time intervals and converting each individual sample to a digital value. The 10 MHz PCVISION*plus* Frame Grabber samples the analog signal 10 million times per second. This results in 512 pixels for each horizontal line in the image. The 12.5 MHz version samples at 12.5 million times per second, resulting in 640 pixels per line. The RS−170/330 standard specifies 480 lines of digitized information per frame. The CCIR standard uses 512 lines per frame.

The digitization process is accomplished with an analog−to−digital converter. The PCVISION*plus* Frame Grabber digitizes to an accuracy of eight bits per pixel or 256 gray levels.

4.1.5 Input Look-Up Tables

A look-up table (LUT) is a list or table of 256 values. The list of numbers programmed corresponds to the solutions to a transformation formula. The incoming data (256 possible levels) is used as a table address for a new data value. The output value is dependent upon the value of the input and the transformation formula. Some examples of LUTs are given at the end of this chapter. Eight input LUTs transform the image data before it is stored. One LUT is always selected for input transformation. The LUTs are programmable through the PC bus interface.

4.2 FRAME MEMORY

The frame memory is organized as a 1024 by 512 array of 8-bit pixels. Two 512 by 480 (512 x 512 for CCIR) images or one 640 by 480 image can be stored. The frame memory architecture allows simultaneous acquisition and display of images, has multiple access modes, and includes a Pixel Buffer to speed memory access and reduce contention between display scanning and host access.

4.2.1 Simultaneous Acquire and Display

The frame memory can simultaneously acquire and display an image. This is accomplished with a read/modify/write cycle called the "pixel scan cycle." The pixel that is read from frame memory is transmitted to the display logic for digital-to-analog conversion. The data is modified by the digitizing logic. The new data is written into the frame memory. A lag of one frame time (1/30th second) exists between the acquired image and the displayed image when the PCVISION*plus* Frame Grabber is simultaneously acquiring and displaying an image.

4.2.2 Frame Memory Write-Protection

The PCVISION*plus* contains two mask registers, the Host Mask and Video Acquisition Mask. These mask registers permit write-protection of individual bit-planes during video acquisition, host access, and clear operations.

The frame memory can be thought of as a 1024 by 512 plane that is eight bits or eight bit-planes deep. Bit-plane 0 is the LSB or bit 0 of all 1024 by 512 locations. Bit-plane 7 is the MSB of all 1024 x 512 locations or pixels. Protecting a bit-

plane protects the bit or bits of all pixels in frame memory on a specific bit–plane. The Host Mask and Video Acquisition Mask Registers are described in greater detail in Chapter 5.

4.2.3 Pan and Scroll

The portion of frame memory being acquired and displayed is referred to as the Active Video Window (AVW). The Pan and Scroll registers define the position of the top left–most pixel of the AVW. This top left–most pixel is referred to as the AVW address because it is the first pixel scanned from the memory and displayed on the monitor. The Pan register controls the horizontal position, and the Scroll register controls the vertical position of the AVW. The active video window is moved around in frame memory, resulting in the image moving around on the display monitor.

When using the 10 MHz rate board, the pan value is used to position the AVW to the second half of frame memory to acquire and display a second image. The resolution of the Pan register is eight pixels. Every increment of the Pan Register moves the AVW horizontally eight pixel positions. The resolution of the Scroll register is two lines. Every increment of this register moves the AVW vertically two line positions.

4.2.4 Zoom

The PCVISION*plus* Frame Grabber can enlarge "zoom" an area of frame memory by a factor of two. Zooming an image by a factor of two means that a 256 by 240 pixel area of the frame memory is expanded to fill a 512 by 480 display area on the monitor. To produce a zoom factor of two, each pixel and each line read from the frame memory is replicated twice upon display.

Zoom also affects video acquisition. In zoom mode every other pixel from the A–to–D Converter is stored in sequential horizontal frame memory locations. Every line of the even field in the video signal is stored in sequential lines of frame memory. The odd frame is not recorded. An RS–170/330 image sampled at 10 MHz only occupies 256 pixels by 240 lines in frame memory. The frame memory can store eight images of this type. The Pan and Scroll registers determine the position the individual images in frame memory.

4.3 DISPLAY INTERFACE

The display interface consists of two parts: the output look-up tables and the digital-to-analog converters, as shown in Figure 4-5.

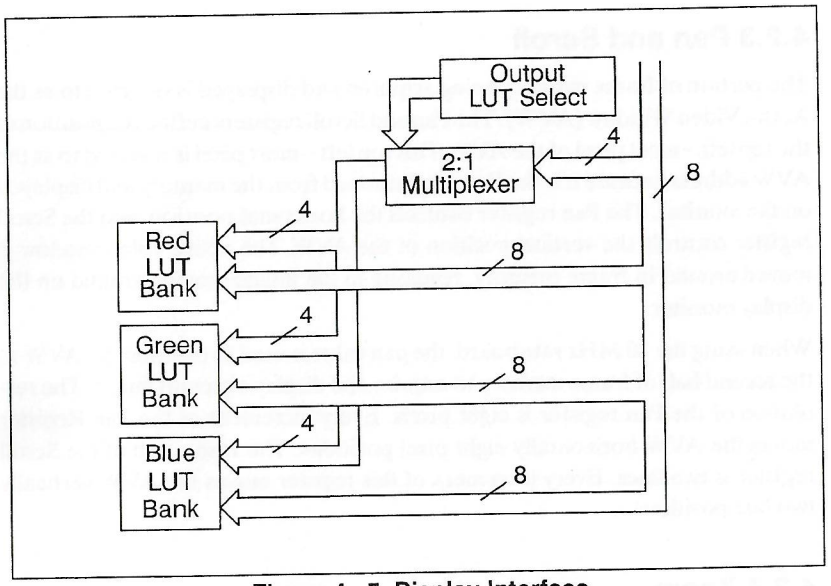


Figure 4-5. Display Interface

4.3.1 Output Look-Up Tables

Look-up tables are used to transform pixel intensities prior to display. Each LUT is a table of 256 bytes and is programmable through the PC bus interface. There are eight LUTs for each of the primary colors (red, green, and blue). Each LUT can have a different transform. LUTs are programmed individually. LUTs are selected as a "bank" during the display process. The same numbered LUT is simultaneously selected from each color. When LUT 0 is selected, Blue LUT 0, Red LUT 0, and Green LUT 0 are all selected and used to transform pixel data.

The eight bits of pixel information from the frame memory are used as an address into the LUT. The value sent to the digital-to-analog converter is the value in the LUT at that address. If the LUT is programmed as a ramp (Figure 4-6), no

transformation of the pixels occurs. If the LUT is programmed for binary thresholding as shown in Figure 4–7, the image displayed on the monitor has only two intensities, 0 and 256 (black and white on a monochrome display). The existence of eight LUTs per channel allows you to load each table with different transformation data and quickly switch between them. It is not necessary to re–load or reprogram the LUTs each time you want to change the transformation performed.

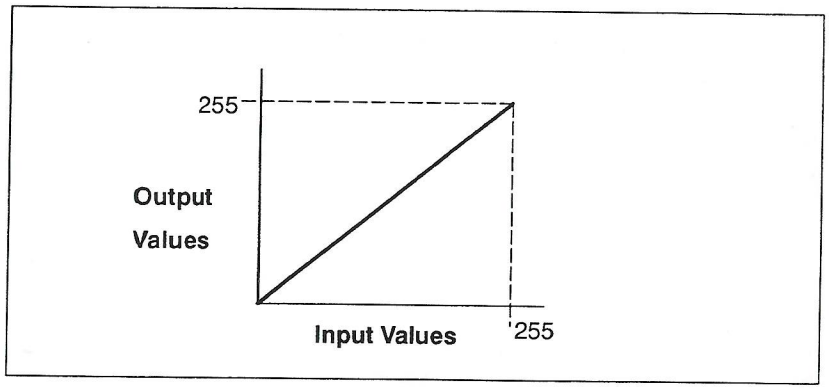


Figure 4–6. LUT Programmed for a Ramp

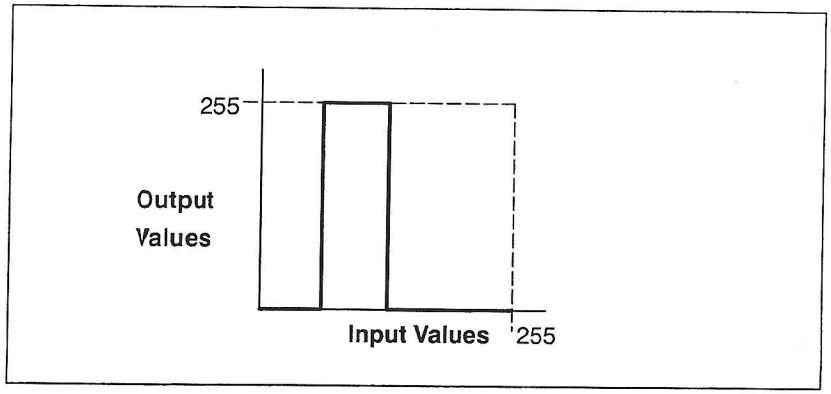


Figure 4–7. LUT Programmed for a Threshold

4.3.2 Digital-to-Analog Conversion

The output of the LUTs is the input to the digital-to-analog converters. The digital-to-analog converters change the digital information back into an analog format for display on a monitor. A composite sync signal that drives the display monitor can be output on the green video channel, or separately as a single composite sync signal, or as separate horizontal and vertical sync signals.

4.4 HOST INTERFACE

The PCVISION*plus* Frame Grabber is directly compatible with the bus structure found in the IBM Personal Computer. This interface is the communication path between the PC and the PCVISION*plus* Frame Grabber. This interface has two parts:

- Control Register Interface
- LUT Memory Interface
- Frame Memory Interface

The PCVISION*plus* Frame Grabber occupies 16 bytes in the I/O space of the PC for its control registers. Of these 16 bytes, the first twelve are currently used. The remainder are reserved for future use and must not be accessed. Refer to Chapter 5 for a detailed discussion of the registers.

The base address of the registers is jumper-selectable to any 16-byte boundary in the I/O space of the PC. If multiple PCVISION*plus* Frame Grabbers are present in the system, each is controlled individually and must reside at a unique register base address. Chapter 2 describes how to set the register base address.

4.4.1 Programming the Look-Up Tables

The look-up tables (LUTs) transform individual pixels. Frequently, the LUTs are programmed linear (a ramp function), as shown in Figure 4-6. The ramp function or linear LUT performs no transformation of the digitized data occurs. An example at the end of this chapter discusses how to program LUTs for a ramp function.

Binary thresholding is a common technique using the LUTs. This technique involves mapping the pixel data into one of two possible values, normally black and white in a monochrome display.

For example, to display all pixel intensities from 20 (decimal) to 30 as white, and all other intensities as black, program a LUT with a step function, as shown in Figure 4–7.

4.4.2 LUT Programming Example

The following example describes how to initialize blue output LUT 1 for a ramp function. Figure 4–8 is a flow chart of these procedures.

1. In the LUT Control register, select the blue LUT bank by setting LUTMOD(1–0) to 10B.
2. Select the second blue LUT by setting OLUT(2–0) to 001B.
3. Define an integer variable X (for use as a counter) and initialize it to 0.
4. Load X into the look–up table Address register.
5. Load X into the Look–Up Table Data register.
6. Increment the counter (X).
7. Repeat steps 4, 5, and 6 until X = 255, then exit.

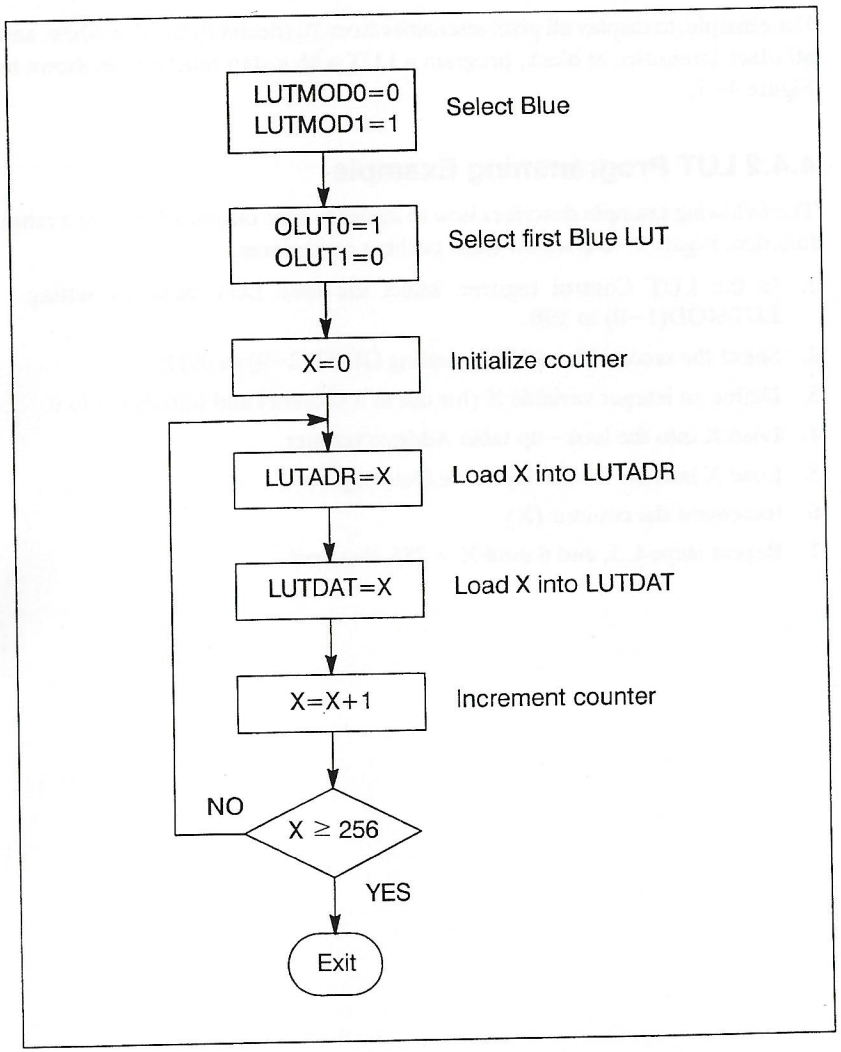


Figure 4–8. Initializing a Blue LUT to a Ramp

4.4.3 Frame Memory Interface

The 1024 by 512 by 8–bit frame memory is divided into eight equally sized blocks, 64K bytes each. Using the Memory Block Select bits (BLOCK) in the Control register, each of the eight blocks are individually selected for access in the memory address space of the PC starting at the memory base address. Frame memory is enabled onto the PC bus by a Board Select (BDSEL) bit in the Control register. To enable the frame memory, BDSEL must be set (1). This allows mapping of more than one PCVISION*plus* Frame Grabber to the same 64K byte block in the memory space of the PC, thereby reducing the amount of host memory space needed in full color systems. All Frame Grabbers can have the same memory base address, as long as only one is enabled at any given time.

Chapter 2 describes how to change the mapping configuration which is hardware–selectable and independent of the sampling rate. The 10MHz board, capable of storing two images, is shipped from the factory with the dual–store frame memory configuration shown in Figure 4–9. Each block is 512 pixels wide and 128 lines deep. To the host it appears as 64K consecutive data bytes starting at the memory base address. The 12.5MHz board, capable of storing one image, is shipped from the factory with the single–store frame memory configuration shown in Figure 4–10. Each block is 1024 pixels wide and 64 lines deep. To the host it appears as 64K consecutive data bytes starting at the memory base address.

NOTE

Some ITEX™ functions are affected by the memory configuration. The dual–store configuration makes save, restore, and single–image operations faster. ITEX Geometric functions like "warp" and "copy_area" require a single–store memory configuration.

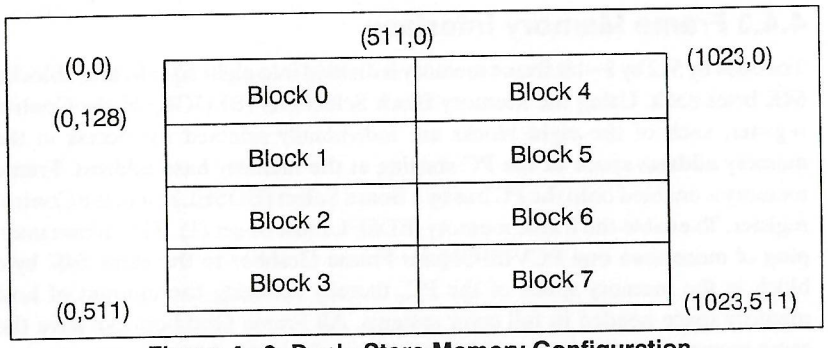


Figure 4-9. Dual-Store Memory Configuration

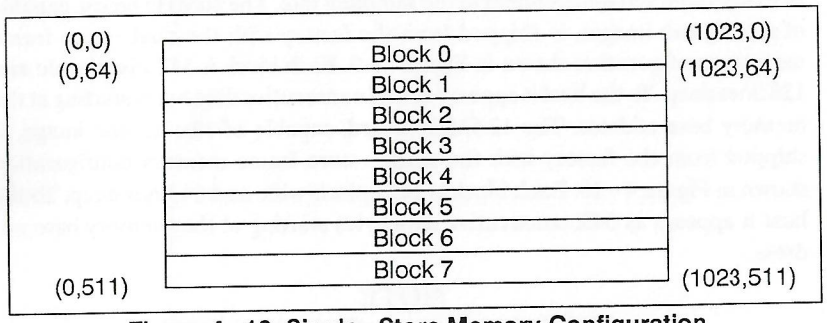


Figure 4-10. Single-Store Memory Configuration

4.4.4 The Pixel Buffer

Standard frame memory architectures are dual–ported to allow the scanning logic to refresh and display sequential pixels at video rates. At the same time, the host needs access the frame memory under the control of the application program. If the host tries to access the memory during the scan cycle, contention arises. Typically this contention is resolved by forcing the host to wait for the scan cycle to end before it accesses the memory. It must, in effect, "wait its turn." To eliminate these wait states, Imaging Technology has developed a frame memory architecture that includes the Pixel Buffer and related mechanisms to speed up host access to pixel data.

The Pixel Buffer is a storage register that buffers eight 8–bit pixels. All data transferred to or from frame memory passes through the Pixel Buffer. Access to the Pixel Buffer is provided via the Pixel Buffer Register. The Pixel Buffer contents are loaded from or written to frame memory by executing a single host memory access cycle.

Every memory read cycle loads the addressed pixel into the Pixel Buffer, along with the next seven consecutive pixels in frame memory. Therefore, frame memory is read only once to fill the Pixel Buffer. The Pixel Buffer is read eight times. This reduces the potential number of wait states due to frame memory accesses from eight to one. The Pixel Buffer can be viewed as a "pixel cache."

It is important to keep memory access on eight pixel boundaries. In other words, the first pixel addressed or accessed must be zero or a multiple of eight. If this rule is not followed, the Pixel Buffer will wrap within an eight–pixel boundary, and scramble the data read or written.

NOTE

Empty the Pixel Buffer after every host access. Failure to do so can result in wrong data the next time the host transfers data.

4.4.5 Emptying the Pixel Buffer

Be aware of the pixel buffer's status. If a previous memory access has left data in the Pixel Buffer, a write access transfers the pixel buffer contents to frame memory. Empty the pixel buffer before a memory write if in doubt.

1 Program PBEN to one allowing modification of the pixel buffer's contents.

2 Write to a frame memory address containing no image data to reset the pixel counter and empty the pixel buffer.

4.4.6 Frame Memory Access Modes

The PCVISION*plus* Frame Grabber provides two different pixel access modes that allow powerful data manipulation:

Z-Mode Access - Each host memory cycle accesses single 8-bit pixels.

X-Mode Access - Each host memory cycle accesses eight horizontally-adjacent bits from one specific bit-plane.

The access mode is controlled by the TMODE bit in the Memory Access Control register. Figure 4-11 illustrates these two modes, and block moves, discussed later in this chapter.

Single pixels are transferred by selecting Z-mode, and enabling the Pixel Buffer (PBEN=1). Perform single read or write cycles to or from frame memory without loading the Pixel Buffer, or using the Pixel Buffer Register.

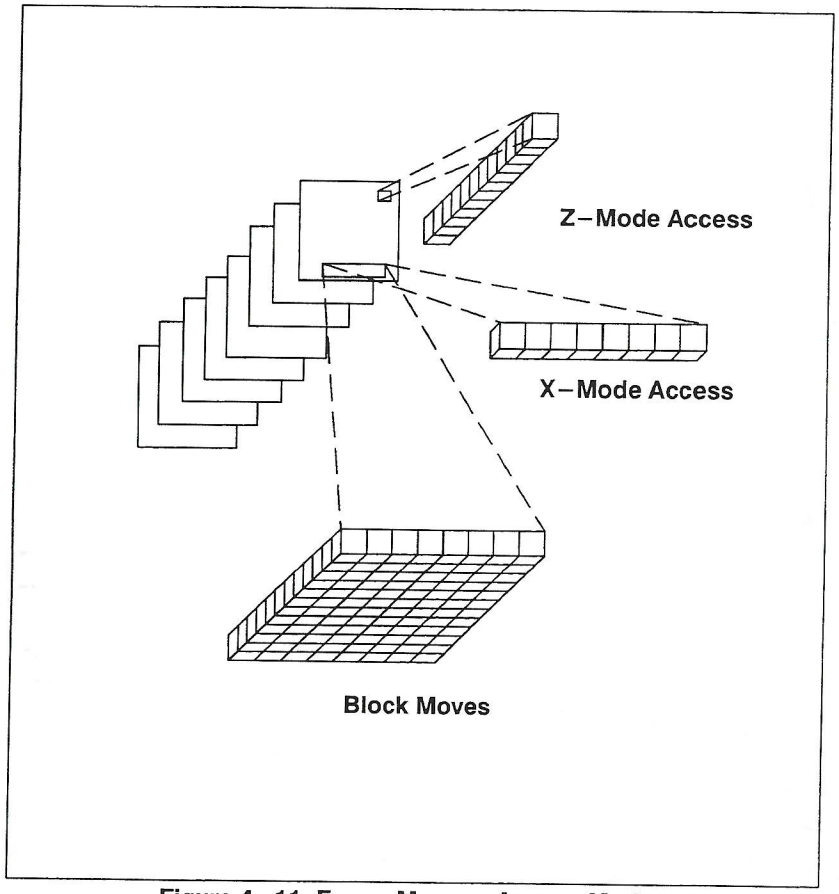


Figure 4–11. Frame Memory Access Modes

4.4.6.1 Z-Mode Memory Access

Z-Mode is the most common way memory is accessed. An 8-bit pixel is transferred between the host and frame memory with the least significant bit corresponding to bit plane 0 and the MSB corresponding to bit plane 7.

4.4.6.2 X-Mode Memory Access

X-Mode access is unique in allowing access to a single bit plane from eight horizontally-adjacent pixels. The bit-plane is selected by the PS(2-0) bits in the Memory Access Control register. X-Mode is used for graphic operations and bit-plane-oriented data transfers.

X-Mode and Z-Mode determine the configuration of the data byte in *all* frame memory data transfers. The next section of this chapter clarifies these two access modes when the Pixel Buffer mechanism is employed.

4.4.7 Memory Read Cycles

Each time the memory is read, eight consecutive (8-bit) pixels from the frame memory are automatically loaded into the Pixel Buffer. The starting address *must* be on an eight pixel boundary. All eight pixels are read from Pixel Buffer without contention problems between host and scan access. This means faster net access to the eight pixel groups than if the pixels were accessed with individual memory read cycles. The first pixel must be read twice, once during the memory read, as the data byte from the read cycle, and again on the first access to Pixel Buffer (via the Pixel Buffer Register).

4.4.8 Reading the Pixel Buffer

The pixels are read from the Pixel Buffer Register according to the transfer mode selected in the Memory Access Control register. All memory read cycles fill the Pixel Buffer with eight consecutive pixels. The transfer mode controls the way the host reads pixels from the Pixel Buffer.

4.4.8.1 Reading the Pixel Buffer in Z–Mode

In Z–Mode, the Pixel Buffer is a FIFO stack. Every I/O read cycle of the Pixel Buffer Register produces the next pixel in the Pixel Buffer up to the eighth pixel. The first access reads Pixel 1, the second access reads Pixel 2, and so on until all eight pixels have been read. Figure 4–12 shows the Pixel Buffer, and the first read access of the Pixel Buffer Register.

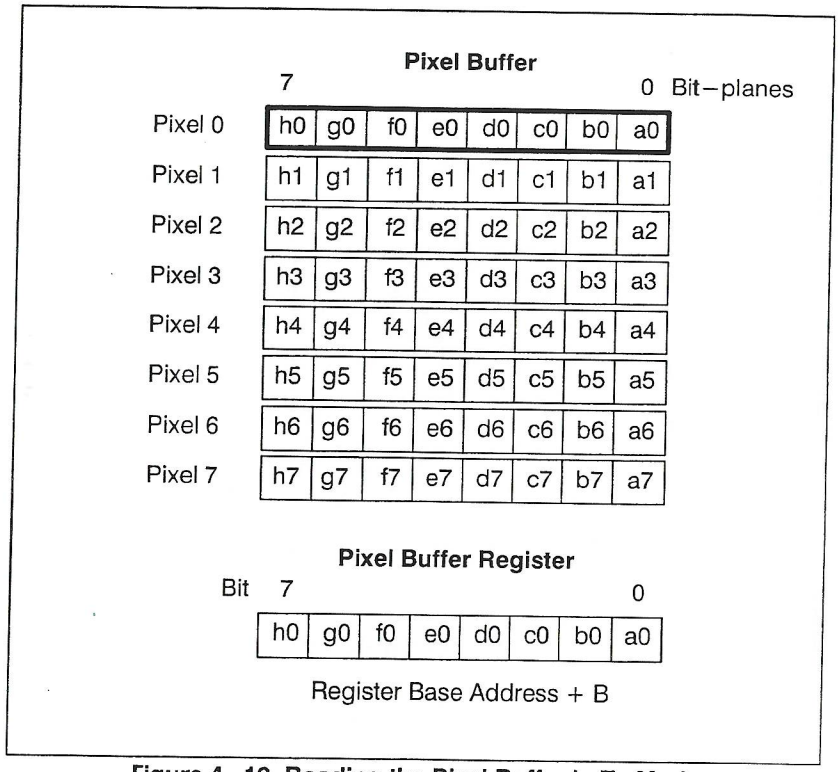


Figure 4–12. Reading the Pixel Buffer in Z–Mode

4.4.8.2 Reading the Pixel Buffer in X-Mode

The I/O read cycle of the Pixel Buffer Register reads a single bit from each of eight horizontally-adjacent pixels in the Pixel Buffer, from one bit plane selected by the plane select (PS) bits of the Memory Access Control register.

Figure 4-13 shows an example where the PS bits are set to 001 (binary), indicating a desire to read bit plane 1. This bit plane is returned in the Pixel Buffer register.

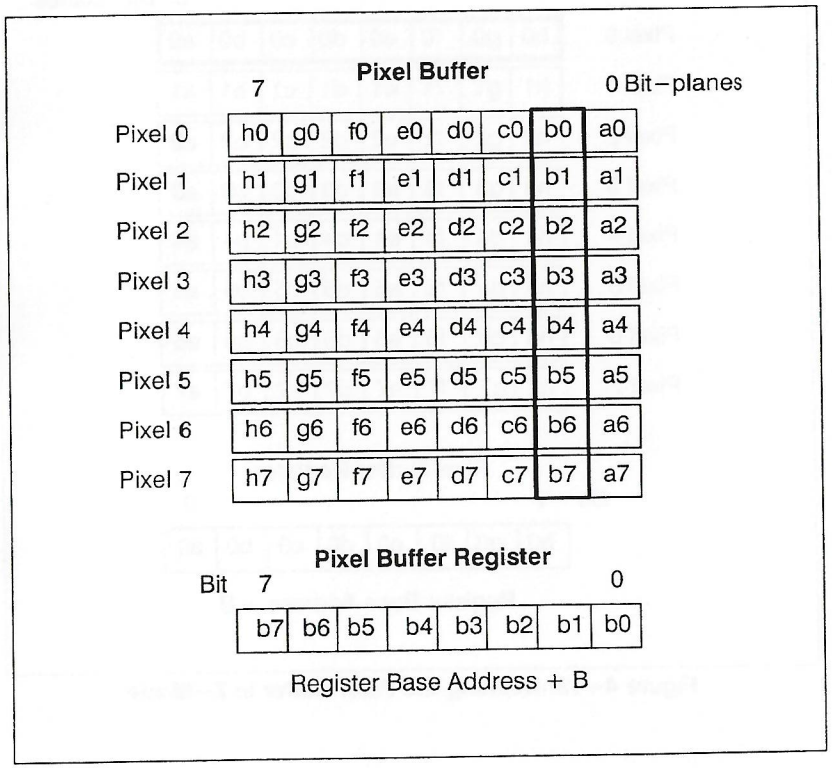


Figure 4-13. Reading the Pixel Buffer in X-Mode

4.4.9 Loading the Pixel Buffer

The Pixel Buffer is loaded with the desired pixels using either X–Mode or Z–mode. The mode selected has a very important effect on the Pixel Buffer data.

4.4.9.1 Loading the Pixel Buffer in Z–Mode

In Z–Mode, the Pixel Buffer is a FIFO with an internal counter, called the Pixel Counter. A frame memory write operation resets the Pixel Counter to one. Every I/O write cycle to the Pixel Buffer Register pushes the new pixel on the top of Pixel Buffer and increments the counter. If more than eight pixels are written to the Pixel Buffer Register, the oldest data is lost from the "bottom" of the Pixel Buffer, the Pixel Counter remains at eight, and will not wrap around. Figure 4–14 shows the Pixel Buffer Register writing a fourth pixel onto the Pixel Buffer.

The Pixel Counter is only used during Z–mode write operations. This includes writing to the Pixel Buffer Register, and writing to frame memory using the Pixel Buffer.

4.4.9.2 Loading the Pixel Buffer in X–Mode

In X–Mode, the Pixel Buffer is "flooded" when a single byte is written to the Pixel Buffer Register. This means that all eight bit planes are loaded with the same pattern by a single write cycle. This is useful when filling large areas of frame memory with repetitive patterns or modifying specific bit planes in frame memory. Figure 4–15 shows the Pixel Buffer after a single write to the Pixel Buffer Register.

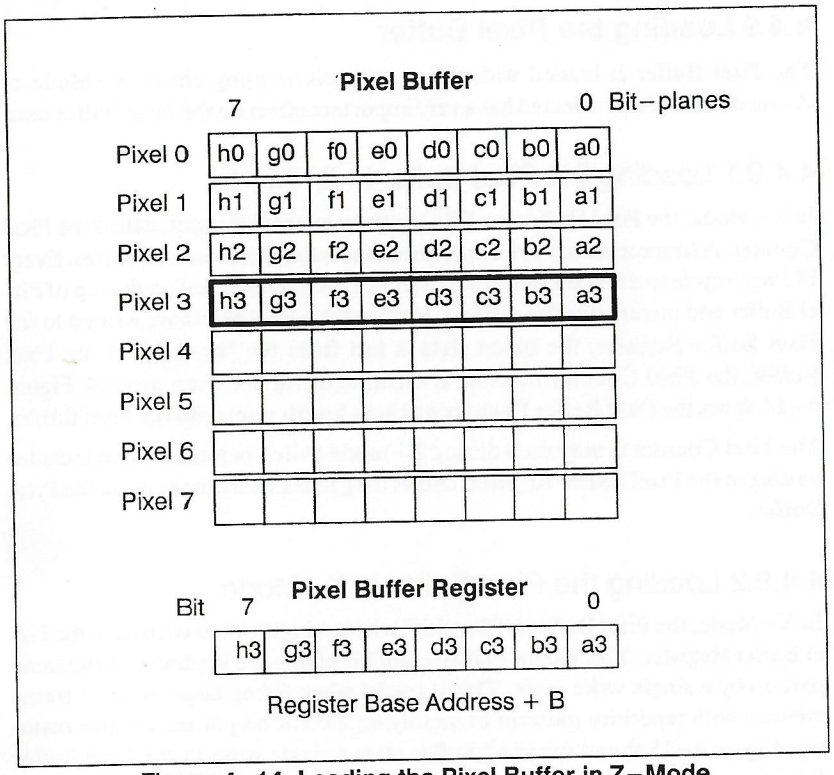


Figure 4–14. Loading the Pixel Buffer in Z–Mode

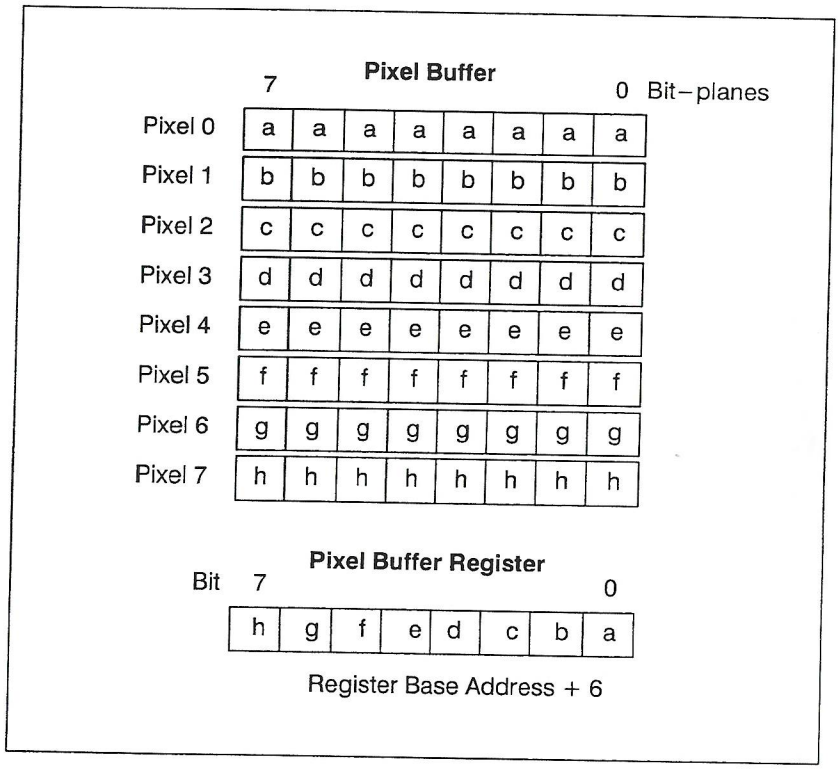


Figure 4–15. Loading the Pixel Buffer in X–Mode

4.4.10 Memory Write Cycles

Memory write cycles conditionally transfer the contents of the Pixel Buffer to frame memory. This implies that the Pixel Buffer already contains pixels that you wish to store in frame memory. The mechanism of this transfer is dependent on:

1 The mode selected (X–Mode or Z–Mode).

2 The state of the PBEN bit (in the Memory Access Control register).

4.4.10.1 Memory Write Cycles in Z-Mode Pixel Buffer Enabled (PBEN=1)

When PBEN is set (1), the PCVISION*plus* transfers the number of pixels in the Pixel Buffer (as determined by the Pixel Counter) into consecutive locations in frame memory. The data written by the host during the frame memory write cycle becomes the last pixel transferred to frame memory. Eight-pixel boundaries must be respected, for the reason already explained in "The Pixel Buffer."

To transfer eight pixels, the host writes the first seven to the Pixel Buffer (via the Pixel Buffer Register), then performs a memory write. During this memory write operation, the write data is the eighth pixel and is put into the Pixel Buffer. The eight pixels contained in the Pixel Buffer are simultaneously transferred into frame memory. To transfer five pixels, write the first four pixels to the Pixel Buffer Register, and write the fifth to frame memory.

An alternative mode of operation is the transfer of single pixels. In this case, select Z-Mode with PBEN=1 (Pixel Buffer enabled), and perform single read or write cycles to or from frame memory without loading the Pixel Buffer, or using the Pixel Buffer Register.

4.4.10.2 Memory Write Cycles in Z-Mode Pixel Buffer Protected (PBEN=0)

When you set PBEN to 0 after loading the Pixel Buffer, the PCVISION*plus* transfers the Pixel Buffer data into consecutive locations in frame memory. Data from the host is *ignored*, and the Pixel Buffer contents are preserved after the write cycle. The data stored in the Pixel Buffer is used repeatedly without being destroyed, allowing the drawing of patterns, such as wedges or bars, at high speeds. This mode is also used in clear operations.

For example, Figure 4-14 shows four pixels in the Pixel Buffer. Clear PBEN in the Memory Access Control register to freeze the Pixel Counter at four. When the host performs a memory write, these four pixels are transferred to frame memory in consecutive locations. Since the Pixel Buffer is preserved, another frame memory write cycle transfers the same four pixels. This continues until the Pixel Buffer is again enabled so the Pixel Buffer data can be modified.

4.4.10.3 Memory Write Cycles in X–Mode

In X–Mode, the entire Pixel Buffer is transferred to frame memory. The counter has no effect. If you wish to write–protect bit planes, use the Host Mask register. This mode is useful in fill operations or any operation that requires modification of a neighborhood of pixels. PBEN has no effect in this case.

4.4.11 Block Moves

The pixel buffer moves blocks of frame memory data from one address to another without transferring any data to the host computer. For each block of eight pixels, the host computer performs a read from one memory address that loads eight consecutive (8–bit) pixels into the pixel buffer. By performing a single Z–mode memory write from the host computer, the same eight pixels transfer to frame memory at a second specified memory address. The length of the data block ranges from one to eight pixels, as specified by the pixel counter. Data cannot be read or written across an eight–pixel boundary. Attempting to cross an eight–pixel boundary causes the data written to wrap around within the boundary, resulting in a scrambled image.

To move blocks that contain N bytes where N is less than eight:

1 Enable Z–Mode.

2. Set PBEN.

3. Write to frame memory (resets the Pixel Counter to 1.)

4. Write to the Pixel Buffer Register N–1 times (sets Pixel Counter to N.)

5. Clear PBEN.

6. Read memory (transfers eight pixels to the Pixel Buffer.)

7. Write to memory (transfers N pixels to memory.)

Using a variation of the block move function, a sub–area (area–of–interest) of frame memory can be set to a particular pattern. This is done by loading the Pixel Buffer with the desired pattern, protecting the Pixel Buffer, and then continuously executing frame memory writes to transfer the contents of the Pixel Buffer into frame memory. Patterns can be generated using either X–Mode or Z–

Mode transfers. Loading and writing in Z–Mode uses the counter, X–Mode does not. The modes can be intermixed in load, read and write operations. This operation is performed very quickly, since each frame memory write transfers up to eight pixels simultaneously.

4.4.12 How to Clear Memory

It is possible to clear the frame memory in one frame time. To do so, you must perform the following:

1. Set PBEN to 1.
2. Enable Z–Mode.
3. Write all zeroes to the Pixel Buffer eight times.
4. Clear PBEN to protect the contents of the Pixel Buffer.
5. Initiate the memory clearing function by setting the ACQMD bits in the Acquire/Status register to 01B.

Protect specific bit planes from the clearing function by setting the appropriate bits in the Host Mask register. Clearing the entire frame takes one frame time, or approximately thirty milliseconds (30 ms).

Set the screen to a specific value by writing some number other than 0000H to the pixel buffer. Unprotect the pixel buffer and perform eight Z–mode writes to the pixel buffer register to fill the pixel buffer. Protect the pixel buffer and use steps 4 through 7 for clearing the frame memory.

4.5 TRUE COLOR IMAGE PROCESSING

Three PCVISION*plus* Frame Grabbers can perform true color image processing. In this configuration, each module receives and digitizes one of the three video signals from an RGB video source.

The three PCVISION*plus* Frame Grabbers are linked via a ten–conductor ribbon cable that connects to ports on the top of the modules. One of the modules is configured as the master and the other two are configured as slaves. The master generates the system timing for all three PCVISION*plus* Frame Grabbers in the system.

When multiple PCVISION*plus* Frame Grabbers are installed, the register base address for each must be unique in the I/O space of the PC. If desired, the frame memory for each module can be mapped into the *same* 64K byte block in the memory space of the PC, by using the same memory base address on all three. Use the board select (BDSEL) bit in the Control register to enable only one frame memory into the 64K byte block at any one time.

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP