HEWLETT-PACKARD

JOURNAL

June 1993

HEWLETT PACKARD

The HP Ultra VGA Graphics Board

By increasing the display memory to 1M byte and providing some local graphics processing, the HP Ultra VGA board is able to increase VGA resolution to 1024 by 768 pixels with 256 colors at all resolutions.

by Myron R. Tuttle, Kenneth M. Wilson, Samuel H. Chau, and Yong Deng

The HP D2325A Ultra VGA board, which represents the latest in the evolution of HP personal computer video systems, is a video accessory card for the HP Vectra line of personal computers. This board offers exceptional video performance for graphics-intensive applications such as Microsoft® Windows and AutoCAD™. It enhances overall system performance by using hardware accelerators to relieve the CPU of common video processing functions. For high-resolution and flicker-free operation, the Ultra VGA board offers display resolutions up to 1024 by 768 noninterlaced and refresh rates up to 72 Hz. Finally, the board is upgrada to 1M bytes of video memory to give 256 colors in 800 by 600 and 1024 by 768 resolutions.

In this article we present a brief history of the evolution of PC video systems. We will then discuss the benefits of adding acceleration to video hardware and the hardware and software partitioning trade-offs that must be made. Finally, the implementation of the HP Ultra VGA board is described, both as a plug-in accessory and as an embedded feature as it is in the HP Vectra 486/U family of personal computers.

Evolution of PC Video

1981 marked a dramatic change in the world of computing because that was the year the IBM personal computer was introduced. The first IBM PC came equipped with 64K bytes of RAM and an alphanumeric Monochrome Display Adapter (MDA). The MDA contained 4K bytes of memory, enough to store one screen (25 lines of 80 characters each) of alphanumeric information. The PC with one of these adapters functioned like most terminals available at the time. It had very clean alphanumerics but lacked any graphical capabilities. Until the introduction of the MDA, virtually all PCs or "home computers" such as the Apple II, Commodore PET, and the Tandy Radio Shack TRS-80 used a television monitor or a modified television monitor as a display, grossly limiting the resolution.

Also available from IBM in 1981 was the Color Graphics Adapter (CGA). This adapter contained 16K bytes of memory, enough to hold four alphanumeric pages and provide limited-resolution graphics. The graphics capabilities of the CGA allowed it to display 320 by 200 pixels in four colors, or 640 by 200 pixels in two colors. The price of memory was still a limiting factor in display resolution. The 200-line vertical resolution severely impacted the CGA's alphanumeric capabilities because all characters were displayed in an eight-by-eight cell and were difficult to read. Several companies, including Hewlett-Packard, introduced their own extensions to the CGA, allowing greater resolution. The HP 45981A multimode adapter increased the resolution to 400 lines but kept the same horizontal and color resolutions and increased the memory to 32K bytes. The CGA became the lowest common denominator for graphics-based programs and, in fact, is still supported today by many applications—especially games.

In 1982 the Hercules Company introduced the Hercules Graphics Card (HGC). This adapter fully supported the alphanumeric capabilities of the Monochrome Display Adapter as well as providing 720-by-348-pixel monochrome graphics. Because of its modest cost and industry support, the HGC became very popular.

The next big breakthrough in PC video came in 1985 when IBM introduced the Enhanced Graphics Adapter (EGA). This was the first affordable PC video adapter to enter the "high-resolution" arena. It supported a resolution of 640 by 350 pixels with up to 16 colors simultaneously displayed from a palette of 64 colors. The memory required was 128K bytes. The EGA was fully backward compatible with the CGA (and with the monochrome monitor, the MDA). In 1987 the IBM PS/2 line of PCs was introduced and with it the Video Graphics Array (VGA) video adapter. The VGA has become the de facto standard of the PC industry today. The original VGA contained 256K bytes of video memory and supported resolutions up to 640 by 480 pixels with up to 16 colors simultaneously displayed from a palette of 262,144 colors. As memory prices have continued to decrease, the VGA has been enhanced. The first enhancement was the Super VGA (SVGA) which increased the resolution to 800 by 600 (or 1024 by 768) pixels. The color depth increased to allow up to 256 colors to be simultaneously displayed. Display memory was increased to 512K bytes. The VGA is fully backward compatible with the CGA, EGA, and HGC video adapters.

The video adapters mentioned above map the display memory into the system processor's memory space. All video and graphics operations are handled directly by the system processor. Newer display adapters, such as the HP Ultra VGA board, have taken the next step by increasing the VGA resolution to 1024 by 768 pixels with 256 colors available at all resolutions (increasing the display memory to 1M bytes) and providing some local graphics processing in the video display system. This frees the system processor from much of the work of updating the display and accelerates display operations.

HP Ultra VGA Board Implementation

In any design there are always trade-offs to improve performance, save board space, add more features, and so on. The Ultra VGA board implementation was confronted with some of these same trade-offs as well as the need to adopt some new technologies to provide a high-resolution, flicker-free graphics system.

Software versus Acceleration Trade-offs

In almost all nonscientific programs, video processing is the performance bottleneck. By taking some of the graphics burden off the applications, a good video solution is able to improve overall system performance dramatically. This is especially true as graphics-oriented user interfaces become more popular.

Performing the high-level graphics functions like area fill and line drawing inside the hardware has only become popular in the last few years. The MDA and CGA video solutions used video routines located in the main system BIOS (basic input/output system) of the computer and offered a few low-resolution modes. The EGA and VGA were logical extensions to MDA and CGA They offered more modes, higher resolutions, more col , and had their own video BIOS. There was no support for any high-level graphics functions, though a few simple graphical mixing functions like XOR were available. It wasn't until the IBM 8514 that high-level graphic functions were performed in the hardware. The 8514 is an accelerated display adapter that contains a graphics engine implemented in hardware. The problem with the 8514 is that it is not backward compatible with VGA.

Since the advent of the 8514, other video manufacturers have started to put high-level graphics functions and support for VGA on the same card. Some manufacturers add the capabilities of high-level graphics functions directly to the VGA modes while others add special video modes and VGA modes that have the extra capabilities the 8514 made popular.

Trade-offs must be made to determine where the high-level graphics functions like line drawing are implemented. In the past, an application would use the CPU to calculate all the points in a line and then write each separate dot in the line to video memory. This works very well if the CPU has bandwidth to spare and all video solutions behave the same. Since all video solutions do not work the same, each application has to either pander to the lowest common denominator or not work on all machines. Two main solutions to this problem have been implemented: video BIOS interrupt calls and application drivers.

The HP Ultra VGA video BIOS contains an industry-standard set of interrupt calls that change the configuration of the video adapter, get information about the video solution, and access all of the functions needed to work with text. All applications that know about the VGA standard can use these interrupt calls to perform the video BIOS functions. This is great for text, but there is almost nothing in the VGA BIOS that helps with drawing objects in graphics modes. Since graphical user interfaces are now becoming very popular, graphics support is very important.

Display Driver. The display driver fills the graphics support gap. A display driver is responsible for providing the means to translate application graphics commands to hardware commands and simulating capabilities not directly provided in the hardware. Each display driver is tailored to a specific application. Every application designer decides on the graphics commands needed and how they will be implemented. In the same way, each video chip maker chooses the graphics commands to implement in hardware. The application must supply a driver for every different video adapter it must run on (or the hardware manufacturer must supply a driver for every application it wants to support). This allows video manufacturers to produce software that allows specific applications to run at peak performance on their hardware.

The combination of display drivers, BIOS, and hardware provides the excellent video performance seen with the HP Ultra VGA video solution. The HP Ultra VGA board not only supports all of the modes that VGA contains, but also has a set of enhanced high-resolution modes that use a graphics engine to accelerate the most commonly used graphics functions. The enhanced modes are not standard VGA modes, but applications can get access to them via the display drivers. The drivers increase application performance by taking a high-level graphics operation like rectangle fill and performing the operation as fast as the video hardware can do it.

Applications send the display driver all graphics-related operations and the driver decides the best way to perform those operations. For example, to switch from mode three (text) to mode 201 (enhanced graphics) an application will send the driver the command to make the mode change. The driver then has to decide whether to make the mode change itself or send the command to the video BIOS. Typically, the driver will call the video BIOS for commands like mode changes. However, for commands like line drawing, the driver will usually communicate directly with the hardware to draw the line.

The implementation of display drivers is described later in this article.

Graphics Engine. The graphics engine is a state machine inside the video ASIC on the HP Ultra VGA board (see Fig. 1). Its purpose is to perform the high-level graphics functions in hardware so that the CPU is free to do other tasks. An eight-word FIFO buffer is provided so that the CPU can send all of the commands needed for at least one operation (the number of commands varies between operations). The FIFO reduces the amount of time the CPU has to wait when the graphics engine is still performing its last command.

The high-level graphics functions supported by the graphics engine include rectangle fills, line drawing, short stroke vectors, hardware cursor, bit-block transfers (BitBlt), and image transfers. Rectangle fill is the ability to fill a rectangular area of video RAM with some specific color and at the same time change what is already at that specified location in video RAM. For example, filling an area with all ones and the XOR operation will invert all colors (pixels) within the rectangle specified.

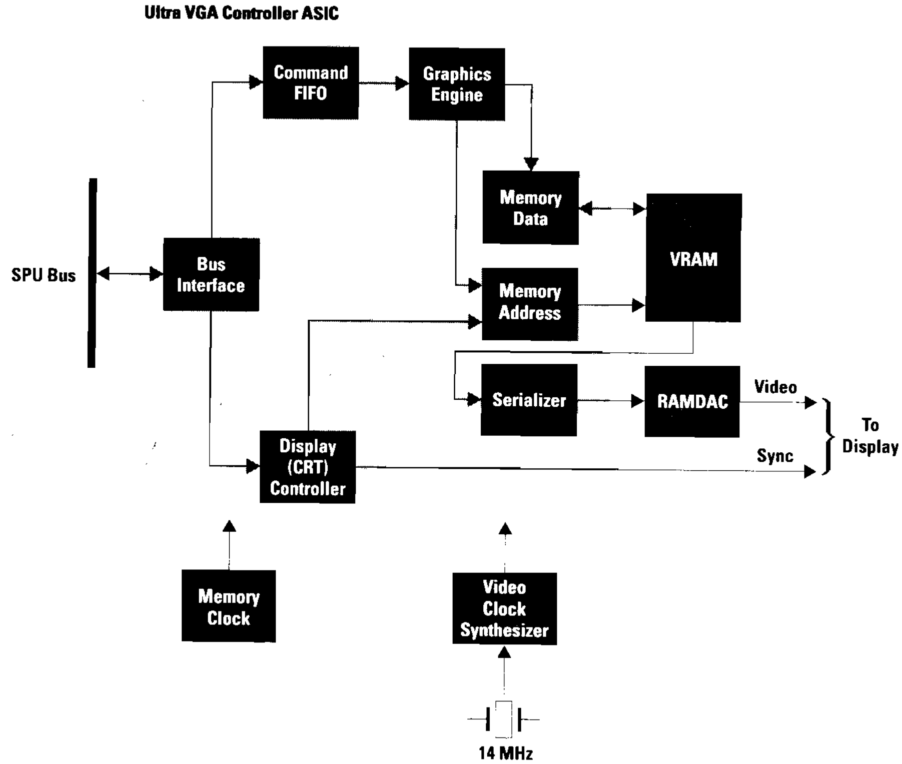


Fig. 1. A simplified block diagram of the Ultra VGA board.

Line drawing is the ability to draw a line between any two points. Short stroke vectors are a special case of line drawing in which short lines of sixteen dots or less are drawn at any forty-five degree angle (see Fig. 2). Each short stroke vector takes only one byte to specify, and the graphics engine can accept two bytes at a time. This is very handy for shapes that use many short lines, like characters, and since only one byte is sent per line, this operation is very fast.

Bit block and image transfers are for moving rectangular images around in video memory. A BitBlt is the fast transfer of a rectangular image from one location in video memory to another. Since the CPU only has to specify the source and destination coordinates and rectangle size, it can do other operations while the graphics engine does all of the work. The BitBlt operation is performed by the CPU first writing the source, destination, and rectangle size to the FIFO buffer if the FIFO has enough empty entries (the CPU requires four empty entries for this operation). Then the CPU writes a command word to the FIFO that tells the graphics engine what operation to perform. As soon as the command word propagates through the FIFO and the graphics engine receives it, the operation begins immediately. The graphics engine will read pixels from the memory within the source rectangle and the corresponding pixels from the destination rectangle. The graphics engine mixes the two sets of data in any of 256 different combinations and then writes the resulting pixels to the destination rectangle's video memory. This will continue until all the pixels in the destination rectangle are written. As soon as the graphics engine has completed this command, it will go back to the FIFO for the next command.

Image transfer performs the same function as a BitBlt except that it transfers an image from the CPU memory to video memory and vice versa. For this operation, the CPU only has to compute the number of bytes to be transferred and then write (or read) that number of bytes to or from a dedicated I/O register.

Since the graphic functions named above are constantly used with any graphical user interface, a large performance gain will be seen on any benchmark that tests video using these graphic operations. Since video is often the performance bottleneck, application-level benchmarks tend to improve as well.

After the CPU offloads operations to the graphics engine, it checks the FIFO to make sure that there is enough room before sending the next command. This means that the CPU might still have to wait (when the FIFO if full) if all it is doing is sending high-level video commands. For benchmarks that do a large number of one type of operation, the results may not indicate real performance. Because of this fact, benchmarks today are moving towards using real applications performing normal operations that can be completely automated.

Today the most useful benchmarks are applications that use a graphical user interface. Because of this, many PC benchmarks measure the performance of graphically intensive applications such as CAD programs. Two industry benchmarks, one running on Microsoft Windows and the other on AutoCAD, showed that HP Vectra 486/U machines using the HP Ultra VGA card significantly outperformed other PCs running the same benchmarks.

Host Bus versus the ISA Bus

Peripheral adapter cards are generally connected to a PC via the ISA (Industry Standard Architecture) bus, which runs at approximately 8 MHz (BCLK). BCLK is the ISA clock, which is obtained by dividing the CPU clock by either three or four depending on whether the processor clock is 25 MHz

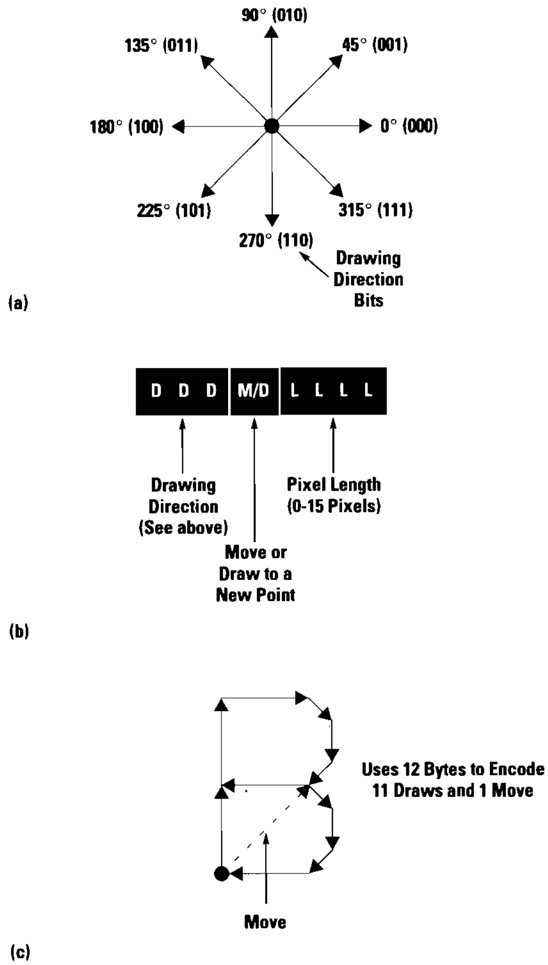


Fig. 2. (a) Short stroke vector directions. (b) Byte encoding of a short stroke vector. (c) Example of a character drawn using short stroke vectors. This character would require 48 bytes if stored as normal long vectors—12 X and 12 Y values each two bytes long.

or 33 MHz. One-byte accesses to an accessory card require a minimum of three BCLKs or approximately 375 ns. Two-byte accesses require a minimum of six BCLKs or approximately 750 ns. If the accessory board is not ready to start a cycle when it is accessed, or if the access takes longer than the minimum, additional wait states of one BCLK each are inserted. In Fig. 3 the signal BALE (bus address latch enable) signifies the start of a processor cycle and that the address on the bus is valid. Read/write (R/W)) and memory-I/O (M-I/O) are control signals indicating the direction and source or destination of the data on the bus. The read and write data signals indicate when the data must be stable for either type of transfer.

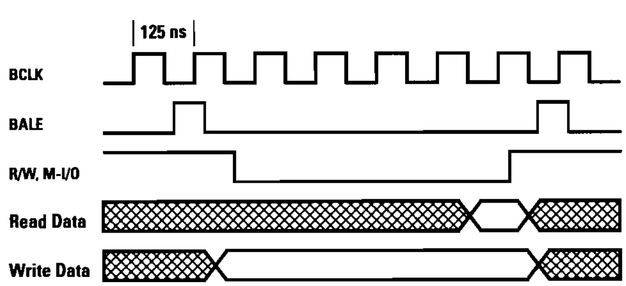


Fig. 3. A typical six-BCLK ISA cycle.

The HP Ultra VGA board is implemented as a 16-bit ISA accessory board. Because of this, its performance is limited by the bus bandwidth. For example, transferring a word from one location to another takes a minimum of six BCLKs (750 ns) even though the memory is capable of 80-ns access time. The HP Vectra 486 and 486/33T computers have an EISA bus, but the additional cost of implementing the Ultra VGA board as an EISA peripheral was not justified since it could not take advantage of the advanced features of the EISA bus such as bus mastering. Also, being an ISA board allows it to be used in other HP computers such as the Vectra RS 25/C, which doesn't have an EISA bus.

In the Vectra 486/U, the video subsystem is not a plug-in accessory board but is embedded on the processor PC board. It is connected directly to the host bus and thus can take advantage of the 32-bit bus width and the fast clock speed. The chipset used in the HP Vectra 486/U allows for four separate buses: the local bus, the host bus, the EISA/ISA bus, and the peripheral bus (see Fig. 4). The Intel486 processor and the secondary cache memory directly interface with the local bus. This bus architecture is unique to the Vectra 486/U. In most Intel486 designs, what we refer to as the host bus is the local bus and the cache shares bus bandwidth with other elements of the system. The HP 486/U gains performance by separating the local bus from the host bus since most high-speed critical operations are processor accesses to the cache. This also allows simultaneous access to main memory or mass storage by intelligent peripherals without interfering with the CPU.

The host bus is a 32-bit bus, operating at the Intel486 clock speed, which connects the main subsystems of the processor. As shown in Fig. 4, these subsystems include the memory controller, the EISA/ISA bus controller, the peripheral controller, and the video controller. The EISA/ISA bus is the backplane bus used for plug-in accessory cards. The peripheral bus connects many of the onboard subsystems in an ISA style protocol.

Devices on the host bus receive the signal HADS (host address data strobe) to begin a cycle (see Fig. 5). This causes an address decode to take place in the device. If the device recognizes the address as its own, it responds by asserting the signal HLAC (host local access). This causes other devices on the host bus to remain quiescent for the duration of the bus cycle. If no device responds, the address is propagated onto the EISA/ISA and peripheral buses. When the responding host bus device completes its operation, it responds by asserting the signal HRDY (host ready) which ends the cycle.

By comparing the timing diagrams shown in Figs. 3 and 5, it can be seen that the host bus implementation can speed up individual I/O or memory cycles by a factor of four to six. The decrease in CPU-to-peripheral transfer time and the accelerator built into the chip (described below) contribute to the superior performance of the embedded Ultra VGA subsystem.

VRAM versus DRAM

Most PC video adapters use standard dynamic RAM (DRAM) for display memory. While this is a cost-effective solution, it leads to performance penalties. Because a DRAM has a single data port, accesses from the CPU and the display controller must be time-multiplexed. The display controller must have

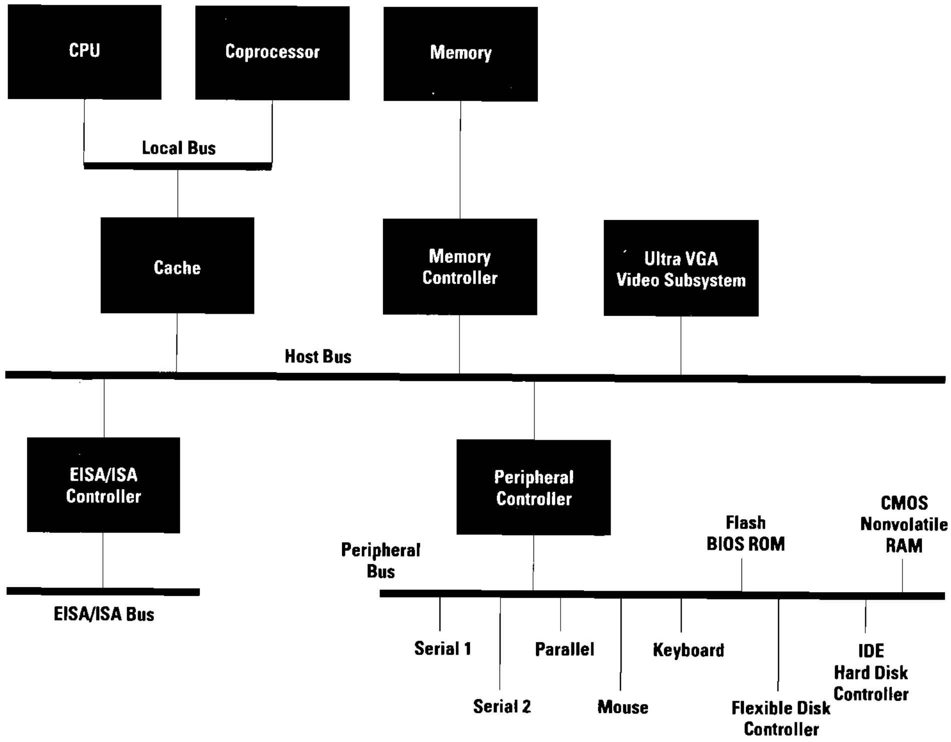


Fig. 4. Basic block diagram of the HP Vectra 486/U.

a high priority because any missing data would show up as noise or snow on the display. In the original CGA, the only time the CPU is allowed to access display memory is during the retrace intervals at the end of each scan line. This means that that the CPU can get only two or three clean accesses every 63 μs (see Fig. 6a).

The EGA and VGA architectures also use DRAM, but by using four-bit wide chips, the CPU/display interleave is brought down to 1:2 (see Fig. 6b). This allows a CPU access every 450 ns. This still necessitates slowing the CPU down, since if the access just misses, the processor has to wait 450 ns until the next access window opens.

Another RAM architecture made especially for video use is the video RAM (VRAM). VRAM is a dual-ported device that allows the CPU almost unlimited access to the display memory while still maintaining a noise-free display. Fig. 7 shows a simplified drawing of a 256-bit VRAM. The RAM array in this case is 16 rows of cells by 16 columns. In an actual VRAM the array would be 64K bytes in a 256-by-256 array.

In normal DRAM accesses, a row is selected by the row address and read in to the sense amplifiers. The column address is used to select one of the column sense amplifiers to read or write a single bit of the array.

In a VRAM, access to a row is also selected by the row address. However, instead of only one column being selected, all of the columns are simultaneously read into a serial shift register. The data is then shifted out of the shift register as it is needed by the display. In this way the display controller need only lock out the CPU for one cycle out of every 256 (or less depending on the width of the VRAMs) to present a clean display.

The Ultra VGA board uses the VRAM mode when it is operating in its enhanced mode. In the standard VGA mode of

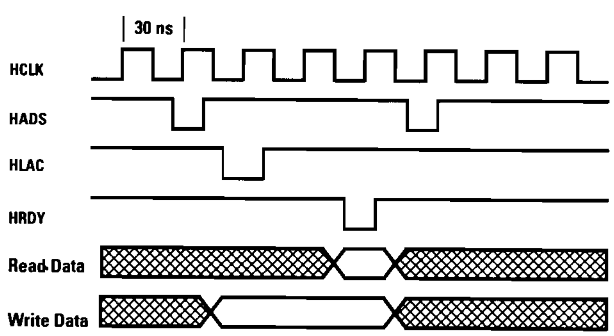


Fig. 5. Typical host bus cycle.

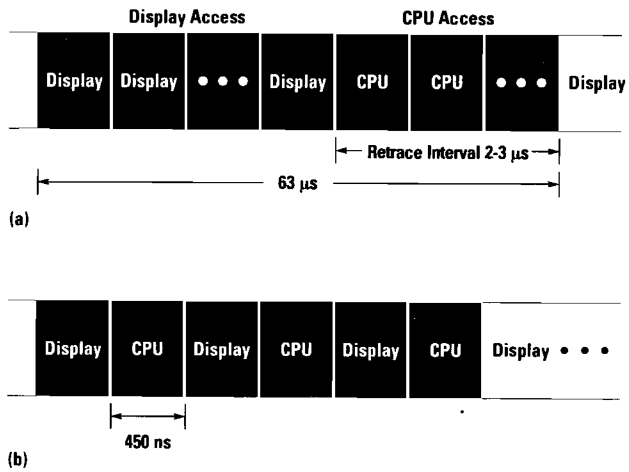


Fig. 6. CPU/display memory accesses. (a) In the original CGA implementation the CPU gets access only during retrace intervals. (b) In the EGA and VGA architectures the CPU gets access to memory one out of every two memory cycles.

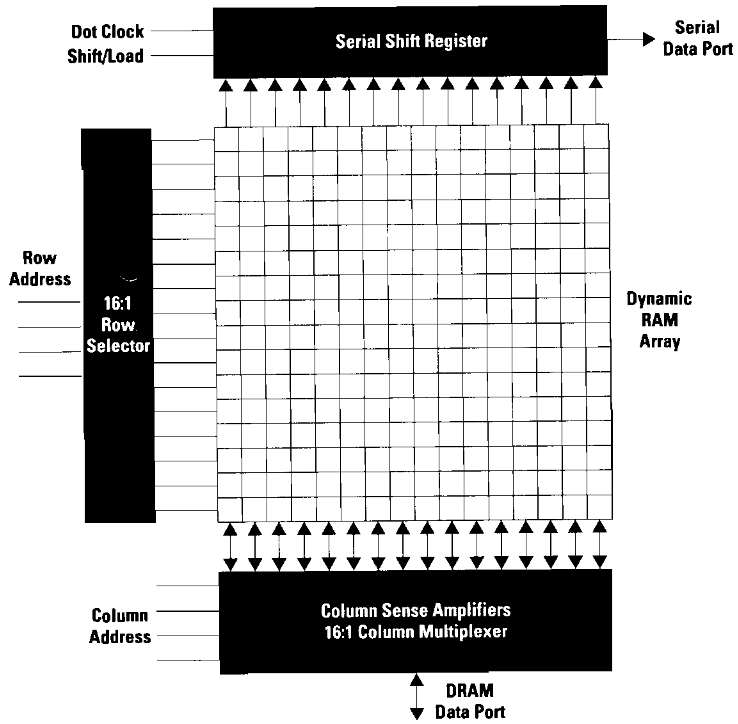


Fig. 7. Simplified diagram of a VRAM.

operation the Ultra VGA board accesses video memory as if it were DRAM.

Clock Synthesizer

Because of the many different display resolutions and monitor characteristics associated with the Ultra VGA board, up to 16 different video dot clock frequencies are needed. The board space needed would be prohibitive if these clocks were generated with discrete crystals or oscillators. Instead we use a clock synthesizer IC. This relatively new chip combines analog and digital circuitry on the same chip. It contains an oscillator, a phase-locked loop, and digital dividers that drive the phase-locked loop. Except for the reference frequency crystal (14.31818 MHz) and an RC filter, all of the necessary components are contained on the chip. This gives enormous capability in very little board space.

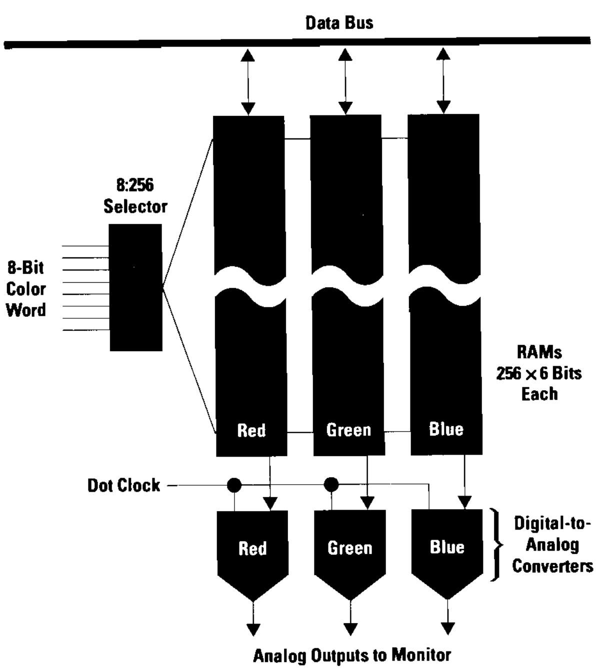


Fig. 8. Simplified diagram of a RAMDAC.

RAMDACs

The CGA runs its monitor with four digital signal lines: red, green, blue, and intensity. This allows a maximum of 16 colors to be displayed. In graphics modes the colors are fixed by the hardware and selected from two palettes of four colors each.

The EGA extends this by providing six digital signals: red, red′, green, green′, blue, and blue′. This allows a maximum of 64 different colors to be displayed. The digital-to-analog converters (DACs) are built into the monitor and the 64 shades are fixed by the manufacturer. The EGA board has a palette consisting of 16 six-bit entries, and each palette entry can be programmed to select one out of 64 shades.

The VGA doesn't drive the monitor with digital signals, but uses analog signals instead, one for each primary color (red, green, and blue). By varying these signals, an almost infinite range of colors can be displayed. The standard VGA uses a RAMDAC with 256 eighteen-bit entries. Each of the entries has six bits each for red, blue, and green (see Fig. 8). The maximum number of colors that can be generated is 2^{18} or 262,144. Of these, any 256 can be displayed simultaneously.

Ergonomics in PC Graphics

Higher Resolutions and Higher Refresh Rates

Since the establishment of the IBM VGA as a PC graphics standard, there has been steady progress in the development of higher screen resolutions. The IBM VGA offers a maximum resolution of 640 by 480 pixels with 16 colors. Recent super-VGA boards from various manufacturers support higher resolutions of 800 by 600 and 1024 by 768 pixels, along with 256 colors.

The most direct benefit of higher screen resolution is a larger display area for the user. This translates to advantages such as the ability to display more rows and columns of a spreadsheet, or larger sections of a word processor document.

The display refresh rate has also been steadily improved to address the problem of screen flicker. Flicker is perceived by the user as a direct result of the monitor screen not being refreshed at an adequate rate. Since all PC monitors are based on cathode ray tube (CRT) technology, the contents of the screen are not static but are constantly being swept onto the screen phosphor on a line-by-line basis. If the graphics system does not support an adequate screen refresh rate, pixel intensity will have time to decrease between successive refresh cycles, resulting in the perception of rapid screen flashing, or flicker. Viewing a monitor screen with significant flicker, especially for long periods of time, can result in eyestrain and other health hazards. The recent improvement in screen refresh rates has been largely successful in reducing the problems associated with screen flicker.

The standard VGA implements a screen refresh rate of 70 Hz for all text and graphics modes, except for 640 by 480 graphics modes, which offer a 60-Hz rate. The Video Electronics Standards Association (VESA) provides standards for refresh rates at higher resolutions including 72 Hz for 800 by 600 resolution and 70 Hz for 1024 by 768 resolution.

Monitors

Increasing the screen resolution or the refresh rate will directly increase the graphics output horizontal scan rate (Hsync), a measure of the time between successive horizontal display lines on the screen. The standard VGA uses a fixed Hsync rate of 31.5 kHz for all text and graphics modes. Combinations of higher resolutions and higher refresh rates can yield an Hsync rate ranging from 31.5 kHz to beyond 64 kHz.

All standard VGA-only analog monitors on the market can support only the standard 31.5-kHz Hsync rate. To properly support modes with higher Hsync rates, dual-sync or multisync monitors are required. Dual-sync monitors, such as the HP Super VGA Display (HP D1194A) and the HP Ergonomic Super VGA Display (HP D1195A), can support Hsync rates other than 31.5 KHz. The capabilities of these two monitors and others are listed in Fig. 9.

Multisync monitors are typically capable of synchronizing to a continuous range of Hsync frequencies, allowing them to support standard VGA modes as well as higher resolutions. The HP Ultra VGA Display (HP D1193A) is an example of a multisync monitor.

The HPUVGA Utility

With the choice of multiple refresh rates and monitors with different resolutions, the user needs to configure the graphics system to select the correct refresh rates for the resolutions supported by a given monitor. The HP Ultra VGA board is shipped with a configuration utility called HPUVGA.EXE, which allows the user to customize the Ultra VGA board output to any HP PC graphics monitor.

Embedded within the HPUVGA utility is information pertaining to the synchronization capabilities of all of HP's PC graphics monitors. By correctly selecting the monitor in use, the user is able to view the refresh rates supported by the monitor at graphical resolutions of 640 by 480, 800 by 600, and 1024 by 768 pixels. In cases in which the monitor can support two or more refresh rates for a given resolution, the user is given a choice. All refresh settings are saved in an HP CMOS video byte, which is described later.

Horizontal Resolution	Vertical Resolution	Colors	Mode Type	Memory Required	Hsync (kHz)	Vsync (Hz)	D1192A	D1187A D1193A	D1194A	D1195A
32 columns	25 rows	16	Text	512K Bytes	31.5	70	✓	✓	✓	✓
640	480	256	Graphics	512K Bytes	31.5	60	64 Shades	✓	✓	✓
640	480	256	Graphics	512K Bytes	37.9	72		✓	✓	
800	600	16	Graphics	512K Bytes	37.9	60		✓	✓	
800	600	16	Graphics	512K Bytes	48.1	72		✓		✓
800	600	256	Graphics	1M Byte	37.9	60		✓	✓	
800	600	256	Graphics	1M Byte	48.1	72		✓		✓
1024	768	16	Graphics	512K Bytes	48.4	60		✓		
1024	768	16	Graphics	512K Bytes	56.5	70		✓		
1024	768	256	Graphics	1M Byte	48.4	60		✓		
1024	768	256	Graphics	1M Byte	56.5	70		✓		

D1192A—HP Monochrome Display

D1187A—HP 20-Inch High-Resolution Display

D1193A—HP Ultra VGA 17-Inch Display

D1194A—HP Super VGA Display

D1195A—HP Ergonomic Super VGA Display

Fig. 9. Summary of HP PC monitor capabilities. In addition to the capabilities listed, all of the monitors provide standard VGA modes.

The HPUVGA utility also supports emulation modes for the MDA, HGC, and CGA PC graphics standards. Two additional 132-column text modes, with 25 and 43 rows respectively, can also be set via the HPUVGA utility.

HP CMOS Video Byte

Refresh rate settings for graphics resolutions of 640 by 480, 800 by 600, and 1024 by 768 pixels are saved in the HP CMOS video byte. The assignments for each bit in this byte are:

Bit 7: Alternate I/O port select
Bit 6: Reserved
Bits 4 and 5: 1024 by 768 refresh timing
Bits 2 and 3: 800 by 600 refresh timing
Bits 0 and 1: 640 by 480 refresh timing

The monitor timings for all supported video modes are stored in table format in the video BIOS, with one table entry per video mode. When an application calls the Int 10h set-mode function of the video BIOS to enter a specific accelerated graphics mode, the video BIOS accesses the HP CMOS video byte to determine the refresh rate currently selected, then uses the corresponding timing table to get the correct refresh rate. This scheme allows the refresh rate control to be application independent.

Since HP CMOS memory is a nonvolatile system resource, the refresh rate settings are preserved in the same way as other standard system configuration information. This scheme is capable of supporting operating systems besides MS-DOS®. Alternatives to HP CMOS memory for saving refresh rate settings have been carefully considered. Adding EEPROM hardware to the HP Ultra VGA board to store the refresh rates had the disadvantages of high cost and increased design complexity. Using a TSR program (memory resident software) to preserve the refresh rates would have worked only for MS-DOS, and other systems such as the OS/2 and UNIX* operating systems would also require specific memory resident software. Memory resident software would occupy valuable system memory and reduce ease of use.

Display Drivers

A display driver is a distinct program module that is made up of a group of display functions that provide a standard interface between an application and a particular type of video display hardware.

The HP Ultra VGA accessory card provides many features, such as hardware line drawing, bit-block image transfer (BitBlt), rectangle fill, and hardware clipping. However, these features can only be accessed through some special video enhanced modes which are unique to the graphics processor in the HP Ultra VGA card. In most cases, application programs, such as Microsoft Windows and AutoCAD, do not know (and do not want to know) how to enter these enhanced modes. It is the manufacturer-specific display driver that lets the application program take full advantage of the graphics processor.

For example, to make the HP Ultra VGA card work in 256-color enhanced mode with 1024-by-768-pixel resolution, a display driver has to call the BIOS interrupt 10h with registers AX=0x4F02 and BX=0x205. In general, to set the display in one of the HP Ultra VGA enhanced video modes, the driver calls BIOS interrupt 10h with the AX and BX registers set to values that represent different resolutions and colors.

To access hardware line drawing, BitBlt, and rectangle fill features of the HP Ultra VGA hardware, the display driver sets the drawing command register at I/O address 9AE8h. Fig. 10 shows a definition of each bit in this register.

For example, when an application wants to draw a line on the screen, the display driver sets the following bits in the drawing command register at I/O address 9AE8h:

Bits	Setting	Meaning
13-15	001	Draw Line Command
04	1	Draw = Yes
00	1	Write

The driver also has to find out the drawing direction to fill in bits 5 to 7.

Another important feature of the HP Ultra VGA card is the short stroke vector drawing ability. Using short vectors for displaying text in the graphics mode improves video performance. When an application program requests to display text on a high-resolution graphics screen, the display driver sets the short stroke vector command register at the I/O address 9EE8h. Fig. 2b shows the bit definitions for the short stroke vector register.

Typically, an application program uses a standard interface to the display so it doesn't have to be concerned with the type of hardware installed on the machine in which it is running. This isolates the application program from the display hardware. For example, most Windows applications are written without regard to the type of video adapter used. Instead, the programs are written to interface with Microsoft Windows.

Bit	Meaning
15, 14, 13	Command Type 001: Draw Line 010: Rectangle Fill 110: BitBlt
12	Byte Swap (1 = Yes 0 = No)
11	0
10	(Reserved)
09	Bus Size (1 = 16 bit 0 = 8 bit)
08	Wait (1 = Yes 0 = No)
07, 06, 05	Drawing Direction in Degrees 000: 0-45 001: 45-90 010: 90-135 011: 135-180 100: 180-225 101: 225-270 110: 270-315 111: 315-0
04	Draw (1 = Yes 0 = No)
03	Direction Type (1 = Radial 0 = x-y)
02	Last Pixel (1 = Off 0 = On)
01	Pixel Mode (1 = Multiple 0 = Single)
00	Read/Write (1 = W 0 = R)

Fig. 10. Definitions of each bit in the drawing command register.

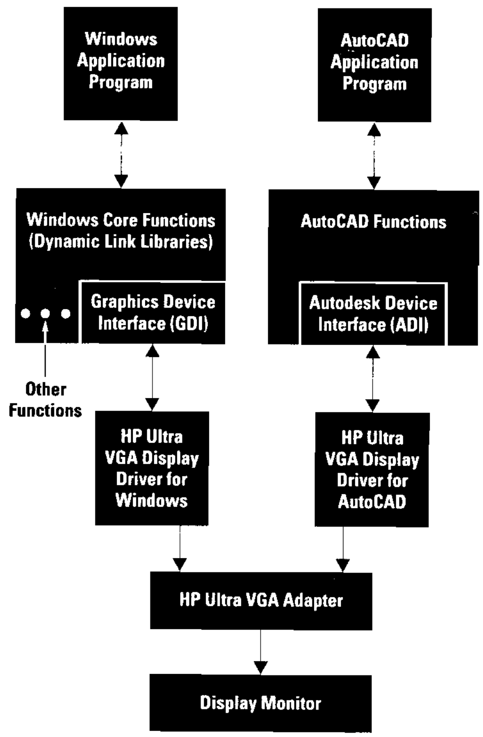


Fig. 11. Software hierarchy from the application to the display driver.

The video adapter's display driver takes care of writing to the display hardware. The Windows display driver works with any Windows program. By going through a standard interface, the display driver developer and the application program developer are isolated from each other (see Fig. 11).

The Windows Display Driver

The display driver for Windows is a dynamic link library that consists of a set of graphics functions for the HP Ultra VGA display card. These functions translate device independent graphics commands from the Windows graphical device interface (GDI) into the commands (such as the drawing command described above) and actions the Ultra VGA graphics engine needs to draw graphics on the screen. These functions also give information to Windows and Windows applications about color resolution, screen size and resolution, graphics capabilities, and other advanced features, such as BitBlt, line-drawing, polygon fill, and hardware cursor support. Applications use this information to create the desired screen output.

The HP Ultra VGA Windows display driver is based on the sample driver for the IBM 8514 graphics adapter. The source code for the 8514 driver is available from Microsoft's Driver Development Kit. Like most Windows display drivers, the Ultra VGA driver provides the following basic functions:

- Output. Draws various shapes.
- Enable. Starts or resumes display activity.
- Disable. Stops display activity.
- RealizeObject. Creates physical objects (e.g., pens, brushes, and device fonts) for exclusive use by the display driver. This is where the translation between device independent (or logical) and device-optimal (or physical) objects takes place.
- ColorInfo. Translates between logical colors, which are passed by Windows as double word RGB values, and physical colors recognized and used by the Ultra VGA display drivers.
- BitBlt. Supports bit-block transfers by copying a rectangular block of bits from bitmap to bitmap while applying some specified logical operations to the source and destination bits. A bitmap is a matrix of memory bits that defines the color and pattern of a corresponding matrix of pixels on the device's display screen. Bitmaps provide the ability to prepare an image in memory and then quickly copy it to the display.
- ExtTextOut. Draws a string of characters at a specified location on the screen and clips any portion of a character that extends beyond a bounding box of the string.
- StrBlt. Supports text drawing for the earlier versions of Windows. (It just makes a call to ExtTextOut.)
- Control. Passes special control information to or receives special information from the Ultra VGA display driver.

Besides the functions listed above, the following features have been added to take full advantage of the graphics engine in the Ultra VGA.

- Different Resolutions. Separate display drivers are provided to support resolutions of 1024 by 768, 800 by 600, and 640 by 480 pixels with 256 colors.
- Hardware Cursor. An onboard hardware cursor (64 by 64 pixels) for fast cursor movement in the enhanced mode.
- Fast Polyline Draw. Onboard hardware is used to draw solid polylines at a very fast speed.
- Polygonal Capabilities. An onboard drawing command register and hardware are used for quick rectangle fill and scanline drawing.
- Fast Rectangular Clipping. Rectangular clipping is provided via a clipping window boundary register and hardware that discards points that are outside of a specified rectangle or region drawn on the screen.
- High-Speed BitBlt. Onboard hardware is used for high-performance bitmap image transfer operations.
- FastBorder Function. A function that draws borders for windows and dialog boxes very quickly.
- Save Screen Bitmap. The SaveScreenBitmap function allows Windows to save bitmaps temporarily in offscreen video memory. This function speeds the drawing operations that require restoring a portion of the screen that was previously overwritten.
- Support for Large Fonts. Support for large fonts is provided in which the font and glyph information can exceed 64K bytes.
- DIBs Support. This function converts device independent bitmaps (DIBs) to physical format for direct transfer to the display without applying a raster operation. Note that a DIB is a color bitmap in a format that eliminates the problems that occur when transferring device dependent bitmaps to devices having difference bitmap formats.
- Support Device Bitmap. A device bitmap is any bitmap whose bitmap bits are stored in device memory (such as RAM on a display adapter) instead of main memory. Device bitmaps can significantly increase the performance of a graphics driver and free system memory for other uses.

- Font Caching. Font caching is temporarily saving the most recently used fonts in offscreen video memory. This function speeds up text-redisplaying operations.
- Small and Large Fonts. The Ultra VGA display driver provides both small and large fonts in the 1024-by-768 high-resolution mode.
- Vector Fonts. The Ultra VGA display driver supports vector fonts. A vector font is a set of glyph definitions, each containing a sequence of points representing the start and end points of the line segments that define the appearance of a character in a particular font.

Working Together

The functions in the Ultra VGA display driver and the Windows graphical device interface (GDI) work together to make efficient use of the features provided in the HP Ultra VGA board. The rest of this section describes how these two entities work together to initialize the display and perform some simple graphical operations.

When the user types WIN to start Windows, a small program WIN.COM determines the mode in which Windows is to run. If it determines that it can run in the enhanced mode, Windows runs KRNL386.EXE (via WIN386.EXE). While initializing, Windows checks the DISPLAY.DRV setting in the SYSTEM.INI file to determine the file name of the display driver to load. The HPUxxx.DRV driver modules are the display drivers for different resolutions and video memory configurations. The Windows graphical device interface (GDI) then calls the selected display driver's initialization routine.

During initialization, the Windows GDI makes two calls to the Enable function in the Ultra VGA display driver. After the first call, the Enable function then returns to the GDI the GDIINFO data structure, which describes the general physical characteristics and capabilities of the HP Ultra VGA graphics engine. The GDI uses this information to determine what the Ultra VGA display driver can do and what the GDI must simulate.

The second time the GDI makes a call to the Enable function, the display driver does three things. First, it initializes the Ultra VGA graphics engine to be ready to run Windows. This includes saving the current mode, using the video BIOS function 10h calls to set the enhanced display mode and colors, load the palette, and so on. Next, the Enable function calls the hook_int_2Fh function in the display driver so that each call to interrupt 2Fh will be checked to detect any screen Switch function calls. This is because in a preemptive multitasking environment such as 386-enhanced-mode Windows, the display driver has to save and restore the display hardware settings, such as video mode and register data, whenever Windows is changed between a Windows application and a non-Windows application.

The last thing the Enable function does is to initialize and copy the PDEVICE data structure. The PDEVICE data structure defines physical objects rather than bitmaps. Physical objects define the attributes (such as color, width, and style) of lines, patterns, and characters drawn by the display driver. Physical objects correspond to the logical pens (used to draw polylines and borders around objects drawn by the Output function), brushes (used to fill figures drawn by the Output function and to fill rectangular areas created by the BitBlt function), and fonts (used by the ExtTextOut function to draw text). Physical objects also contain Ultra VGA hardware device dependent information that a display driver needs to generate output. These physical objects are created by the RealizeObject function.

After the RealizeObject function is finished creating the default pens and brushes for Windows and the brushes needed to draw the desktop and fill the Program Manager window, the BitBlt and ExtTextOut functions are called to do all the drawing on the screen. First, the BitBlt function draws a rectangle on the screen with the background color by using a pattern copy operation. Next, the BitBlt function is called to draw some borders and rectangles. Finally, to complete display initialization all the icons, text, and pictures are put on the Windows screen by using functions such as BitBlt, ExtTextOut, and the brushes created by the RealizeObject function.

When a Windows application requests to draw a line on the screen, the GDI checks the dpLines entry in the GDIINFO structure, which was filled by the display driver during initialization, to see if the display driver supports line drawing. Since the Ultra VGA driver supports hardware line drawing, the GDI calls the Output function to draw a line on the screen. Otherwise, the GDI has to simulate line drawing by combining scan lines and polylines.

If a Windows application asks to display text on the screen, the GDI calls the ExtTextOut function in the display driver. The ExtTextOut function receives a string of character values, a count of the characters in the string, a starting position, a physical font, and a DRAWMODE structure. These values are used to create the individual glyph images on the screen.

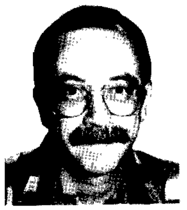
Finally, when a user asks to quit Windows, the GDI calls the Disable function in the display driver. This function frees any resources associated with the physical device and restores the Ultra VGA hardware to the state it was in before Windows started. After the display driver returns from the Disable function, the GDI frees the memory it allocated for the PDEVICE structure and frees the display driver, removing any driver code and data from memory.

Acknowledgments

The authors would like to thank the following individuals who participated in the development of the HP Ultra VGA products: Mark Linsky, Mark Hanlon, Jean-Claude Roy, Larry Durandette, Kevin Owen, Kiran Mundkur, Dave Wilkins, and Mike Milani. We would also like to thank all the members of the HP Vectra 486/U hardware, BIOS, and utilities teams for their assistance and cooperation.

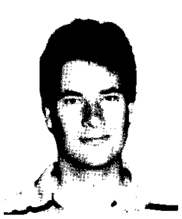
31 HP Ultra VGA Graphics Board

Myron R. Tuttle



A development engineer at the California PC Division, Myron Tuttle studied electrical engineering at the University of California at Berkeley (BSEE 1973 and MSEE 1974). With HP since 1974, he worked on the HP 2625/28 terminals and the original multimode video board for the HP Vectra. He contributed to the development of the video subsystem for the HP Ultra VGA board and is now involved in video and graphics development. Myron is named as the inventor for a patent on automated software testing and is coauthor of an earlier HP Journal article as well as a paper for an HP software conference. He also served in the U.S. Navy as an electronic technician. His hobbies include computer programming, home improvement projects, and classical music.

Kenneth M. Wilson



With HP's California PC Division since 1989, Ken Wilson has worked on a series of HP Vectra products, including the Vectra 486/25T and 33T, the Vectra 486s/20, and the HP Super VGA board. Most recently, he contributed to the development of the HP Ultra VGA board. He completed work for a BSEE degree from California State Polytechnic College at San Luis Obispo in 1988 and expects to receive his MSEE degree from Stanford University in 1993. His professional specialty is computer architecture, and when he takes a break from work, he enjoys boardsailing and relaxing in his hot tub.

Samuel H. Chau

R&D engineer Sam Chau was born in Hong Kong and attended the University of California at Berkeley. He received his BA degree in computer science in 1984 and came to HP's Santa Clara Division in 1985. Now at the California Personal Computer Division, he contributed to the development of the HP Super VGA board and worked on the hardware and display timings for the HP Ultra VGA board and the HP Vectra 486U embedded Ultra VGA+. Sam's outside interests include personal computers, audio and video technologies, photography, piano, classical music, and badminton.

Yong Deng

Born in Shanghai, China, software design engineer Yong Deng joined HP's California Personal Computer Division in 1989. He studied for his bachelor's degree in computer science at the University of California at Santa Cruz and graduated in 1986. In the past, he was responsible for software drivers and utilities for HP's intelligent graphics controllers. He developed a new display redraw method that improved CAD display list performance. He also ported Texas Instruments Graphics Language (TIGA) 2.05 and 2.20 to HP's intelligent graphics controllers. For the HP Ultra VGA graphics project, he was responsible for the AutoCAD and Windows high-resolution display drivers and video BIOS. His other professional experience includes software development at National Semiconductor and Autodesk Inc. His professional interests include high-resolution display drivers and application development for Windows and CAD tools. Yong is married and has a young daughter.

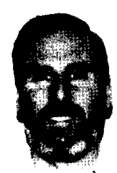
41 POSIX Interface for MPE/iX

Rajesh Lalwani

A software engineer at the Commercial Systems Division, Rajesh Lalwani joined HP in 1988. He was born in Mandsaur in the Madhya Pradesh state of India. He received a master of technology degree in computer science from the Indian Institute of Technology, New Delhi in 1986 and an MSCS degree from Pennsylvania State University in 1988. In the past, he enhanced and maintained command interpreter software and components of the MPE operating system kernel. More recently, he developed a procedure for parsing MPE and POSIX file names and a directory traversal routine. He's currently working on symbolic links functionality and device files for MPE/iX. Rajesh is the author of several POSIX articles, has presented a number of conference papers on the same topic, and is working on a book on POSIX.1. His outside activities include tennis, watching classic movies, and staying in touch with his wife, who is finishing her medical degree in India.

47 Preventing Software Hazards

Brian Connolly



Brian Connolly is a project manager for software quality engineering in HP's Patient Monitoring Systems Division and has been with the company since 1984. Previously, he developed real-time software systems for Raytheon Corporation and Westinghouse Corporation. Since joining HP, he has worked on real-time software development for a bedside monitoring application, object-oriented software development in a clinical information system, and software quality engineering for several bedside monitor and central reporting station products. He has written several papers related to hazard avoidance and software quality and testing for internal HP publication. He's also a member of the IEEE. His educational background include a BS degree in physics and engineering awarded by Loyola College in 1977, and an MES degree (1983) in digital systems, also from Loyola. Brian is married and has two children. His leisure activities include running, swimming, woodworking, and coaching youth soccer.

53 Configuration Management for Tests

Leonard T. Schroath

With HP since 1985, software quality engineer Len Schroath worked at the Logic Systems Division and the Colorado Springs Division before moving to his current position at the Boise Printer Division. He's the author or coauthor of six papers for HP conferences related to software quality, testing, and reuse. Len was born in Detroit, Michigan and attended Brigham Young University, from which he received a BS degree in computer science in 1985. He is married, has three small children, and is a coach at his local YMCA. He also enjoys music and sports and officiates at basketball games.

60 Software Inspections

Jean M. MacLeod

A native of Arlington, Massachusetts, Jean MacLeod studied elementary education and sociology at Emmanuel College in Boston, Massachusetts, and received a BA degree in 1971. She has worked in the software quality field since 1974, initially in several small to mid-sized companies before joining Apollo in 1987. At Apollo, she was responsible for initiating a software inspection program in an R&D lab. She joined HP's Corporate Engineering Software Initiative in 1991, and contributed to the improvement of software inspections for the Patient Care Monitoring Systems Division at Waltham. She's now working on software process improvement with other divisions in the Northeast. Jean is a member of the Society of Women Engineers. She has two teenage children and enjoys golf, racquetball, and reading.