SPARCstation 10SX Graphics Technology

A White Paper

Sun

Contents

Overview 1

Introduction

Over the last several decades, computer graphics has grown from specialized applications in niche markets such as mechanical design and mapping, to broad usage in areas ranging from electronic design and animation to graphics arts and multimedia. Accompanying the spread of computer graphics use have been the general availability of low-cost graphics subsystems, the proliferation of development tools and graphical user interfaces, and an explosion in the number of available applications.

These general trends have largely occurred through advances in graphics technology, such as high-quality rendering tools, sophisticated 2-D and 3-D accelerators, and the emergence of standards in application program interfaces. Coincident with these advances has also been the trend toward providing these features on low-cost desktop systems, leading to affordable solutions for increasingly broader groups of users.

Despite the continued reduction in the cost of desktop systems, graphics device architecture has not changed significantly over time. 2-D and 3-D graphics display devices, even with clear innovation and integration in design, are still based on the philosophy of accelerating limited predefined geometry primitives, such as vectors and polygons. Designing graphics products with such specialized applicability results in the delivery of performance and functionality to geometry-oriented applications only.

The image manipulation capabilities of current graphics accelerators are limited by their design, which focuses principally on the computation of geometric data. The complexity of such devices also requires that they reside on an I/O bus, which needs additional logic and has greater overhead in communicating with the CPU-memory bus. Thus, imaging applications that use general-purpose graphics processors are limited by such architectures, because they stress the manipulation of geometry over image data.

The SPARCstation™ 10SX™ represents an innovative approach to the requirements of imaging applications. Residing directly on the CPU-memory bus, the SPARCstation 10SX memory controller and rendering engine, called the SMC, can perform extremely fast transfer of data from physical memory (DRAM) to the frame buffer. This architecture serves to increase the image transfer rate by more that ten times the equivalent processor technology residing on an I/O bus.

The SMC also acts as a *vector integer coprocessor* working in a balanced fashion with the host processor. The SMC provides pixel and vector manipulation logic, which complements the floating-point computational capability of the CPU. The SMC's direct access to physical and display memory additionally provides flexible and powerful capabilities for the management, manipulation, and display of large image sets.

The SPARCstation 10SX system was designed to leverage significant advances in multiprocessor bus design, high-performance RISC technology, and graphics integration. The development of the SPARCstation 10SX system reflects a redefinition of integrated graphics, based on exhaustive analyses of rendering requirements and the advanced technology of today's multiprocessing workstations.

The SPARCstation 10SX graphics architecture employs a new and powerful approach to graphics integration. Never before has such integration with CPU and memory been achieved, with the significant results of flexible image manipulation and unsurpassed performance. By combining the SPARCstation 10 processor with the SMC, graphics rendering performance is no longer limited to the bus bandwidth of peripheral devices. The SPARCstation 10SX CPU-graphics integration further enables closely coupled execution of graphics commands without the necessity of maintaining separate graphics display and processor memory, thus increasing performance at lower cost.

This white paper describes the innovative design of the SPARCstation 10SX graphics architecture, and its optimized feature set for providing graphics acceleration to a broad range of application areas.

The Market for Image Manipulation

Among the key trends computing users face today are the move toward running multiple, varied applications and the growing demand for high-performance graphics. Multitasking operating systems, window-based graphical user interfaces, and powerful multimedia applications are just some of the factors influencing the trend toward more powerful graphics on the desktop.

Once the domain of high-end users in computer-aided design (CAD), technical imaging, color prepress and other markets, powerful graphics subsystems are increasingly of interest to volume users with broader requirements. To meet the needs of this large segment of users, specialized, moderately-priced accelerators have become available for personal computers and workstations. Often, these devices are optimized for focused acceleration of windowing, image manipulation, or 3-D CAD. However, it is uncommon for a graphics display processor to address all of these areas in an equal manner.

This lack of equitable support for multiple application areas is in distinct contrast to the requirements of users today, who require the same performance to run a multimedia or color publishing application as they do a 3-D graphics one. Balanced graphics performance, which includes image manipulation performance, is the key.

Figure 1.1 illustrates the range of graphics rendering operations typically associated with the key graphics market areas. Commercial imaging, for example, relies on support for image manipulation operations, such as halftoning, dithering, and color transformation. At the same time, multimedia applications are characterized by requirements for image compression and decompression, audio/video signal processing, and data format conversion. The other graphics market segments have similarly diverse requirements for specialized operations.

Figure 1.2 illustrates the capabilities of generic graphics accelerator products in the key market areas. For example, the optimized functions present in common frame buffers generally are limited to 2-D rendering, while 3-D accelerators provide broader coverage to include 2-D, 3-D, and high-quality rendering.

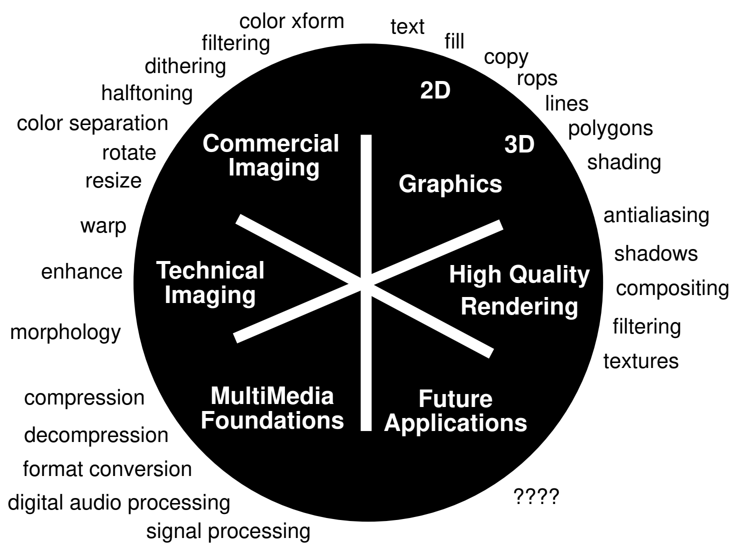

Figure 1.1 Display-oriented Application Space and Relevant Operations

As indicated in Figure 1.2, the acceleration of off-screen images for typical graphics accelerators is virtually nonexistent. Off-screen images are used in imaging applications and application programming interfaces (APIs) as intermediate storage for data prior to display. Efficient off-screen image support relies on the use of memory rasters, both by the graphics hardware and through software APIs.

The reason for the lack of memory raster support in common graphics devices is because almost all graphics accelerators are display-oriented. Thus, they are optimized for operations that directly render geometry, rather than provide facilities for the manipulation of images.

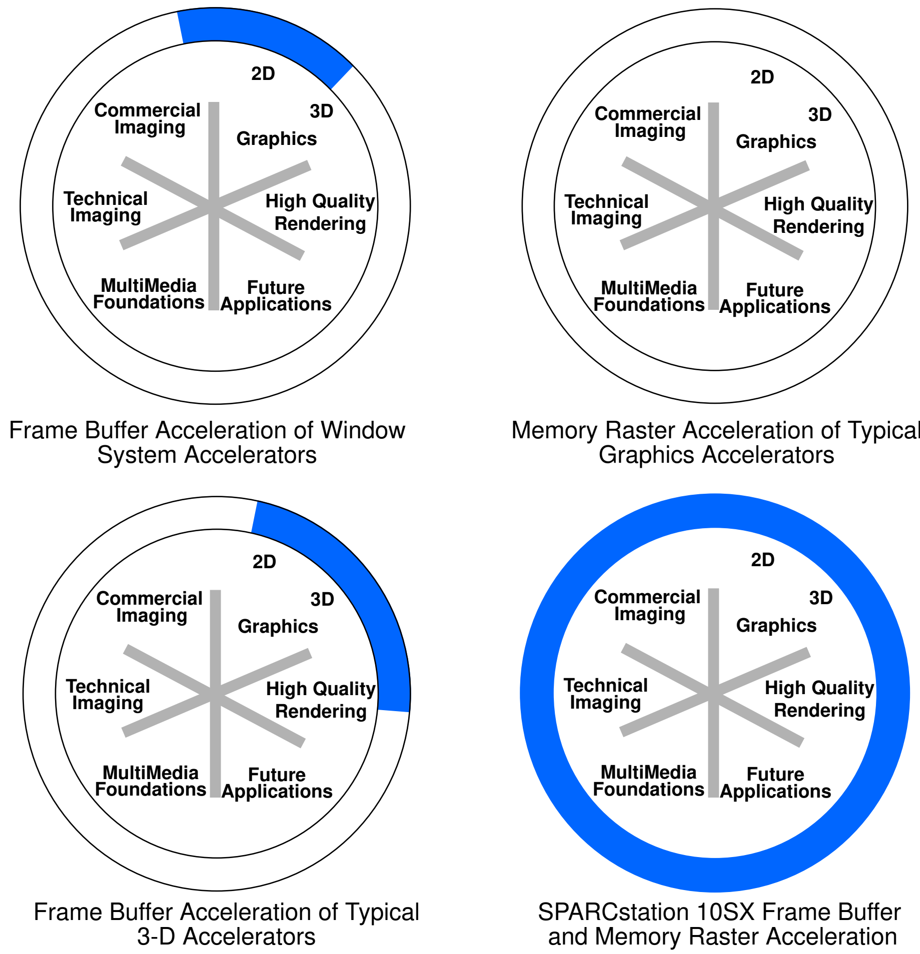

Figure 1.2 Acceleration Characteristics of Common Graphics Devices

To address the independent requirements of the major market segments, the SPARCstation 10SX graphics technology was designed to employ features that allow it to accelerate the low-level rendering operations common to all graphics accelerator devices. In addition, the unique coupling within the SPARCstation 10SX system between the SMC, memory controller and CPU enable direct access to system and display memory in such a way as to also provide acceleration for image manipulation.

The SPARCstation 10SX system includes broad-based acceleration of all current graphics rendering and manipulation operations, making it suitable as a volume platform for common graphics applications of all types. In addition, its unique features make it especially well suited for very specific markets. This is, in part, due to the SPARCstation 10SX system's high-performance imaging capabilities, particularly in areas where imaging requirements have historically been poorly addressed.

The sections that follow describe the key primary and secondary market segments, the characteristics of their applications, and how the SPARCstation 10SX architecture provides acceleration to them.

Principal Markets

Color Prepress

A typical customer in this field wants to scan in high-resolution color photographs and prepare them for printing on a color printer. The first step in this preparation might be to perform simple image processing operations on the images, such as sharpening, blurring, rotation, or cropping. Finally, the images must be converted to the CMYK color space, which provides the digital equivalent of photographic color separation. Morphological dilation of selected image features (known as "trapping") may be employed to avoid registration problems.

Because color prepress is concerned with guaranteeing image fidelity, it is important that the image data be digitally maintained at the highest possible resolution. The SPARCstation 10SX system's ability to efficiently manipulate large image rasters, coupled with its support for color space manipulation, make it ideal for color prepress requirements. The high level of integration between the SMC, CPU and display memory also insures that image processing operations can be performed quickly on multiple images.

Technical Imaging

Technical imaging applications include remote sensing, medical imaging, machine vision, and other areas where imaging is used to distinguish physical characteristics from digitized imagery. Technical imaging applications require broad base-level functionality for image manipulation, display, compression, and color. These applications frequently require advanced image processing capabilities, including transformations, filtering, and image enhancement.

The SPARCstation 10SX system is especially well suited for the wide range of image sizes, pixel depths, and compression characteristics common in technical imaging markets. The ability to manipulate and display sequences of images is extremely important to technical imaging users in areas such as medicine and remote sensing — and the SPARCstation 10SX system's high-speed access to display memory ensures that high frame rates can be achieved. In addition, the common image manipulation facilities (such as convolution, warping, logical operations, comparisons, and multiply) are directly accelerated by the SPARCstation 10SX system.

Technical Document Imaging

This market handles fewer images than the commercial market, although they are generally much larger. Potential applications in the technical document imaging market include textile design, cartography, facilities layout, construction, and architecture. Applications in this market need to process very large images, provide image-manipulation capabilities such as 'de-speckling' for removal of scanning artifacts, realignment to correct for inaccurate scanning registration, and warping to deal with the attributes of old hard-copy drawings printed on unstable materials.

The SPARCstation 10SX system was designed to accelerate the low-level image raster operations specifically characterized by the technical document imaging market. For example, warping, enhancement, and morphology operations are directly supported through individual and multiple optimized operations. The memory access speed between the SMC and the display memory further ensures that the copying and manipulation of multiple source images occurs very quickly.

Commercial Document Imaging

Banks and other financial institutions, airlines, and insurance companies process enormous numbers of paper documents during a typical day. Applications addressing this market must read digital versions of documents from scanning devices, compress the images into CCITT Group 3 or 4 facsimile format, index, and store them. Later, it may be necessary to retrieve the documents in a rapid manner for additional image processing, scaling, and enhancement

The SPARCstation 10SX system provides direct acceleration of the manipulation and processing of images with its advanced compression/decompression capabilities, data conversion, resizing (including zooming), and enhancement facilities. The SPARCstation 10SX system's ability to manipulate image raster data very quickly is especially important to this market segment, because of the extremely large volume of images that must be previewed and managed.

Geographic Information Systems (GIS)

In general, GIS applications vary broadly in data format, and manipulate large amounts of data. As a result, GIS applications depend equally upon both integer and floating point data manipulation and rendering.

The SPARCstation 10SX system is well suited for GIS applications for a variety of reasons. Copy speed from main memory to the screen, large in-memory support, and support for overlays (through an appropriate graphics API) provide capabilities important to GIS. Where data format or quantity is particularly important, the SPARCstation 10SX architecture enables graphics data to be processed very quickly, resulting in faster data fetching. The close coupling between the SMC and CPU ensures optimized floating point performance while also accelerating integer instructions. The 24-bit rendering capabilities of the SPARCstation 10SX system, together with its ability to manipulate very large rasters, also make it attractive to GIS.

Earth Resources

Like geographic information systems, the earth resources market typically deals with large volumes of data. Earth resources market requirements include 2-D and 3-D geometry data, and frequently, 2-D image data. With such varied requirements, earth resources applications need hardware with the flexibility to accelerate geometry with a range of coordinate types, and with the capability of manipulating images efficiently.

The SPARCstation 10SX system is an ideal platform for earth resources applications. Its ability to address video and display memory directly, as well as providing strong vector and triangle performance enables earth resources applications to manipulate and render large data sets easily.

Architectural Overview 2

SPARCstation 10SX Architecture

The SPARCstation 10SX graphics processor is a scalable graphics architecture consisting of an enhancement to the memory controller chip present on MBus-based SPARC™ systems. The memory controller chip is specifically enhanced with arithmetic processing capability to become an "intelligent" memory interface capable of optimized pixel operations applicable to graphics and imaging.

SPARCstation 10SX technology is based on the concept of graphics operations becoming extensions of the common integer and floating-point operations of the CPU. The SPARCstation 10SX system employs a closely-coupled architecture, wherein a dedicated processor handles low-level display operations through processor-memory direct access.

In addition to the usual buffering of data between the bus and memory interface and the generation of address and control signals, the SMC loads and stores memory data to and from its internal registers. It also performs arithmetic and other processing on that data. The processing consists of extensive pixel operations that support graphics and imaging functionality present in graphics and window system software APIs such as the XGL™ and XIL™ graphics libraries, and the OpenWindows™ windowing environment. The SMC is a slave processor to the SPARC CPU. It is designed to accept commands from the CPU, and has no independent instruction-fetch capabilities.

The SMC was designed as an extension to the memory controller chip for several reasons. The pixel processing speed of CPUs is, in general, a limiting factor for many application areas — thus making any significant acceleration an attractive proposition. Second, since a large part of the cost of graphics and imaging accelerators is in the need for dedicated memory components, an architecture that reduces memory requirements is desirable.

Finally, the memory interface on most systems typically has the highest bandwidth of any data path in the system. Its direct relationship to the CPU makes it an ideal candidate for providing a path for access to additional graphics functionality.

The SMC processor extends the functionality of the standard memory control found on MBus-based systems by incorporating logic for arithmetic and logical operations, vector (multiple) instruction support, multipurpose registers, and a variety of other high-level computational features. Table 2.1 describes some of the architectural elements found in the SMC.

Feature	Description
40 MHz Clock Speed	Synchronous operation with motherboard clock and asynchronous with the CPU
128 32-bit Registers	Multipurpose registers for intermediate SX calculations
Register Files with Dual Read Ports and Single Write Ports	Employed and configured to allow simultaneous read access of 8 registers and write access of up to 4 registers
Instruction Queue of 64 Levels	Used to buffer between 32 and 64 instructions, depending on type, sent from the host CPU
Control Registers on Separate Physical Page	Separate from noncritical resources allowing only supervisor access to critical resources
Error Checking and Flagging	For handling of exceptions should the SMC encounter a fault condition

Table 2.1 SMC Processor Architecture Features and Description

Feature	Description
MBus Interface Logic	Should the SMC be busy and unable to make a normal termination response, this logic issues requests to the CPU for relinquishing and retrying access
Four-Stage Pipeline	For execution instruction queuing
Vector Instruction Support	Allows for up to 16 iterations of an instruction (up to 32 with load and store)
Dual Arithmetic Logic Units	Allows individual operations within vector operations to be executed simultaneously
Dual Multiply Units	Allows two 16x16-bit multiply operations to be executed simultaneously
64-bit Interface with Memory Controller	Allows multiple vector elements within an instruction to be accessed simultaneously
Store and Load Instructions	Supporting direct and array addressing, 32 distinct data types, data clamping on stores, and masking of stored data at item and bit level
Boolean Logic Operation Support	Providing direct support for AND, OR, and XOR operations
Multiply Operation Support	Including multiply, dot product, and SAXPY, available in either 16x16 or 32x16 operands
32-bit Barrel Shifter Support	Supporting logical, arithmetic and funnel shifts
32-bit Signed Value Comparison Support	Supporting tests for >, <, =, >=, and <=
SELECT Instruction Support	For vector merge and 1-to-N expansion
Raster Operation Support	Through common ROP instructions
2-D Register Support	With GATHER and SCATTER functions
Incremental Addition Instruction	For use in Gouraud shading
Bresenham Interpolation Instruction	For line drawing

Table 2.1 SMC Processor Architecture Features and Description

SPARCstation 10SX High-Level Architecture

The SMC processor lies on the MBus between the SPARC CPU(s) and memory. In addition, Video RAM (VRAM) is mapped into the same address space, thus allowing the SMC to access and manipulate the display in the same manner as DRAM. This addressing gives the SPARCstation 10SX system superior bandwidth in moving images from memory to the screen. VRAM is packaged as a portion of a VSIMM, which also includes a Video Buffer Chip (VBC), a memory display interface (MDI), and Digital-to-Analog Converter (DAC) logic.

Instructions executing in one or more of the SPARC CPUs make requests of the memory controller (MC) portion of the processor. The memory controller logic fulfills CPU requests for memory accesses, such as those required for fetching, loading, and storing instructions.

When one of the CPUs encounters a graphics operation, it passes it to the SMC pixel processor (SXPP), which executes the instruction directly. When memory access is required to complete an operation, the SMC makes requests of the memory controller to process its requests.

Figure 2.1 illustrates the high-level architecture of the SPARCstation MBus system, including CPU, SMC processor, and video memory.

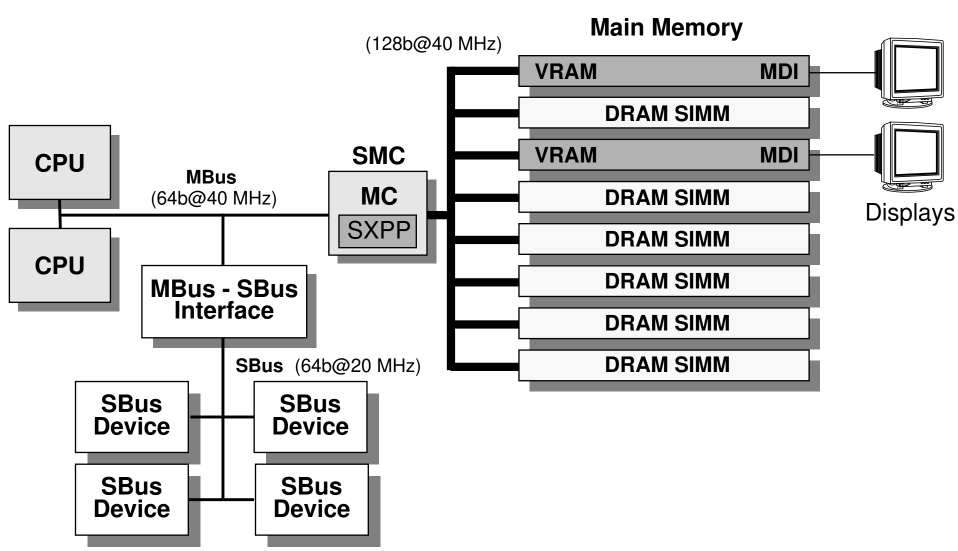

Figure 2.1 SPARCstation 10SX System High-level Architecture

SMC Low-Level Architecture

The SMC processor section executes the commands written to it by the SPARC CPU over the MBus, including loads and stores between the SMC internal registers and main memory, using the memory interface. When a normal memory access over the MBus occurs, the SMC pauses and lets it through. The SMC stall latency needs to be quite short in this case to avoid burdening normal memory traffic. This short latency is accomplished by implementing register loads and stores that act a great deal like MBus memory block accesses (e.g., bursts of 4 or 8 64-bit words). The blocking effect is ameliorated by the asynchronous nature of the CPU and SMC operation. Figure 2.2 illustrates the low-level view of the SMC processor.

The internal register set is designed with 128 registers, so that it can contain the variables required for common operations, such as 5x5 and 7x7 convolutions and edge and span interpolation.

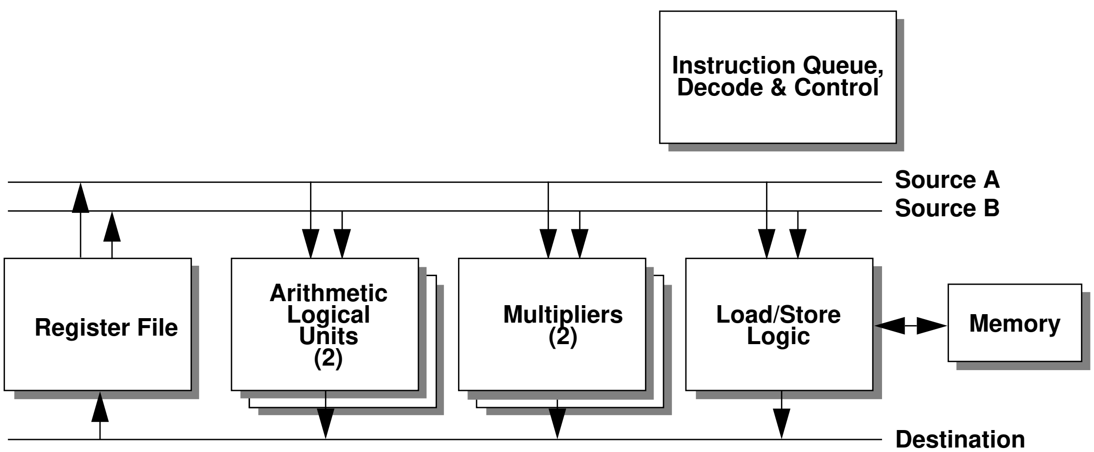

Figure 2.2 SPARCstation 10SX SMC Processor Low-level Architecture

The SMC memory data types include signed and unsigned bytes, shorts, and longs to support common pseudocolor, greyscale and full-color images. Memory data types are all converted to and from internal 32-bit register values, with additional specification of their fixed-point precision, if appropriate. Internal processing is carried out on 32-bit integers, with specification of the multiply source and destination precision.

SMC operations include common arithmetic and logical operators, with the addition of various multiply and multiply/add operations. The SMC itself has no conditional instruction execution, since it doesn't fetch instructions directly and can't perform branches. However, memory writes are conditioned by masking, and arithmetic operations can conditionally write destination registers and set the mask.

SMC operations can be specified as multiples (e.g., with an operation count field, or vector instructions). This approach reduces the bandwidth needed for the CPU command interface and matches the block memory behavior of the memory interface. The count is designed to be small enough to keep the SMC latency low. Longer counts, such as polygon spans or image scan lines, are broken by the driver software into smaller pieces.

The SMC includes an instruction queue to de-couple its execution speed from the CPU command write speed. The SMC can be halted at instruction boundaries, and the queue saved and restored. All internal SMC registers are accessible to the CPU on the Mbus. The SMC has no exception or error states caused by command execution.

SPARCstation 10SX Packaging

The SMC processor replaces the memory controller chip present on MBus-based SPARCstations. The SPARCstation 10 system requires a full CPU board upgrade to add the SPARCstation 10SX processor.

The SMC is located directly on the SPARCstation 10SX motherboard. Figure 2.3 illustrates the location of the SMC processor in the SPARCstation 10SX desktop.

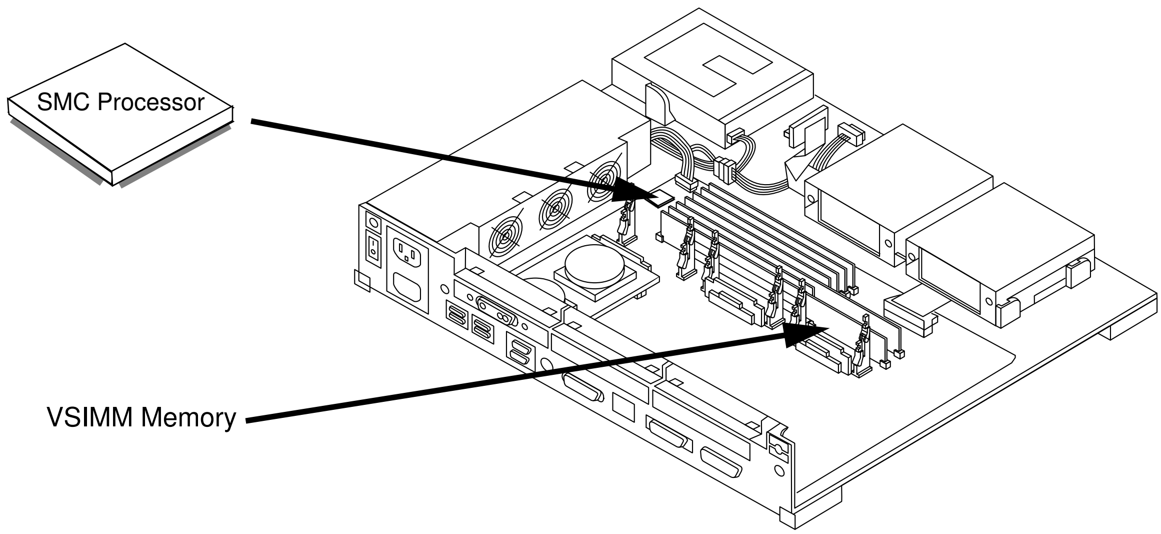

Figure 2.3 SPARCstation 10SX System Packaging - SPARCstation MBus-based Desktop

SPARCstation 10SX Configuration Options

The flexible memory architecture of the SPARCstation 10SX system permits several alternative display configurations, based on the amount of configured VSIMM memory present and user-selected options for resolution and frame buffer depth. Frame buffer depth and display frequency are determined at boot time. Additional selection of display options is performed via command line options before the initiation of the OpenWindows window server.

In SMCC's current desktop configurations, VSIMM options are based upon the installation of either one or two 4- or 8-MByte VSIMM boards in the dedicated VSIMM slots adjacent to processor memory on the SPARCstation motherboard (see Figure 2.3).

Table 2.2 illustrates the display options for the SPARCstation 10SX system, based on available VSIMM memory.

Features	Characteristics
Frame Buffer Depth	Supported in Processor Memory Space
Configurable Depth and Resolution 8-MByte VSIMM 4-MByte VSIMM	 1152x900 24 bits with 8-bit Overlay 1152x900 8 bits 1280x1024 24 bits with 8-bit Overlay 1280x1024 8 bits 1600x1280 24 bits with 8-bit Overlay 1152x900 24 bits with 8-bit Overlay 1152x900 8 bits 1280x1024 8 bits
Multiple Lookup Tables	Simultaneous Display of Truecolor Visuals - 512 distinct 8-bit visual colors and direct greyscale visuals without colormap flashing

Table 2.2 SPARCstation 10SX Frame Buffer Features

SPARCstation 10SX Graphics Features and Functionality

The SPARCstation 10SX system's SMC processor is a special-purpose computer. While it can accelerate a wide range of graphics and imaging functions, it is principally a highly functional imaging accelerator. Most imaging and many vector functions are very appropriate for implementation on the SPARCstation 10SX system.

Access to the SPARCstation 10SX graphics capabilities is achieved by the Solaris® graphics and window system APIs. As a result, developers of graphics and imaging applications need not concern themselves with SPARCstation 10SX instructions and their optimization. The APIs employ their internal pipelines to optimize SPARCstation 10SX instruction execution.

Integer Arithmetic

As mentioned previously, the SMC is an integer processor. It is able to support fixed-point precision (integer plus fractional bits) suitable for graphics and imaging, while geometric and numeric algorithms are better left to the floating-point capabilities of the host CPU.

The approach of utilizing the CPUs floating point has the additional advantage of dual processing. While the CPU is performing a floating-point operation, the SMC can be performing related integer operations.

Vectorization

To achieve optimum performance, the SMC processor supports 'vector' instructions, that is, ones that perform a sequence of operations. This allows the SMC's instruction bandwidth across the MBus to be relatively low compared to the SPARC CPU — usually on the order of 20-25%. Many functions can be trivially vectorized, while others can be recast into vector form with some thought.

The CPU write buffers and the SMC instruction queue enable load balancing between the SMC processor and the CPU, so that single-operation instructions are efficiently executed when intermixed with longer vector instructions. Often, the beginning or end loops need to be 'filled in' with single-operand or short-vector instructions, and the SMC's variable length vector instructions simplify this.

Conditionals

The SMC processor is most useful when executing a stream of instructions unconditionally. CPU read back of the SMC status registers will halt write pipelining. For this reason, several innovative features of the SMC processor instruction set, including the ability to conditionally write to registers or memory based on a bit mask that can be set by a comparison, serve to implement conditional processing without a test and branch. This allows conditionals to execute while preserving write pipelining performance.

Accelerated Memory

The SMC processor operates directly on physical memory, or on its internal register set. On a SPARCstation 10SX system, the video memory (VRAM) and dynamic memory (DRAM) are in the same physical address space, so the SMC processor can accelerate pixel operations on both of them. When the SMC has physically contiguous blocks of memory to operate on, it can avoid the overhead of page boundary checking. This provides further performance optimization. Certain classes of rendering, in particular, rendering vectors to an off-screen image, or maintaining a Z-buffer, benefit by reserving some system memory for the exclusive use of the SMC at boot time.

For typical XGL applications, 8 MBytes of memory reserved as contiguous physical memory for the SMC processor, provides optimal performance.

Accelerated Rendering Functions

The architecture of the SPARCstation 10SX system provides for optimized rendering of a number of imaging and graphics operations. Solaris APIs, optimized to utilize the above-mentioned features of the SPARCstation 10SX SMC processor, combine together with pipelining, casting and other methods to result in extremely fast performance for most base-level operations.

Tables 2.3 and 2.4 illustrate the various types of imaging and graphics operations that are accelerated by the SMC processor.

Image Operation	Types
Move	Copy, BLIT, Fill, Clear
Point	Arithmetic, Logical, Composite, ROP
Conversion	Monochrome/Greyscale/Halftone Color/Pseudocolor/Dither
Geometry	Resizing, Rotation, Warping
Filtering	Convolution, Interpolation
Enhancement	Equalization, Contrast/Range Adjustment

Table 2.3 Accelerated Imaging Operations

Geometry Operation	Types
Symbols & Patterns	Text, Markers, Stencils, Patterns
Vectors (Lines)	Solid, Patterned, Wide, Depth-cued, Antialiased
Polygons	Filled, Patterned, Shaded, Z-buffered, Antialiased

Table 2.4 Accelerated Geometry Operations

SPARCstation 10SX System Performance

The SPARCstation 10SX processor delivers broad performance to a large range of graphics and imaging functions. These functions, illustrated in Table 2.5, represent common operations for both geometry and raster-based graphics rendering, which are accelerated through Sun's low-level APIs such as the XGL, XIL, and Direct Xlib libraries. Additional, 3rd-Party APIs layered above the XGL, XIL, and Direct Xlib libraries, will provide performance corresponding to the level of optimization at which they were ported. In general, most layered APIs experience substantial performance boosts using the SPARCstation 10SX system over common unaccelerated frame buffers. Chapter 3 describes the SPARCstation 10SX system graphics software interfaces in more detail.

Operation	Pixel Rate[1] (32-bit)
Fill, Tile	45 MPixel/s
Copy, BLIT, Scroll	20 MPixel/s
Stipple, Pattern	33 MPixel/s
Monadics	24 MPixel/s
Dyadics	17 MPixel/s
Blend	2.4 MPixel/s
Convolve (3x3 8-bit Separable)	6 MPixel/s
Convolve (3x3 8-bit Full)	2.5 MPixel/s

[1] 32-bit, except where otherwise specified

Table 2.5 Estimated SPARCstation 10SX Performance

Operation	Pixel Rate[1] (32-bit)
Convolve (5x5 8-bit Separable)	3.1 MPixel/s
Convolve (5x5 8-bit Full)	1 MPixel/s
Nearest Neighbor Resize, 2x	18 MPixel/s
Nearest Neighbor Resize, Arbitrary, 2.5x	6 MPixel/s
Bilinear Resize, 2x	3.6 MPixel/s
Bicubic Resize, 2x	1.6 MPixel/s

[1] 32-bit, except where otherwise specified

Table 2.5 Estimated SPARCstation 10SX Performance

While the SPARCstation 10SX SMC is an integer processor, 2-D and 3-D vector performance (assuming CPU floating-point calculation of transformations) will also render at significant rates. SPARCstation 10SX performance is not specifically limited to rendering operations, since the internal architecture provides support for non-display oriented imaging, compression, and correlation functions. Performance for a range of general 2-D and 3-D geometry and image operations are illustrated in Table 2.6.

Operations	Performance
2-D Vectors, 24 bit (Transformed, Clip Checked) Text	 600 K/s 180 K Characters/s
3-D Lines, 24 bit (Transformed, Clip Checked) Triangles (Z-buffered, Shaded, 100 pixel)	 300 K/s 24 K/s
Image Processing[2] Image Copy (8-bit Megapixel) Image Filtering JPEG Compression (CIF Resolution) Color Space Conversion Image Resize - 2x Pixel Replication Image Resize - Bilinear Interpolation	 40 Frames/s 15 Frames/s Symmetric 30 Frames/s ~80 Frames/s ~200 Frames/s ~45 Frames/s

[2] 512x512 pixel images, except where otherwise specified

Table 2.6 Estimated SPARCstation 10SX Performance

SPARCstation 10SX Software 3

SPARCstation 10SX Software Interfaces

The SPARCstation 10SX system is supported by all Solaris graphics and window systems APIs, including the XGL, XIL libraries, PostScript, and the OpenWindows Version 3.3 (X11 R5) window system. Industry-standard X-extension libraries, such as Xlib and PEXlib, are also available and are accelerated via the XGL and XIL foundation graphics libraries. Emerging standards such as the X Imaging Extension (XIE), and Display PostScript (DPS), together with their related APIs (XIElib, DPSlib), will be supported in the future.

As both an imaging and geometry device, the SPARCstation 10SX system accelerates each of the APIs mentioned above. The sections that follow briefly describe the foundation graphics interfaces and the functions accelerated on the SPARCstation 10SX processor.

Figure 3.1 illustrates the Solaris graphics and windowing APIs and their relationship to the SPARCstation 10SX system.

The XIL Imaging and Video Library

The XIL library is a foundation-level imaging interface providing the common functions required by many imaging applications. As a foundation software layer for imaging applications, the XIL library defines how imaging operations, such as display, image manipulation, and compression and decompression, are carried out.

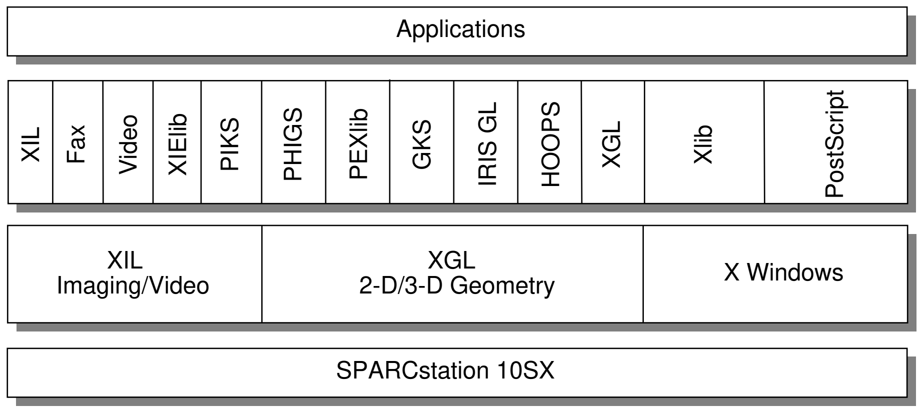

Figure 3.1 Solaris Foundation Graphics Libraries and Layered Interfaces

The XIL library contains three primary components: a programming interface specification for basic imaging functionality, a high-performance implementation of the specification, and a standard hardware interface specification, which enables third-party hardware developers to readily support the XIL library and applications built on it.

The XIL library is an object-oriented interface. As such, it hides the details of its implementation from the programmer. This enables the internals of the library to be extended without changing the interface in an incompatible way. C language bindings provide compatibility with other tools available under the Solaris environment including a large number of software development products. Figure 3.1 illustrates the XIL library in relationship to other foundation-level interfaces.

All XIL integer functions, whether they are 8, 16, or 32 bits, are accelerated by the SPARCstation 10SX system. Floating-point operations are handled by the SPARC CPU's integrated floating-point processor. The SPARCstation 10SX system accelerates broad classes of functions, including:

- Pan, zoom, scale
- Flips, transformations
- Raster rotation
- Sharpen, blur, convolution
- Monadic and dyadic image operations
- Blend, composite

The monadic and dyadic operations referred to above are of particular interest because of the structure of the XIL library, and its architectural reliance on them. The SPARCstation 10SX system is especially well suited to accelerating such operations.

The graphics porting interface employed with the XIL foundation library provides a flexible and powerful mapping to the functionality of the SPARCstation 10SX processor. Every XIL operator can be considered an atom. Groups of XIL operators concatenated together in a sequential manner are known as molecules.

Molecules are particularly important to the SPARCstation 10SX SMC processor because they allow the device to notify the XIL library at runtime that it (the device) wishes to provide hardware support for groups of chained operations. For example, the SMC provides image scaling and casting from 16- to 8-bit data. As a result, and unlike many other graphics devices, the SMC processor does not have to move the data back to CPU memory for XIL software to cast from 16 to 8 bits. The additional copy operation is avoided, because the SMC processor can scale and cast on-chip.

At runtime, the SMC device handler loads into memory and notifies the XIL library of the atoms and molecules it wishes to replace with its own mapping to hardware. The XIL library uses a directed acyclic graph to hold a list of these dependencies during application execution. When the XIL library encounters a known sequence of operations (defined by a molecule), it transfers the operation to the device for execution.

Table 3.1 illustrates the XIL operations accelerated by the SPARCstation 10SX SMC processor.

Operation	Description
xil_add	Add two images
xil_add_const	Add image and constant
xil_affine	Affine transformation
xil_and	Logical AND of two images
xil_and_const	Logical AND of image and constant
xil_blend	Alpha blend

[1] Other compression formats are supported by XIL, but are not accelerated by the SMC

Table 3.1 XIL Operations Accelerated by the SPARCstation 10SX System

Operation	Description
xil_color_convert	Color space conversion
xil_convolve	3x3, 5x5, 7x7 convolution
xil_decompress	JPEG decompress[1]
xil_copy	Copy image
xil_divide	Divide two images
xil_divide_into_const	Divide constant by image
xil_extrema	Find image minimum and maximum
xil_get_pixel	Get a pixel's value
xil_lookup	LUT operation
xil_multiply	Multiply two images
xil_multiply_const	Multiply image by constant
xil_not	Logical NOT
xil_or	Logical OR
xil_or_const	Logical OR of image with constant
xil_ordered_dither	Ordered dither
xil_paint	Paint an image
xil_rescale	Destination = k1 * source + k2
xil_rotate	Rotate an image
xil_scale	Resize an image
xil_set_pixel	Set a pixel's value
xil_set_value	Fill an image with a constant
xil_subtract	Subtract one image from another
xil_subtract_const	Subtract constant from an image
xil_subtract_from_const	Subtract image from constant
xil_threshhold	Threshold the image
xil_translate	Translation of an image
xil_transpose	Transpose an image
xil_xor	Logical exclusive OR of two images
xil_xor_const	Logical exclusive OR of image/constant

[1] Other compression formats are supported by XIL, but are not accelerated by the SMC

Table 3.1 XIL Operations Accelerated by the SPARCstation 10SX System

The XGL Geometry Library

The XGL library is the Solaris foundation geometry library providing the functionality and performance required by applications requiring geometry manipulation and display. The XGL library provides for optimization of 2-D and 3-D rendering, including high-quality lighting and shading, advanced primitives (such as NURBS and meshes, texture mapping, antialiasing, and transparency), and a flexible geometry pipeline. The XGL library includes sophisticated state-of-the-art features, such as dynamic tessellation of NURBS surfaces, as well as flexible pipelines for lighting and transformation.

The XGL library has an object-oriented interface that hides the low-level details of implementation from the application developer. As such, it is architected so that it may evolve as an interface, permitting new algorithms to be implemented within the XGL library without incompatibly changing the application program interface bindings. Table 3.2 illustrates the XGL operations accelerated by the SPARCstation 10SX system.

Operation	Description
2-D	- Solid and patterned thin lines, 8 and 24 bit - Triangles, quads, simple polygons, and general polygons, solid and patterned, 8 and 24 bit - Markers, 8 and 24 bit
3-D	- Solid and patterned thin lines, 8 and 24 bit - Triangles, triangle lists, quad meshes, and general polygons, lighted and shaded, 24 bit only - Markers, 8 and 24 bit - Antialiased lines, 24 bit only, 4 types of blending including transparent and opaque - Blended Z buffered triangles, triangle strips, quad meshes and general polygons, 24 bit only, 4 types of blending including transparent and opaque - Antialiased dot markers, 24 bit only - Raster copy, including depth conversion and Z-buffering - Accumulation buffering (for global antialiasing)
Other	- All other primitive/attribute combinations are accelerated via acceleration of dots, lines, and spans

Table 3.2 XGL Operations Accelerated by the SPARCstation 10SX System

The OpenWindows Environment

The OpenWindows environment is an MIT X Window System (X11 R5) compliant server. Applications that use the X protocol, running both locally or remotely, will receive acceleration from the SPARCstation 10SX System.

Window management functions can be optimized to take advantage of the SPARCstation 10SX architecture. The following functions are accelerated on the SPARCstation 10SX system.

- Hide/expose window
- Resize window
- Copy pixmaps to and from windows
- Text scrolling
- Get and put image

Many 2-D rendering primitives are also accelerated with the SPARCstation 10SX platform, including those associated with text, vectors, raster operations and others.

Direct Xlib vs. Xlib Graphics Libraries

The Xlib interface is the basic low-level 2-D graphics library shipped as part of the X11 Window System. The Direct Xlib library is a supplemental product that transparently enhances graphics rendering performance for graphics intensive Xlib applications in the OpenWindows environment. The Direct Xlib library was developed to provide an optimized Xlib interface for applications that typically run locally (client application and X Windows server on the same machine).

The Direct Xlib library allows most X Windows graphics rendering operations to occur directly in the client application process. This helps avoid the overhead associated with the copying of client application data to the X Windows request buffer, client/server context switching, and server parsing and processing of the request buffer for these operations. The key to the Direct Xlib library is the use of Sun's Direct Graphics Access (DGA) technology, which provides a synchronization mechanism between the Direct Xlib library and the OpenWindows server.

The principal benefit in using the Direct Xlib library is rendering performance. In general, most applications experience two times the rendering performance using the Direct Xlib library versus the Xlib interface. Use of the standard Xlib library is appropriate when graphics performance is not a factor, and when the application must be statically linked. The following common 2-D rendering operations in the Xlib interface and the OpenWindows environment that are accelerated by the SPARCstation 10SX graphics system include:

- Copy
- Solid, stippled, textured fill and Raster Operation (ROP)
- Vectors
- Text

Summary 4

Today's desktop computing environment is becoming the focus of converging technologies supporting window systems, video, imaging, CAD, multimedia and other rendering-intensive applications. While users today typically run several graphics or imaging programs, the display hardware products available are generally limited to one, or a few areas of graphics acceleration. Only by using expensive graphics accelerators do users overcome the limitations of their computers and realize the potential of their applications.

The SPARCstation 10SX graphics technology represents a fundamental breakthrough in delivering optimized imaging acceleration balanced with geometry. By employing a closely coupled CPU-to-memory controller design, the SPARCstation 10SX system is able to boost performance of all rendering operations, whether they are geometry, pixel, or raster image oriented.

The key benefits of such an architecture include lower cost per seat across the applications spectrum, from EDA and GIS markets to color publishing, imaging and visualization. By using the host system's dynamic memory for raster manipulation, the SPARCstation 10SX platform not only outperforms other 24-bit systems, but does so at lower cost.

The SPARCstation 10SX technology represents a clear industry direction in embedded graphics design. In the future, it will become commonplace to use the CPU for the simultaneous processing of floating-point data and non-rendering application code while a dedicated processor manages rendering directly to non-display memory.

References

Solaris XGL Reference Manual, SunSoft Inc.

Solaris XGL Programmer's Manual, SunSoft Inc.

Solaris XIL Imaging Library Reference Manual, SunSoft Inc.

Solaris XIL Imaging Library Programmers Guide, SunSoft Inc.

Xlib Reference Manual, O'Reilly & Associates. Inc.

*Xlib Programming Manua*l, O'Reilly & Associates. Inc.

Solaris OpenWindows User's Guide, SunSoft

Solaris OpenWindows Programmers Guide, SunSoft

X Window System User's Guide, O'Reilly & Associates. Inc.

X Image Extension Protocol Reference Manual, MIT X Consortium, Public Review Draft Version 4.1.2

An Introduction to Computer Graphics Concepts, Addison-Wesley, 1991

Solaris LIVE Technical White Paper, SunSoft, 1993

Solaris XGL Technical White Paper, SunSoft Inc., 1993

Solaris VISUAL Technical White Paper, SunSoft Inc., 1993

Solaris XIL Imaging Library Technical White Paper, SunSoft Inc., 1993

Sun Microsystems
Computer Corporation
A Sun Microsystems, Inc. Business

For U.S. Sales Office locations, call: 800 821-4643
In California: 800 821-4642

Australia: (02) 844 5000
Belgium: +32 2 730 38 11
Brazil: 011-55-11-887-9011
Canada: 416 477-6745
Caribbean: +1 415 688-9464
Finland: +358-0-502 27 00

France: (1) 30 67 50 00
Germany: (0) 89-46 00 8-0
Hong Kong: 852 802 4188
Italy: 039 60551
Japan: (03) 3221-7021
Korea: 822-563-8700

Latin America: 415 688-9464
Mexico: 001-525-580-5229
The Netherlands: 033 501234
New Zealand: (04) 499 2344
Nordic Countries: +46 (0) 8 623 90 00
PRC: 861-831-5568

Singapore: 224 3388
Spain: (91) 5551648
Switzerland: (01) 825 71 11
Taiwan: 2-514-0567
UK: 0276 20444
Elsewhere in the world, call
Corporate Headquarters:
415 960-1300
Intercontinental Sales:
415 688-9000

Printed in USA