Evolution of High Performance Graphics Systems

Nick England
Sun Microsystems, Inc.
Research Triangle Park, North Carolina

ABSTRACT

High performance graphics system hardware has benefited from improvement in component technology and advances in architectural design over the last decade. This paper examines the limiting factors in system design and analyzes several current architectures in light of these limits. Future trends in component technology are discussed along with possible effects on system architecture. Major themes of today which will greatly influence future systems include: parallelism, programmability, and integration.

KEYWORDS: Graphics Hardware, Parallel Processing, High Performance

INTRODUCTION

This paper will examine the development (both past and possible future) of interactive raster graphics display systems. The discussion will focus primarily on systems that might be built or bought for an advanced graphics application, excluding PC graphics and flight simulators in one fell swoop.

To examine this evolution, we will investigate the broad areas of technology, architecture, and integration. In each of these areas we will attempt to identify some important events and trends in the past and to point out likely trends and problem areas of the future.

TECHNOLOGY

To examine the impact of technology on high performance graphics hardware design, several portions of a typical system need to be examined--

1) Geometric calculation

2) Pixel value calculation

3) Pixel writing

4) Display

Although this paper attempts to separate discussions of architecture and technology, in practice the two are quite intertwined. Technology advances act as enabling agents for new architectural development.

Nevertheless, we shall concentrate on the functional elements making up systems in the technology discussion, and then address organizational issues. We shall first examine the capabilities of individual building blocks, then of connecting elements, before moving on to discuss how systems are assembled.

GEOMETRIC OPERATIONS

Common to most high performance graphics applications is the need for geometric manipulation. The techniques and mathematics have been fairly well-known for some time:

(1) building concatenated transformation matrices

(2) evaluating points on curved surfaces

(3) transforming 3-D elements

(4) clipping to viewing frustum

(5) calculating lighting from surface normals

(6) projecting perspective views

THE MAC

The key hardware element in carrying out many of these operations is the Multipler-Accumulator or MAC (Figure 1). A MAC's basic operation is Y=AX+Y. Multiple cycles or multiple MACs are used for:

(1) multiplying a 3x3 matrix times another 3x3 matrix to build a concatenated rotation matrix.

(2) finding the inner product of two 4x4 matrices to evaluate a point on a bi-cubic parametric patch.

(3) multiplying a 4-element vector times a 4x4 matrix to transform a point in homogeneous coordinates.

(4) finding the dot product of two 3 element vectors to determine shading.

The Past. Ten years ago, the first commercial integrated multiplier and MAC chips became available. Until that time, high performance 3-D graphics systems had used MACs built from dozens and dozens of components [SUTH68] or in a few cases, from hybrid digital/analog circuits called multiplying DACs [HAGAN68].

The first integrated MACs for graphics applications could provide a 16 bit integer multiply with 32 bit accumulation in less than 200 nanoseconds, yielding a 4x3 rotation plus translation rate of over 400,000 points per second [ENGL78]. This is sufficient for all but today's newest top performance commercial systems! Over five years ago, commercial floating point MACs appeared at similar rates, and today 20 MHz (50 nsec) floating point parts are available - a single chip can provide 1.6 million point transformations per second. Graphics has definitely benefited from the demand for MACs in other applications such as signal processing.

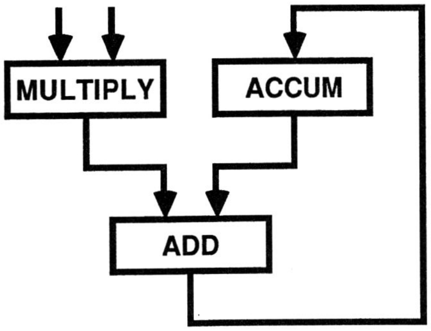

Figure 1 - Multiplier/Accumulator (MAC)

The Future. Faster clock rates are difficult to deal with going off and on chip, so there will probably be MACs integrated with other functions running up to 50 MHz (20 nsec) in the not so distant future - that yields a 4 million point/sec transformation rate. When greater rates are needed, multiple MACs will be integrated on chips. A 4-element dot product generator, for example, should handle almost all requirements for geometric operations in graphics.

DIVIDERS

Division is not a common operation in signal processing, so the benefits of commercial circuits have not been very evident. Division is necessary, however, for clipping lines or edges, and for perspective projection. Early systems used bit-serial clipping divider circuits [SPROULL69] as do some specialized proprietary chips today.

Many high performance graphics systems include microprogrammed bit-slice integer units which incorporate some hardware assist for bit-serial division. These bit-slice units are used for a variety of control and set-up tasks, and adding clipping and projection functions is quite straightforward. Such devices can cycle at 10 MHz, yielding about 300K divides/second - sufficient for 150K points/second perspective projection.

Since the comparison step of clipping is relatively easy and actual division takes place on only a small percentage of the elements, the computation limit should still be above 100K points/second for such devices.

For the last five years, however, Newton-Raphson iteration (requiring a small look-up table plus a floating point MAC) has been used to get acceptable perspective projection accuracy in only 8 cycles or less. Today, this yields over 1M points/sec. Future trends again point to greater integration, and chips with built-in look-up tables and division control logic are starting to appear. We might expect to see the use of a generalized clipping divider circuit (parallel and in floating point) for such applications in the future.

Once again, it appears that the computational requirements for the function can now be met quite readily and we must look elsewhere for the system bottlenecks and limiting factors.

DATA CONNECTIONS AND MEMORIES

It is hard to discuss data connection technology without getting too much into architectural issues, but we can once again examine what building blocks have been available and what we are likely to see in the future. The primary advances in technology during the 1980's have been in reducing size and power of integrated circuits. Speed of data connection parts has increased some, from around 40 MHz to 70 MHz maximum chip-to-chip connection rate. Size has been helped by integrating collections of random logic into programmable array logic (PAL) chips and more complex circuits into gate arrays. Data paths are often pinout limited, however, and no dramatic increase in speed or decrease in size has taken place, except where data connection elements (latches and multiplexers) and data processing elements (multipliers and ALUs) have been integrated onto one chip.

Quite often, however, buffer memories are used between various processing elements. Advances in technology have indeed significantly changed the size of memory. From the old 256 x 4 static random access memory (SRAM) of a decade ago to the 256k x 4 SRAM appearing now is quite a jump! However, the increase has primarily been in memory depth, not speed or width. This means we can buffer more items between processing stages, but not much faster, nor in much less board space.

Today's memories can be purchased with perhaps twice the speed of former ones, allowing us to readily build memories with the ability to be accessed by two processors at 10 MHz apiece. Again, this should not be a bottleneck, as this implies over 300K points/second for a simple implementation and speed-ups through parallelism are trivial. First-in, first-out (FIFO) memories have likewise gotten deeper, but not much wider or faster. A 20 MHz FIFO is quite commonly used to connect processing elements within a graphics system.

Future trends for data connection and memory components will probably follow the past, becoming deeper, but marginally faster. Again, 30-40 MHz is about a limit on chip-to-chip communication and faster systems will involve integration of these elements with processing functions on chip to reap speed and space benefits. Also, the use of surface mount technology (SMT) will become widespread to achieve PC board space savings.

Overall, it is reasonable to assume that component technology for geometric operations has not been, and will probably not, be the limiting factor in high performance graphics systems. We mush look at architectural issues, and at other parts of the system to discover likely bottlenecks.

PIXEL VALUE CALCULATION

One of the potential bottlenecks that we must consider is calculating where pixels should be written and what color they should be. For almost all graphical primitives, such as lines and shaded polygons, the task decomposes into set-up followed by iterative pixel rendering.

SET-UP

The set-up task requires some calculation and decision making capability and almost all high performance systems have used some microprogrammed processor to:

1) deal with polylines and polyhedra

2) sort polygon vectices

3) decompose polygons into trapezoids or triangles

4) swap drawing axes as necessary

5) initialize pixel drawing hardware

Since these systems almost always have separate set-up and drawing hardware, the two functions are usually overlapped and the slower one becomes the limiting factor on performance. A 10 MHz commercial bit-slice set-up processor can usually set-up a Bresenham or DDA drawing processor for a line or edge in 20-40 cycles (250K-500K lines/second). A more specialized, but still microprogrammed, set-up processor might run twice as fast (20 MHz).

For polygon drawing, several functions must be set up (dX/dY, dZ/dY, dS/dY, (S=shade)) and these may all be done in parallel for speed. This is done for each edge, and a subset for each scan line or span. These may all be calculated completely individually or a single parametric value (1/dY) may be used.

If the functions are not parallelized, then 40-80K trapezoids/sec and 125-250K scan lines/sec become the applicable limiting factors. These complicated set-up steps seem to be a bottleneck in overall system performance today and parallelism in set-up units is used to overcome this. This will be discussed further in a review of architectures.

In the future, advances in speed of set-up seem limited to possibly a factor of two in circuit speed, so increased parallelism seems the way to go.

PIXEL RENDERING

By pixel rendering we mean calculating the color and position of each pixel (or in the case of parallel pixel writing architectures, group of pixels) to be written. For line drawing, this involves one cycle to iterate position, and one to check for end-of-line. Simple circuits (Figure 2) can perform these calculations (an adder for position, a counter for length) in parallel at 10 MHz (1979) to 20 MHz (1989). Line drawing at 10-20 M pixels/sec (1-2M lines/sec assuming 10 pixel lines) has been easily achievable for the past decade.

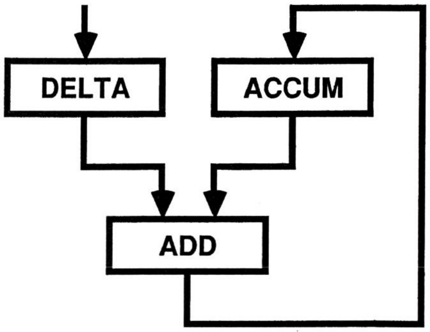

Figure 2 - Interpolator

For polygon drawing, simple parallel circuits again allow 10-20 M pixels/sec to be calculated for 100K-200K polygons/sec (assuming 100 pixel polygons). If parallel circuits are not used (in a 10 MHz microprogrammed bit-slice processor for example) figures of 3 million pixels/sec (300K lines/sec) and 2.5 million pixels/sec (25K polygons/sec) have been readily achievable for a number of years.

For the future, once again, circuit technology favors parallelism over computational speedup. The operations are simple and major advances beyond 30-40 MHz seem very difficult. It will be far easier to double the number of processors than to double the clock speed.

PIXEL WRITING

Pixel writing is almost always done into a frame buffer store built from dynamic RAM chips. Once again, as with SRAM technology, we find that the last decade has brought significant advances in memory density (from 16K chips to 1 M chips) but only a modest increase in rawspeed (from 400 nsec cycles to 250 nsec). This makes frame buffers cheaper and smaller today, but not much faster. A 512 x 512 x 8 frame buffer of a decade ago is over 5 times as large as a 1024 x 1024 x 24 frame buffer of today, but only 30-40% slower. Pixel drawing rates, then, have been possible at around 2.5 million pixels/sec (250K lines/sec and 25K polygons/sec), even for simple systems. Furthermore, dynamic RAM technology makes it faster to access bits within a column of the memory array, and this can be used advantageously to write pixels along a scan line or within a few scan lines at about twice the full random access rate. Again, we realize that only the most recent high-performance systems have come close to or exceeded (by writing multiple pixels in parallel) this rate which has been relatively constant for a decade.

For the future, it should once again come as no surprise that, while we might expect a factor-of-two clock rate increase, parallelism is far and away the best choice for increasing speed. For example, two newer systems (from Stellar and Silicon Graphics) [APGAR88, AKEL88] write spans of pixels in parallel to achieve around 40 million pixels/second. Meanwhile, we might expect simpler and cheaper systems to come closer to the one-pixel-at-a-time writing limit.

DISPLAY

The frame buffer store must not only be read and written from the drawing processor, it must also be continuously read at a fixed rate in order to refresh the display screen. Pixel rates for current display range from 10 MHz (640 x 480 interlaced) to 135 MHz (1280 x 1024 non-interlaced). Even the lowest of these rates requires parallelism of some sort.

VIDEO RAM

Here, though, a major advance in technology has occurred in the last five years. [WHIT 84]. Previously, dynamic RAM chips were used to construct frame buffers. Multiple pixels were read in parallel into a video shift register (perhaps 16 or 20 pixels at a time) and then shifted at serially to the rest of the display circuitry.

Texas Instruments was the first manufacturer to introduce a dynamic RAM chip with this video shift register on-chip. The first Video Ram (VRAM) was a 16K x 4 device with a 128 x 4 internal video shift register. Today, all frame buffers are built from VRAMs with 64K x 4 in existing designs and 256K x 4 showing up in new designs (Figure 3). With a maximum video shift rate of 20-30 MHz, four or five pixels must still be shifted out in parallel. Designs for higher resolution display (1600 x 1280 for example) shift out eight pixels in parallel. Again, we are unlikely to see significantly faster parts in the future. Speed will be obtained through parallelism.

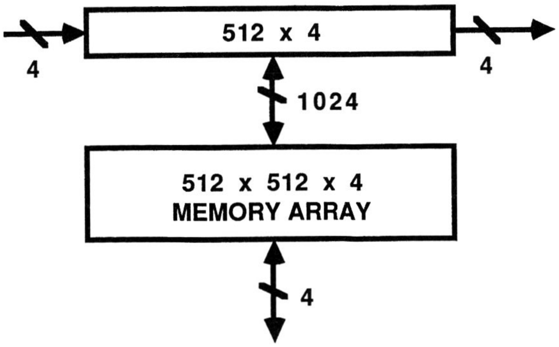

Figure 3 - Video RAM (VRAM)

FUTURE MEMORY

We are at an important point in memory density, however. Architectures which rely on writing a 2-D array of pixels are well-suited to 64K x 4 chips - a 4 x 4 array of such chips yields a 1024 x 1024 x 4 frame buffer with four 4-bit pixels per clock out the video port. With today's 256K x 4 chip, however, only 4 chips are needed for the same size buffer with the same video output rate. Consequently only four writing processors can be applied in parallel - and must be applied along a scan line. In a next-generation 1M x 4 chip, only one processor can be used and the video port provides only 25% of the needed ba ndwidth. Therefore, we must see a change in technology for the next generation. Proposals include x8 organization, and going to a two megabit chip instead of the four megabit allowed by the advances in semiconductor technology. One hope is that a 256K x 8 VRAM would still allow parallelism and perhaps be half the price of a four megabit chip.

Note, however, that mid-range systems are likely to continue to benefit from at least one more round of VRAM advances. Such systems will use the increased memory depth for double buffering (as high performance systems do) as well as z-buffering (for which high performance systems have an additional parallel memory).

VIDEO

Coming to the last part of the system, we find that high performance systems have reaped great benefits from technology advances (as have all raster display systems). Video Digital-to-Analog-Converters (DACs) and Cathode Ray Tubes (CRTs) have steadily advanced from 640 x 480 interlaced 30 Hz displays of a decade ago, to the common 1280 x 1024 non-interlaced 60 Hz+ display of today.

DACs have advanced dramatically in size and speed. A decade ago, an 8 bit, 10 MHz video DAC occupied six square inches of board space. Today three 8 bit, 125 MHz DACs occupy a portion of a chip, the rest occupied by integrated video look up tables, data multiplexer, and overlay control - an entire board's worth of expensive components only a very few years ago. Brooktree has been the leader in providing these integrated RAMDACs (Figure 4). The combination of VRAMs and RAMDACs has turned frame buffer display into an easy design task.

FUTURE VIDEO

Current RAMDACs include 10 bit in/8 bit out look up tables and the future will probably eventually bring this to 12 bit in/10 bit out, covering almost all video display requirements directly. Clock rate will increase as well, to match CRT technology.

Today, a few very expensive 2K x 2K color tubes (even higher monochrome resolution) are available. The biggest influence of the future, however, is bound to be High Definition Television (HDTV). The economics of commercial development will affect the computer graphics world dramatically. HDTV breaks with the 5:4 or 4:3 picture aspect rations common in graphics today. We should expect to see very large, very wide (2:1 aspect ratio) tubes appearing in graphics applications (Figure 5). Color tube resolution for HDTV is about 2K x 1K and we should expect to see high bandwidth versions of such CRTs.

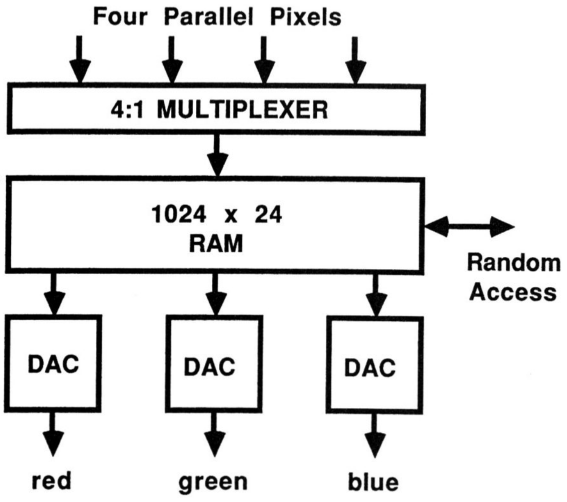

Figure 4 - Video RAMDAC

The HDTV system is interlaced and severe flickering would occur in high contrast small area formats such as line drawing. The effects of this flicker are minimized with anti-aliasing, but we will probably see little acceptance of HDTV tubes until higher bandwidth non-interlaced versions appear. These large screen displays will be roughly equivalent to two of today's largest graphics displays, truly enabling use of the electronic "desktop" instead of (as Fred Brooks puts it) the "fold-down tray in the center seat of a packed 737."

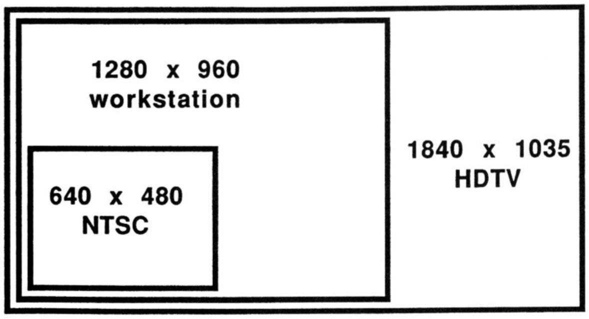

Figure 5 - Display Formats

TECHNOLOGY SUMMARY

The progress in technology over the past decade has primarily affected size and cost. Raw speed has advanced by perhaps a factor of two, but this pales in comparison to a factor of sixty-four increase in memory density, for example.

The basic capability of a top-end graphics systems remains the same in many ways, however. Tim Van Hook's CVD-1 of 1982 [IVER82] built with off-the-shelf components, delivered 250K z-buffered polygons/sec. Today's very best systems are starting to approach that performance level. The major difference is that vendors today can include a very high performance workstation for the same cost and in the same size package as the CVD-1 graphics processor.

However, there have been several quite significant advances in technology which have enabled us to do new things, or to do old things far more easily.

1) floating point arithmetic chips

2) Video RAM

3) integrated RAMDACs

4) high resolution CRTs

In addition, the advent of readily accessible integrated circuit design and fabrication (gate arrays, silicon compilers, etc.) has had a profound effect on the economics of making specialized application specific circuits. This has freed system designers from having to depend exclusively on chip manufacturers and is a large part of today's surge in affordable high performance graphics systems.

By far, the biggest effect technology advances are having today, and will continue to have in the future, is in making parallel processing techniques economically feasible. This will be the driving force for the architectures of the future

ARCHITECTURES

Having examined the components from which systems are build, it is time to turn our attention to the more important question of how those components are used. We should start by summarizing the performance characteristics of today's state-of-the-art components (Figure 6).

FUNCTION	COMPONENT	OPERATION	OPS/ SEC	LINES/ SEC	POLYS/ SEC
GEOMETRY	MAC	XFORM VERTEX	20M	1.7M	560K
		XFORM NORMAL	20M	1.7M	560K
		LIGHT VERTEX	20M	----	560K
	DIVIDER	CLIP	20M	3.3M	1.1M
		PERSPECTIVE	2.5M	1.3M	420K
SET-UP	PROCESSOR	SET-UP LINE	10M	500K	----
		SET-UP EDGES	10M	----	80K
		SET-UP SPAN	10M	----	25K
DRAWING	INTERP	LINE	3M	300K	----
		POLY FILL	2.5M	----	25K
	MEMORY	PIXEL WRITE	5M	500K	50K

FIGURE 6 - TECHNOLOGY SUMMARY

The above chart assumes 10 pixel long vectors in a polyline (1 new vertex per line) and 100 pixel individual triangles (3 new vertices per poly). These are roughly in line with commercially quoted performance numbers.

UNIPROCESSORS

Our chart readily leads us to some interesting conclusions of what system performance we can expect. For a uniprocessor system, only the render and write times can be overlapped, yielding a total estimated throughput of 130K vectors/second or 10K polygons/sec. Indeed, this is roughly in line with results from the Sun TAAC-1 programmable accelerator [ENGL88] which performs all of these operations in a VLIW uniprocessor at 6 MHz. The TAAC-1 overlaps some operations but has a relatively low clock speed compared to state-of-the-art components, so the comparison is rough but probably valid.

PIPELINE WITH PARALLEL STAGES

Our second architectural analysis looks at a very typical separation of geometry, set-up, and drawing into separate processors yielding:

geometry	440K vectors/sec	115K polys/sec
set-up	550K vectors/sec	19K polys/sec
drawing	300K vectors/sec	25K polys/sec

This tells us that, with a single geometry processor, we should perform multiple set-up and drawing operations in parallel. Indeed, all new high-end systems do precisely that. We shall examine one of them in this regard (Figure 7).

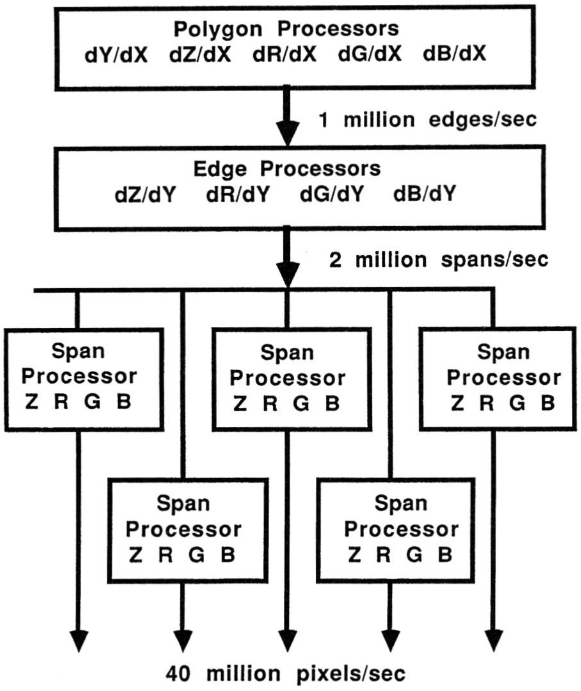

Figure 7 - SGI Iris 4D

The Silicon Graphics [AKEL88] Iris 4D system uses a Polygon Processor to compute, in parallel, slopes for Y, Z, R, G, B relative to X for trapezoids. This data is passed to an Edge Processor which walks along the edges of the trapezoid computing, in parallel, slopes of Z, R, G, B relative to Y for vertical spans. The Edge Processor feeds five Span Processors and each of these iterates Z, R, G, B along its span.

Other systems from Stellar [APGAR88] and Raster Technologies [TORB87] are basically similar in the use of parallel interpolation and pixel writing. We should expect to see such architectures continue to be popular, using parallelism in all parts of what is essentially a pipelined process. For example, the Silicon Graphics system uses multiple pipelined geometry processors, while the Raster Tech uses parallel geometry processors. This latter system uses tagged elements so that sequentiality can be maintained when necessary. This is an important point of consideration for future systems. The parallel drawing processors within most systems only handle one graphical element at a time, and the Z-buffer drawing operation removes sequentially problems in most cases anyway.

HIGHLY PARALLEL DRAWING PROCESSORS

There are a number of experimental systems where parallelism of this type is carried out to a very high degree. The Triangle Processor [DEER88] and the SAGE chip [GHAR88] are examples of scan-line parallel processors, one with a processor per pixel, the other with a processor per polygon, both with pipelined communication. The most advanced parallel system of this type is the Pixel Planes architecture [FUCH81] where a two dimensional array of processors (processor per pixel) is used. Work in this area will continue as it is a challenging and potentially very rewarding area of research and development.

PIXEL CACHES

One variation on the parallelism theme should be mentioned here, the use of a pixel array cache updated and then written to the frame buffer. The Hewlett Packard [GORIS87] high performance system writes one pixel at a time into a 4 x 4 cache and then the cache is written to the main frame buffer. For a z-buffer operation, of course, the cache must be loaded with the correct z-depth data before being updated. In the latest version of the Pixel Planes system [FUCHS89A, FUCHS89B], a 128 x 128 array of parallel processors is used as a combination processor/cache and then data is read/written into a much larger backing store and thence to a frame buffer for display. This new Pixel Planes system also incorporates the ability to include multiple parallel geometric processors as well as parallel pixel rendering units.

PROGRAMMABLE PROCESSORS

One of the very important issues in high performance system design involves the trade-off of flexibility for specific functionality. Most systems have hardwired line and Gouraud shaded triangle hardware. An emerging trend, however, is to incorporate more flexibility into systems, so that features can be added or modified to more closely match the requirements of particular applications. The Ikonas system [ENGL86] and Sun TAAC-1 board [ENGL88] are examples of completely programmable uniprocessors optimized and used for graphics and imaging operations. The Pixar CHAP [LEVIN84] is another example, this time with four processors in an SIMD arrangement. The newest systems from Stellar (APGAR88) and Silicon Graphics [AKEL88] have programmed parallel processors at the pixel rendering level. In addition, the Silicon Graphics geometry pipelined programmed processors. The Stellar and Ardent [DIEDE88] machines as well as the newest Apollo system use general purpose floating point hardware within the workstation CPU for the graphics geometry pipeline, for quite broad flexibility. This too, is an important trend, and we might expect that flexible performance will become a major differentiating characteristic. Less flexible systems will be used for lower cost (MCAD) applications or for real-time (flight simulation) applications, while flexible systems will be preferred for scientific visualization.

GRAPHICS MULTI-COMPUTERS

The trends to parallelism and programmability are leading to a new class of high end graphics systems, the graphics multi-computer. The advent of very powerful single chip computers including floating point hardware have made it possible to build an array of such processors tightly coupled to the frame buffer. The AT&T Pixel Machine [MCMIL87, POTM89] is the first machine of this type (Figure 8). While framebuffers have been tied to multi-computers (such as the Meiko Computing Surface), the Pixel Machine is the first to achieve an intimate binding of general purpose processors to display memory.

This tight integration offers one of the most intriguing and challenging areas of development for the future. It is particularly worth noting that the performance of a general purpose programmable processor has gotten high enough to be used in such a fashion. The rapid pace of technology development for such processors means that emphasis on algorithm development for such machines can bring benefits for multiple generations of hardware. Besides algorithm development, the other "hot topic" will be processor/memory organization as new machines are sure to come to market. Only time will tell, but the Pixel Machine appears to be one of the most significant developments in high performance graphics architecture in the last decade.

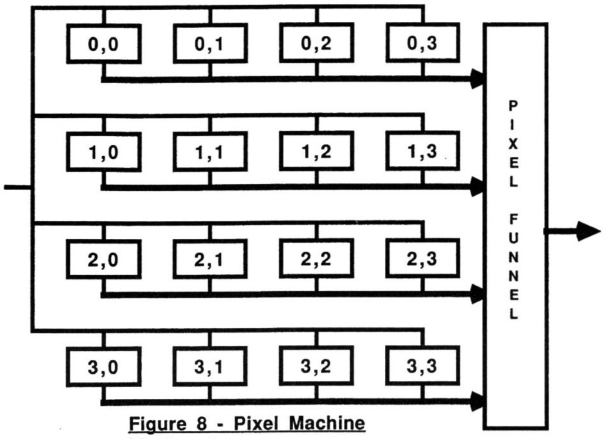

Figure 8 - Pixel Machine with 4 x 4 processor array

INTEGRATION

The other major development in systems design has been the integration of compute and graphics engines. The current high level of integration introduced by the Stellar and Ardent machines has been carried one step farther in the Intel 80860 processor [KOHN89]. This is a fully general purpose RISC processor chip (Figure 9), with on-chip caches, floating point unit, and graphics unit. This graphics unit allows the computation of multiple pixel color and depth values in parallel as well as Z-buffer comparison of multiple pixels in parallel (Figure 10). Although no machines have been introduced with this chip yet, projected performance of 500K transforms/sec and 25K Z-buffered, Gouraud shaded polygons/sec mean that the chip is sure to find its way into graphics workstations, accelerator boards, and graphics multi-computers. We can expect to see similar capabilities in more general computing chips in the future.

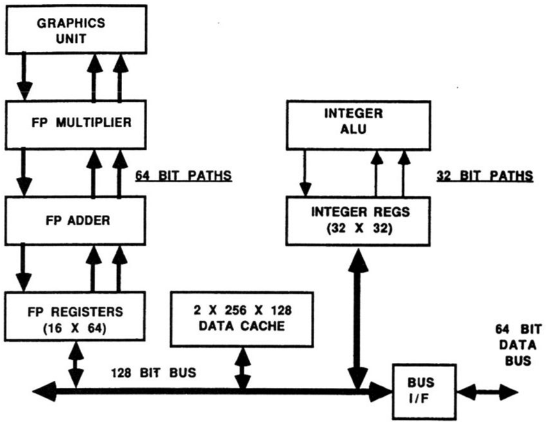

FIGURE 9 - 80860 Data Paths

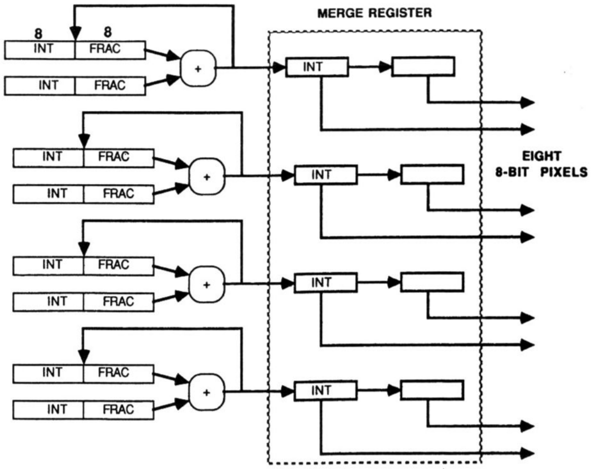

FIGURE 10 - 80860 Pixel Interpolate Instruction

CONCLUSION

In general, the past decade of high performance graphics system development has mainly been successful in bringing cost down with only a moderate increase in potential performance. Broad capability has been made available in a few niche systems, but in general polygon/line drawing has dominated the industry. In the last two years, however, we have seen significant architectural steps taken. Parallelism, flexibility, and integration will be the hallmarks of the next decade. We will continue to benefit from declining costs of traditional technology, but the frontiers of development will be in the utilization of graphics-enhanced processors tightly coupled to display memory.

REFERENCES

SUTH68 - Sutherland, Ivan E., "A Head-Mounted Three-Dimensional Display." Proceedings of the Fall Joint Computer Conference. (1968) 757-764

HAGAN68 - Hagan, Thomas G., Richard J. Nixon, and Luis J. Schaefer. "The Adage Graphics Terminal." Proceedings of the Fall Joint Computer Conference (1968)

ENGL78 - England, Nick. "A System for Interactive Modeling of Physical Curved Surface Objects." Proceedings of SIGGRAPH 78. In Computer Graphics 12, 3 (1978). 336-340

SPROULL69 - Sproull, Robert F. and Ivan E. Sutherland. "A Clipping Divider." Proceedings of the Fall Joint Computer Conference. (1968) 765-775

APGAR88 - Apgar, Brian, Bret Bersack, and Abraham Mammen. "A Display System for the Stellar Graphics Supercomputer Model GS1000." Proceedings of SIGGRAPH88. In Computer Graphics 22, 4. (1988), 255-262

AKEL88 - Akeley, Kurt, and Tom Jermoluk, "High Performance Polygon Rendering." Proceedings of SIGGRAPH88. In Computer Graphics 22, 4 (1988), 239-246

WHIT84 - Whitton, Mary C. "Memory Design for Raster Graphics Displays." IEEE Computer Graphics and Applications, March 1984, 48-65

IVER82 - Iverson, Wesley R. "Processor Animates 3-D Surface Images." Electronics (August 11, 1982). 149-50

ENGL88 - England, Nick. "Application Acceleration: Development of the TAAC-1." Sun Technology (Winter 1988). 34-41

TORB87 - Torborg, John G. "A Parallel Processor Architecture for Graphics Arithmetic Operations." Proceedings of SIGGRAPH 87. In Computer Graphics 21, 4 (1987). 197-204

DEER88 - Deering, Michael, Stephanie Winner, Bic Schediwy, Chris Duffy, and Neil Hunt. "The Triangle Processor and Normal Vector Shader: A VLSI System for High Performance Graphics." Proceedings of SIGGRAPH 88. In Computer Graphics 22,4 (1988). 21-30

GHAR88 - Gharachorloo, Nader, Satish Gupta, Erdem Hokenek, Peruvemba Balasubramanian, Bill Bogholtz, Christian Mathieu, and Christos Zoulas. "Subnanosecond Pixel Rendering with Million Transistor Chips." Proceedings of SIGGRAPH 88. In Computer Graphics 22, 4 (1988). 41-49

FUCHS81 - Fuchs, Henry and John Poulton. "Pixel Planes: A VLSI-Oriented Design for a Raster Graphics Engine." VLSI Design 2,3 (1981). 20-28

GORIS87 - Goris, Andy, Bob Fredrickson and Harold L. Baeverstad, Jr. "A Configurable Pixel Cache for Fast Image Generation." IEEE Computer Graphics and Applications (March 1987). 24-32

FUCHS89A - Fuchs, Henry, John Pouton, John Eyles, Trey Greer, Jack Golfeather, David Ellsworth, Steve Molnar, Greg Turk, Brice Tebbs, Laura Israel. "A Heterogeneous Multiprocessor Graphics System Using Processor-Enhanced Memories." University of North Carolina Dept. of Computer Science TR 89-005 (Feb. 1989)

FUCHS89B - Fuchs, Henry, et.al. "A Heterogeneous Multiprocessor Graphics System Using Processor-Enhanced Memories." To appear in Proceedings of SIGGRAPH 89.

ENGL86 - England, Nick. "A Graphics System Architecture for Interactive Application-Specific Display Functions." IEEE Computer Graphics and Applications (Jan. 1986). 60-70

LEVIN84 - Levinthal, Adam and Thomas Porter "Chap-ASIMD Graphics Processor." Proceedings of SIGGRAPH 84. In Computer Graphics 18,3 (1984). 77-82

DIEDE88 - Diede, T., C.F. Hagenmaier, G.S. Mirawker, J.J. Rubinstein and W.S. Wozley. "The Titan Graphics Supercomputer Architecture." IEEE Computer 21,9 (Sept 1987). 87-95

MCMIL87 - McMillan, Leonard. "Graphics at 820 MFLOPS." ESD: The Electronic Systems Design Magazine. (Sept 1987). 87-95

POTM89 - Potmesil, Michael, and Eric Hoffert. "The Pixel Machine: A Parallel Image Computer." To appear in Proceedings of SIGGRAPH 89.

KOHN89 - Kohn, Leslie and Sai-Wai Fu. "A 1,000,000 Transistor Microprocessor." IEEE International Solid-State Circuits Conference Digest of Technical Papers (Feb. 1989)