57

Supercomputing for one

Interactive, high-resolution graphics and vector processing combine for the first time in these desk-side units

Reprinted from April 1988 IEEE SPECTRUM

Supercomputing for one

Interactive, high-resolution graphics and vector processing combine for the first time in these desk-side units

In the culture of engineering—where a project's importance is often gauged by the computational power it requires—supercomputers have something of a mystique. Virtually all electrical engineers know what supercomputers can do, who makes them, and how much they cost. But because of its price tag, most EEs will never see a supercomputer, let alone use one—there are only 300 or so worldwide.

But supercomputing no longer necessarily means multimillion dollar machines produced by a few companies for a tiny technical and scientific elite. Minisupercomputers have been steadily gaining in popularity since they were first introduced a few years ago. Minisuper processing rates range from roughly 3 million to 20 million floating-point operations per second (megaflops) on the standard Linpack benchmark. They offer about one-third, some as much as one-half, the peak performance of a typical full-size supercomputer.

Last month, supercomputing made another great stride toward egalitarianism. Two U.S. manufacturers introduced their versions of a graphics supercomputer, a new class of machine that integrates a portion of the computational power of a supercomputer and the interactive, three-dimensional visual capability of a state-of-the-art workstation.

Graphics supercomputers are parallel, multiprocessor systems. They have high-speed integer processors and 64-bit vector processors like those used in supercomputers and minisupers to handle calculations that simulate complex physical events. They use the Unix operating system and have compilers that automatically transform and optimize code written in Fortran or C to exploit vector and parallel hardware, with no need for machine-specific extensions or assembly language. Supported by high processor-to-memory bus bandwidth and highly interleaved memory, graphics supercomputers can sustain more than 6 megaflops on a 100-by-100 compiled Linpack benchmark, and can peak at 64 megaflops. (The best technical workstations, like the Sun-4 from Sun Microsystems Inc. of Mountain View, Calif., run at no more than 1.1 megaflops on the Linpack.)

Priced between $80 000 and $150 000, these new machines offer about one-fourth the performance of a Cray X-MP for no more than one-twentieth the price. An EE, for example, could use one to shrink the design cycle by simulating such complex circuitry as a floating-point chip, or by precisely modeling the emission pattern of a new antenna. Further, graphics supercomputers make it possible to simulate circuits too large—and therefore too expensive—to be handled by existing computers.

As their name suggests, graphics supercomputers also provide integral graphics processing so that engineers can express computations visually, modify a design interactively, and see the results immediately. An engineer might alter some element of the circuit's design and get instant, visual feedback on how that change affects the chip's function or output. Likewise, modifying the antenna would produce an immediate change in the emission pattern shown on the computer's monitor.

C. Gordon Bell, Glen S. Miranker, and Jonathan J. Rubinstein
Ardent Computer Corp.

Unlike a supercomputer or minisuper, which is usually accessed by several users at once, a graphics supercomputer can be dedicated to a single user. Designed to be interactive, it lets a scientist or engineer close in on an optimal design or solution through step-by-step refinements. By executing computations under the direct control of a single user, it may actually provide higher throughput and productivity for the one application than would a faster machine that typically runs multiple jobs in a noninteractive, or batch, environment.

Start-ups carry the flag

The first two full-fledged graphics supercomputers both come from start-up companies: Ardent Computer Corp. of Sunnyvale, Calif., and Stellar Computer Inc. of Newton, Mass. But they are unlikely to have the field to themselves for long.

On March 1, when Ardent was introducing its machine in San Francisco, workstation manufacturer Apollo Computer Inc. was introducing its Series 10000 Personal Supercomputers in Boston. The Ardent and Apollo machines both incorporate from one to four reduced instruction-set processors that are said to offer, correspondingly, integer processing capabilities from 16 million to 64 million instructions per second (MIPS). But unlike the Ardent and Stellar machines, the Apollo uses proprietary floating-point hardware rather than vector processors, and it will not have full three-dimensional graphics capabilities until after it reaches market.

Other workstation manufacturers, like Silicon Graphics Inc. and Sun Microsystems Corp., both of Mountain View, Calif., are working on machines similar to Apollo's. Hewlett-Packard Co.,

Defining terms

Backplane: a hardware system for transferring data at very high speeds between a computer's circuit boards.
Linpack: a package of linear algebra subroutines, widely used as a performance benchmark for floating-point performance; as a benchmark, the 100-by-100 Linpack solves a system of 100 equations with 100 unknowns.
Port: a read or write channel to a memory or register file.
Vector: an ordered sequence of numbers often used to represent physical characteristics or quantities in a simulation.
Vector processor, vector unit: a high-speed processing unit designed to perform simultaneous operations on vectors.
Virtual memory: a technique using both hardware and software that permits storage of programs and data outside a computer's main memory. In a multiuser machine, virtual memory also protects data and code when several programs are running at once.

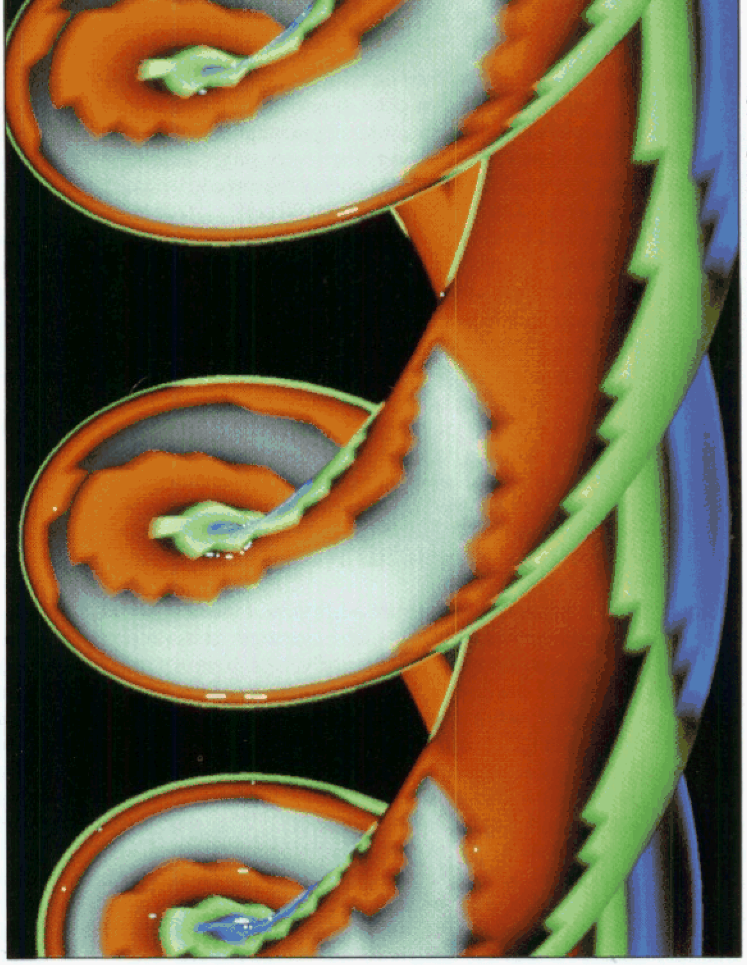
Palo Alto, Calif., Data General Corp., Westboro, Mass., and Prime Computer, Natick, Mass., are likewise expected to beef up the graphical and computational capabilities in their new technical workstations.

New, economical hardware and software technologies made graphics supercomputers possible. For Ardent's machine, called the Titan, very fast, high-density, high pin-count gate arrays let designers build an entire vector-processing unit out of just a few chips. (Similar units in conventional supercomputers occupy several boards.) Titan's 1-megabit dynamic RAMs pack up to 128 megabytes into a desk-side chassis—as much memory as in the larger, far more expensive Cray X-MP.

Low-cost, high-speed static RAM with an access time of 25 nanoseconds was essential for the buffers that support virtual memory, for the high-speed instruction and data caches, and for the large vector-register files that store data values for processing in the vector unit. Compiler technology, now graduated from the research laboratory, provided automatic vectorization and parallelization of code, often with an efficiency as good as or better than that of a human programmer.

Each Titan central processing unit (CPU) has an integer processing unit (IPU) built with a combination of off-the-shelf and proprietary hardware, as well as a custom-designed coexecuting vector unit. The core of each IPU is a single-chip, 32-bit R2000 processor from MIPS Computer Systems Inc. of Sunnyvale. The vector units are built of chips fabricated for Ardent by LSI Logic Inc. of Milpitas, Calif., and AMCC of San Diego, Calif., as well as commercial mathematics chips from Weitek Inc. of Sunnyvale. The vector units each executed 6.1 megaflops on the 100-by-100 Linpack benchmark, according to test results released in Technical Memorandum No. 23, dated Feb. 29, of the Mathematics and Computer Science Division of Argonne National Laboratory in Argonne, Illinois. Ardent engineers measured peak performance for a four-processor system at 64 megaflops.

Because it uses technologies aimed at keeping the price down, the Titan's CPU runs on a much longer clock cycle than would a supercomputer: 62.5 ns against 8.5 ns for a Cray X-MP. But by exploiting parallelism at many levels— especially within the vector units—graphics supercomputers can approach a substantial fraction of supercomputer performance in spite of the roughly 7-to-1 difference in the clock cycle.



Bob Olson, United Technologies Research Center

Supercomputers are often used to solve problems in computational fluid dynamics, with applications ranging from the study of automobile aerodynamics to the simulation of hypersonic propulsion of aerospace vehicles. A spinning helicopter rotor is simulated here in three dimensions on the screen of a Titan graphics supercomputer, by Ardent Computer Corp. Color shows the loading on the blades of the rotor, with blue representing low values and white the highest. Graphics supercomputers, introduced last month by Ardent and another U.S. company, pack a portion of the numerical capabilities of supercomputers into a compact, desk-size chassis and add high-resolution, interactive graphics.

Parallelism and performance

Scientific computing relies heavily on vector-oriented algorithms to simulate and model complex processes and physical events, many of which cannot be reproduced economically in a laboratory, or are of such short or long duration that they cannot be directly observed. But represented by an algorithm, such phenomena can be studied on a computer equipped with vector processing units.

In vector systems, designed to manipulate the huge arrays of numbers in simulations, a single instruction can carry out the same operation (such as load, store, or multiply) on all elements of an array or vector at once—parallelism that offers two advantages. For one thing, it reduces the ratio of overhead functions (such as fetching and decoding instructions) to actual work done, over the entire time it takes to perform a series of operations. Further, since vector operations specify a sequence of identical independent operations, they lend themselves well to efficient execution in pipeline fashion.

Pipelining, a standard CPU feature, is an assembly-line approach to executing instructions in which several instructions are in various stages of execution at the same time. Part of the processing takes place at each stage, and the passing of an instruction through all the stages of the pipeline completes its execution. Different kinds of instruction may require different stages, or a different order, which may cause delays—known as bubbles—in the pipeline, with a consequent loss in efficiency. But since vector operations are sequences of identical operations, bubbles do not occur and efficiency is improved.

The amount of work required to set up a vector instruction for execution is usually greater than for an instruction involving a single-number, or scalar, value. This extra work is done only once, however, for all operations on all elements of the vector. With a scalar instruction, on the other hand, every instruction must be set up individually. So if the vector instruction is long enough, the amount of work it accomplishes outweighs the additional set-up time.

A minimum length thus determines how many elements the vectors must have before the rate of operations on them exceeds the rate for an equivalent scalar routine. That minimum varies from machine to machine, and in well-designed vector machines ranges from 3 to 10 elements. On the Titan, vectors longer than 5 are processed more quickly, per operation, than equivalent scalars. The system patterns its vector units on those in Cray supercomputers, which have three major components:

- Vector registers—which store array elements to be processed by vector instructions.
- Memory pipes—high-speed, direct-memory access devices that moves data between vector registers and memory.

- Functional units—the pipelined hardware that operates on vectors, typically one element at a time in each pipe stage.

Vector registers came into use about 10 years ago as a far more economical way to provide operands for vector processing than the multiported, high-bandwidth capability that would be needed to transfer operands from main memory. Vector registers concentrate that capability in a smaller, faster resource where the operands are stored temporarily for processing in the functional units.

Graphics supercomputers emphasize such real-time processes as interactive graphics and so must be able to switch contexts (that is, respond to sudden commands from users or interrupts from peripherals) extremely quickly. One inexpensive way to reduce the time taken in switching contexts is to use multiple vector-register sets. With this approach, a set is assigned to each context. When the machine switches from one context to another, the old vector-register set does not need to be saved to memory and, later on, the new set does not have to be restored to the vector registers. The technique saves the Titan, which has 256 double-precision 32-element vector registers, as much as 200 microseconds on each context switch.

Using vector registers imposes the additional task of moving data at high rates between registers and memory. In designing a graphics supercomputer, the number of vector memory pipes (and the number of functional units) must be weighed against cost. Adding memory pipes to a vector unit substantially increases the size of the units that implement each pipe, control the unit's concurrent execution, and provide the extra bandwidth to the vector-register file needed by each pipe.

While more pipes result in more concurrent actions (arithmetic and memory load/store operations), there is a cost: in the Titan, for example, each extra memory pipe adds 15 000 gates to the floating-pointing unit's control circuitry, which has 80 000 gates already. Extra memory pipes also complicate the translation of memory addresses and the design of the vector-register file.

More functional units mean more parallelism and therefore faster computation, but they also require more hardware, more ports, and more bandwidth to the vector-register file—all of which add to the machine's cost. The Titan vector unit has three functional units, one each for floating-point arithmetic logic, multiplication, and division. The Cray X-MP/14, by contrast, has 12 functional units and can run at 31 megaflops, at a cost of about $200 000 per megaflop. The Convex C-120 minisupercomputer, with 10 functional units, costs $130 000 for each of its 3 megaflops; the Titan's cost breaks down to $13 000 per megaflop.

The main function of the vector unit's memory pipes is to load into the vector register file, as efficiently as possible, operands from the main-memory data structures created by a programmer. The pipes do this by supporting a variety of memory access patterns. To take a simple example, suppose a program calls for each of 32 variables to be compared with zero: any variable less than zero is to equal zero; any variable greater than zero is to be doubled. That small task could be done with a loop requiring the execution of 96 separate scalar instructions, along with associated bookkeeping functions and instructions for moving data to and from memory.

With the Titan's vector units, however, as well as with others similarly designed, the same task could be done with only three vector instructions. The key in that case would be a mask vector whose elements are typically set by a vector instruction that compares each element with the corresponding element in another vector. In this case, each of the 32 elements of the original vector would be compared with zero. The result is the mask vector, which has a zero in each location corresponding to an element in the original vector less than zero, and a one corresponding to each element greater than zero. The elements of the original vector can be set to zero at each element where the mask is zero and the result multiplied by two.

When it comes to virtual memory, graphics supercomputers part company with most U.S. supercomputers. Virtual memory greatly reduces the cost of a computer by making it possible to cut down on the size of its physical memory. Large programs can still be written, but at any given moment during a program's run, a portion of its code is on a hard disk or in some other secondary or mass-storage system. The operating system moves portions of the program, called pages, into and out of main memory as they are needed to keep the program running. When an instruction or piece of data is to be fetched by the CPU, its address must first be translated in order to determine if it is indeed in main memory, and if so, where.

Although virtual memory is a standard feature on workstations, minicomputers, mainframes, and minisupers, it is not used in supercomputers because address translations and page accesses reduce performance. U.S. supercomputers rely instead on massive main memories—a significant reason why they are so expensive. Graphics supercomputers, designed to be reasonably inexpensive, could never have such huge memories. Like minisupers, they employ custom-designed virtual-memory systems that let users run a multitasking operating system like Unix.

The Titan supports virtual-memory translation in several places. The IPU has its own translation mechanism—one for each memory pipe—that combines part of the vector unit's control chips with external RAMs. In addition, the graphics and I/O subsystems can translate virtual into physical addresses. Address translation and consistency across the multiple processors is managed in software. All those translation mechanisms share a set of page tables residing in memory and support 4-Kbyte pages.

Ardent used 1-Mbit dynamic RAMs to give the Titan main-memory capacity between 8 Mbytes and 128 Mbytes. Static column mode, a new feature available with these RAMs, lets the CPU access two memory locations in little more time than it takes to access one, which in turn lets the Titan's 32-bit-wide memory appear to the programmer like a 64-bit-wide memory.

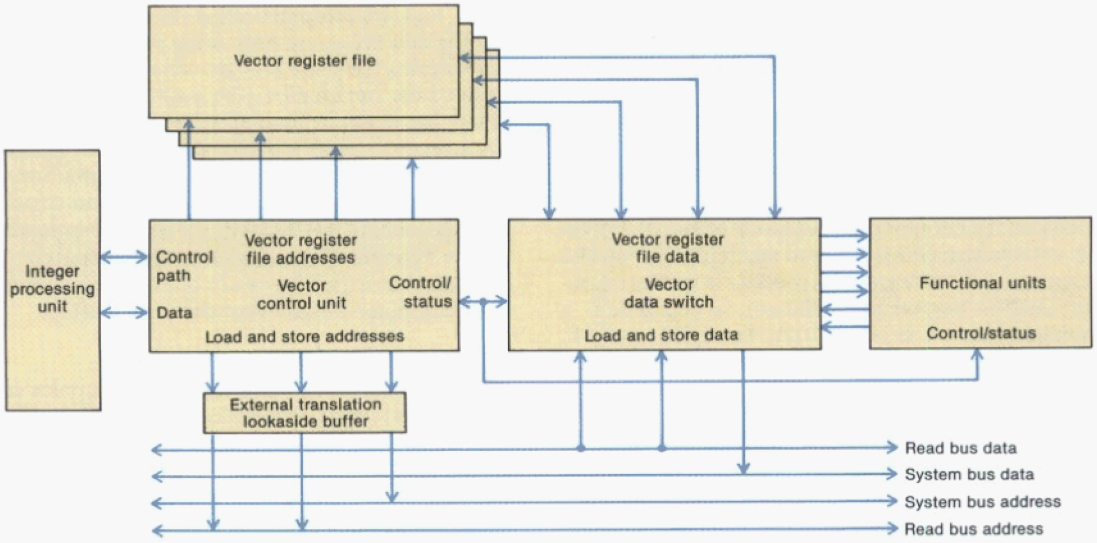
Other built-in mechanisms cut the time it takes for the Titan to access memory. In the vector unit, for example, the speed with which the memory pipes fetch vectors from main memory ensures that the pipe always has several requests to main memory waiting to be filled. To reduce the time it takes to fill those requests, the Titan, like many supercomputers and minisupers, uses memory interleaving. With this technique, the vector unit does not necessarily wait for one bank of memory to respond before going on to access data in another bank. That lets the banks cycle independently, and lets the processor have multiple requests for data outstanding.

Supercomputers and minisupers can typically service from 8 to 256 requests for data at once (one for each memory bank), an arrangement known as 8-way to 256-way interleaving; the Titan has 8-way or 16-way interleaving. The more interleaves, the faster the average access time to memory, since multiple requests can be filled simultaneously, but there is a catch: extra interleaves increase the costs of controlling them.

Once data values are in main memory, getting them to the CPU quickly is crucial to a vector-oriented computer's overall performance. The inner loop of the double-precision Linpack benchmark is a good example. The loop fetches two operands from memory, performs a multiplication and an addition, and stores the result. Executing that kernel requires moving 24 bytes to and from memory for every two floating-point operations. The execution can therefore be performed at 10 megaflops only if the bus linking memory and CPU can handle at least 120 megabytes per second. The Cray X-MP bus moves 3.2 gigabytes per second; the Alliant FX-1, 188 megabytes/s; the Titan can handle 256 megabytes/s.

Compilation is key

Most compilers for such older supercomputers as the Cyber 205 started out in sequential machines. In some cases, parallel and vector code generation, necessary to wring the most from the hardware, were added to those compilers as afterthoughts. Programmers had to master the intricacies of vector and parallel hardware themselves and incorporate into their programs



The Titan graphics supercomputer incorporates up to four vector processing units, each one coupled to an integer processor. Patterned after those used by Cray Research Inc., their main components are a vector register file, functional units, a data switch, and a control unit, which receives and decodes instructions from the integer-processing unit and executes them. Data fetched from the register file are routed through the data switch to the functional units, which perform arithmetic operations. Results are routed back through the data switch to the register file. Meanwhile, the control unit may generate addresses for storing and loading vectors moving between main memory and register file.

directives—special instructions to the compiler—so that the compiled program ran efficiently.

Minisuper maker Convex Computer Corp. of Richardson, Texas, was the first to relieve programmers of some of their hand-vectorizing work. The company's C-1 introduced a compiler that automatically found those parts of a program that could best be executed on the vector unit and rearranged code appropriately.

Convex's East Coast rival, Alliant Computer Systems Corp. of Littleton, Mass., was the first to relieve minisuper programmers of some of their hand-parallelizing work. Alliant's FX series, introduced in 1986, has a compiler that vectorizes and distributes code among multiple processors. In automatic parallelization, parts of the program that can be executed concurrently are arranged in loops. The compiler inserts protective synchronization code into these loops so that the multiple processors work together without interfering with each other. To lower the overhead of parallelism, the compiler includes as much source code as possible in each parallel section. During execution, the system and compiled code work together to balance the load over the multiple processors and so utilize the overall system as effectively as possible.

Graphics supercomputers and the latest minisupers incorporate powerful dependency-analysis techniques. These build a graph for each program that describes the constraints any optimization scheme must observe for the production of correct code. Older compilers typically made conservative assumptions about the extent to which software could be broken down for parallel or vector execution. But dependency analysis uses wherever possible exact knowledge about the program to limit constraints that inhibit transformations. The technique maximizes opportunities for vectorization and parallelization.

In systems software, the graphics supercomputer reveals its roots in both the supercomputer and the workstation. The graphics supercomputer supports users who write their own programs, as is common with supercomputers, as well as those who run standard packages, as workstation users tend to do. For the latter group, graphics-supercomputer compilers can handle applications in VAX/VMS Fortran and with Cray directives.

Supercomputer users often write their own application programs because they have no other choice. Manufacturers at first supplied proprietary operating systems and expected users to write software in-house. Because the expense of the machines kept the user base small, portability and compatibility of code were not issues.

However, Unix's support for multitasking and its portability of applications has encouraged minisuper makers to adopt the operating system as a standard. Indeed, Scientific Computer Systems Corp. of San Diego, Calif., the only company in the field that had bucked this trend with a Cray-compatible operating system, is now said to be moving toward Unix. Even Cray Research Inc., of Minneapolis, Minn., now offers Unix on the X-MP, the new Y-MP, and the Cray-2, and the company intends to optimize future systems for Unix.

Whatever its virtues, Unix was designed for small, single-processor minicomputers. It is not inherently well suited to the demands of supercomputer-class applications. Unix does well with small programs written in C but bogs down on Fortran applications with millions of bytes of source code and large amounts of input and output. Unix has evolved to accommodate machines equipped with large virtual memories, but it still cannot support multiprocessing without extensive changes to the heart of the system.

Minisuper and graphics supercomputer makers alike have had to adapt Unix to multiprocessing and real-time environments. Ardent, for example, added a fast-file system that delivers 1000 kilobytes per second by handling large I/O transfers at the controller level. This technique results in fewer disk accesses than is required in Unix System V. The Ardent system also allocates files contiguously on the disk, which reduces disk latency—the delay between a processor's request for the contents of a memory location and its receipt of the information.

Visualizing science

Much supercomputer work— such as finite element analysis, fluid and molecular dynamics, simulation, and weather prediction—produces complex, quantitative data that can best be understood through images. Scientists and engineers typically study these graphics displays on workstations or terminals coupled to a minisuper or supercomputer, an arrangement that has an inherent drawback: data cannot be moved fast enough from compute engine to monitor for users to see full-color, three-dimensional results as they are computed. Titan's designers eliminated that communication bottleneck by coupling the graphics directly into a 256-megabytes/s system backplane. Graphics data move so fast and generate and change images so quickly that a user can react intuitively.

While workstations are known for their interactive graphics, most achieve their performance through custom hardware. Such key graphics vector operations as 4-by-4 matrix transforms are typically hardwired into a custom geometry engine. This gives extremely high performance for a limited group of graphics operations, but it restricts high performance to that limited group and requires programmers to write graphics routines in special microcode. For any analysis beyond its computational capabilities, a workstation might access a larger machine through a network, a process that brings back the communication bottleneck problem.

The Titan handles most graphics processing with its general-purpose vector and integer-processing units. Any kind of graphics or image-processing operation, including fast Fourier transforms and convolutions, can be done that way, and because there is no specialized graphics pipeline, the high performance extends to all graphics functions. Instead of microcoding their routines, programmers can write graphics code in C or Fortran. There is no performance penalty for these graphics advantages—graphics instructions can be processed simultaneously with general-purpose computation in the CPU.

Graphics supercomputers do have some special-purpose graphics hardware. The Titan, for example, has a rasterizing subsystem for painting each bit of the 1280-by-1024 picture element display screen. With 24-bit pixels (eight bits each for red, green, and blue), Titan displays images with full color—that is, up to 16 million colors. The hardware can generate up to 200 000 Gouraud-shaded, full-color polygons per second, or sustain an animated sequence of 100 000 Gouraud-shaded triangles per second. The fastest workstation on the market, the Sun-4, has a 20 000 Gouraud-shaded polygon capability. (Gouraud-shading is a software technique for smooth-shading solid objects, more realistic than faceted shading but lacking the specular reflections of more sophisticated shading techniques.)

While vector and graphics hardware provide the raw speed necessary for interactive graphics, software is equally critical, because it determines whether or not users can work easily and effectively with the hardware. The Titan's main graphics software is the Dynamic Object-Rendering Environment, or Doré, created at Ardent by a team led by Michael R. Kaplan, an early developer of the spatial-subdivision ray-tracing approach.

Doré is a comprehensive library of graphics subroutines that lets a user, with or without graphics expertise, create a wide variety of imagery. Programmers using Doré do not need to write different code to represent the same object in different ways—they describe an object once and rely on Doré to produce it in other perspectives and render different styles. Doré also handles the intricacies of decomposition—breaking complex objects down into simpler representations that can be shaded and displayed. It can even tailor a graphics application to different hardware or graphics drivers.

Using Fortran or C code, programmers can add their own primitives, textures, shading, and rendering functions to Doré, to tailor applications to their particular needs. The additions are then portable across all Doré implementations, which can run on supercomputers and workstations as well as on Titans. Widely-used graphics standards, such as the Programmer's Hierarchical Interactive Graphics Standard, as well as low-level interfaces, can also be used as device drivers in interfaces with workstations.

Graphics supercomputers are at home in the write-your-own-code environment of supercomputers and minisupers. But it is the ready availability of standard application software in a variety of disciplines that will make or break graphics supercomputers in the increasingly crowded market for vector machines. Several packages for computer-aided mechanical engineering, computational chemistry, and fluid dynamics were introduced with the Titan, and two computer-aided electrical engineering packages, Verilog and Spice, are now being prepared for the machine.

Designing the Titan brought constant reminders of the limitations of the workstations the machine was being built to transcend. For example, simulating the Titan's backplane and memory arrays with Spice took about 1.5 million seconds of computer time, which in turn took a month on multiple Sun workstations. Ardent's designers estimate that a single-processor Titan would have spent about two weeks on the job. Correspondingly, simulating the Titan floating-point unit, which has about 250 000 gates, proceeded on a Sun at a rate of one clock per second. A graphics supercomputer would boost that rate tenfold.

Altering the design landscape

The next three years likely will see a graphics supercomputer with the performance of a minisuper and the price of a workstation. Supercomputer performance roughly equal to that of a Cray X-MP/14, but in a desk-side unit costing less than $100 000, is not far behind. It only takes a little imagination to see that individual access to computational power like that will soon be reshaping design work in such fields as electrical and mechanical design, fluid dynamics, seismology, molecular modeling, and algorithm development.

To probe further

Last January's *Spectrum* carried roundups of recent developments and products in supercomputers, minisupercomputers, and workstations ["Minis and mainframes," pp. 29–31, and "Design automation," pp. 35–37]. For more on reduced instruction-set fundamentals, see *Tutorial: Reduced Instruction Set Computers*, by William Stallings, IEEE Computer Society Press, 1986. The same imprint published Kai Hwang's *Tutorial: Supercomputers—Design and Applications* in 1984. State-of-the-art compilers are surveyed in "Unifying Vectorization, Parallelization, and Optimization: the Ardent Compiler," by J.R. Allen, to be published next month in the *Proceedings of the Third International Conference on Supercomputing*, available from the International Supercomputing Institute in St. Petersburg, Fla.

The history and basics of reduced instruction-set computers were covered by Paul Wallich in "Toward simpler, faster computers" [*Spectrum*, August 1985, p. 38]. The turbulent and expanding supercomputer and vector-processing field was the subject of a special report in the March 3 issue of *Electronics*. The report includes short outlines of both the Ardent and Apollo machines.

About the authors

C. Gordon Bell (F) is vice president of research and development at Ardent Computer Corp. of Sunnyvale, Calif. Best known as the architect of the VAX superminicomputer, Bell was recently chairman of the subcommittee on computer networking, infrastructure, and digital communications, organized by the Federal Coordinating Council for Science, Engineering and Technology. In 1986 and 1987 he was assistant director for computing at the National Science Foundation. He earned his B.S. and M.S. degrees in electrical engineering at the Massachusetts Institute of Technology in Cambridge.

Glen S. Miranker (M) is Ardent's chief architect. He previously served as engineering director at Valid Logic Systems Inc. of San Jose, Calif., and was involved in processor development at IBM's Thomas J. Watson Research Center of Yorktown Heights, N.Y. He holds a B.S. from Yale University in New Haven, Conn., and a Ph.D. in computer science from MIT.

Jonathan J. Rubinstein (M) was responsible for the design and architecture of the Ardent Titan's integer processor, cache, bus, and vector unit. Previously with Hewlett-Packard, he was the architect of the HP9000 Series 300 computer family. He earned B.S. and M.Eng. degrees in electrical engineering from Cornell University in Ithaca, N. Y., and an M.S. in computer science from Colorado State University in Fort Collins. ◆