

PARALLEL PROCESSING GETS DOWN TO BUSINESS

For years, the designers of highly parallel computers—those with hundreds or thousands of processors working together on a single problem—have been like “composers describing how wonderful their symphonies are before they are ever performed,” in the words of one researcher. While hundreds of theoretical designs existed, no such machine had been built. Now years of research are starting to pay off, with a number of parallel computers being built, tested, and even sold commercially. These computers and others likely to appear during the next 18 months will begin to answer some of the questions that have long plagued this area: Can parallel processors actually top the speeds of more conventional supercomputers? How hard will they be to program? What designs will succeed?

The activity in the parallel processing market began to heat up about a year ago. The current projects not only signal an enthusiasm for parallel architectures but also illustrate differences of opinion about how many processors are most effectively linked within a single machine. NASA's Goddard Space Flight Center already uses a limited-application computer, the Massively Parallel Processor—built by Goodyear Aerospace (Akron, Ohio)—with over 16,000 subprocessors. For more than a year, Bolt Beranek and Newman (Cambridge, Mass.) has marketed the Butterfly multiprocessor, which can encompass more than 100 subunits, and has government contracts to build ten systems. In February, Intel Scientific Computers (Beaver-

ton, Ore.) announced a 128-processor machine, the iPSC. In the course of this year, Thinking Machines (Cambridge, Mass.) will complete its first commercial Connection Machine with a whopping 64,000 processors. And next year the Japanese plan to complete the Sigma-1, in which a program is divided among 200 processors automatically, instead of by a programmer. Other projects will be completing full- or partial-scale machines in the same period.

While the ultimate aim of most parallel processor designers is to create extremely powerful machines that will outshine current supercomputers such as those from Cray and Control Data (both in Minneapolis), most of the

Machines with hundreds or thousands of processors begin their assault on traditional computers

initial parallel machines will compete against superminicomputers like DEC's VAX-11/780. The supermini market is potentially much larger than the specialized supercomputer field, and the first versions of most parallel machines don't yet equal the performance of the supercomputers. Still, the likely applications for future parallel processors are similar to those currently addressed by the supercomputers. The computers will be used by scientists doing simulations of weather and other complex processes, by engineers studying aerodynamic flows, and by artificial intelligence researchers designing expert systems.

Why parallel? The interest in parallel computers, which dates back two

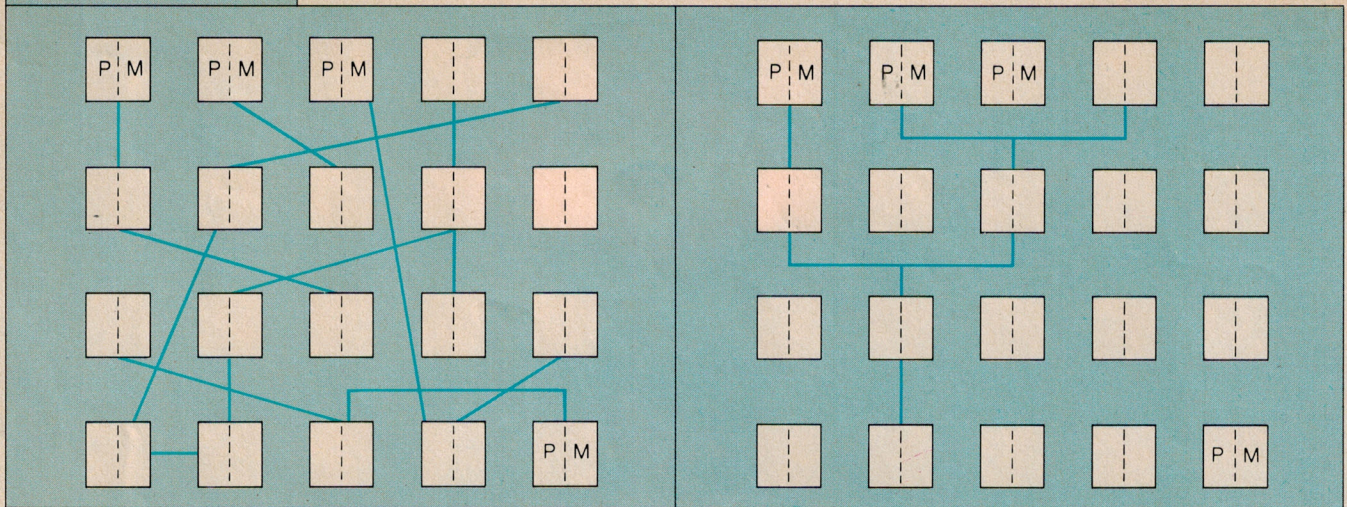
decades, is based on some straightforward reasoning. Conventional serial computers, employing the basic architecture first developed by John von Neumann in the 1940s, work by taking data from a central store, operating on it, and returning it to the store. For many applications, this is an exceedingly inefficient procedure. Take, for instance, an image-processing application in which a series of 100 operations must be applied to millions of individual picture elements (pixels). A serial computer would waste most of its time repeatedly retrieving and storing a vast amount of data, with relatively little time devoted to actual computation.

By contrast, a parallel processor with, say, a processor for every 10 pixels, could take several pieces of data and perform a series of operations on them in parallel before returning them to a distributed memory; most of the time would be spent in processing. Since accessing a memory typically takes longer than processing information, the parallel computer theoretically could work far more efficiently than the von Neumann one.

What's more, parallel multiprocessors should, in principle, cost less than von Neumann machines. The only way to speed up a serial computer is to reduce its clock (or step) time—the time between the start of one operation and the start of the next. This is what supercomputer vendors have done; their computers currently run with step times of about 10 nanoseconds and, within a few years, will drop to about 1 nanosecond. But such high-speed devices require the latest in very-large-scale integration (VLSI) semiconductor technology, as well as elaborate cooling systems. A hundred devices operating

by Eric J. Lerner

Connection Machine



Thinking Machines' Connection Machine, still under development, will contain 64,000 processors (P), each with its own dedicated memory (M). Conceptually, the machine has a 12-dimensional geometry, but links between the thousands of processors

are dynamically reconfigured as required by each application. In some cases these connections may appear totally random (left); in others, such as a program with a tree structure, the connections adjust to resemble the application (right).

in parallel, each with a 100-nanosecond step, could theoretically match the power of a single 1-nanosecond device, with each costing less than a hundredth as much as the fast device.

Finally, there are clear limits to how much speed can be achieved by conventional supercomputers. Electrical signals traveling close to the speed of light can move only about 30 centimeters in 1 nanosecond, so computers with time steps at this level must be built within a one-foot cube lest their speed be gobbled up in transmission delays. This size is much smaller than that of today's supercomputer central processing units, or CPUs (typically at least a meter on a side), and cooling requirements make it unlikely that such compactness can be achieved under present technologies.

The obvious way to overcome these limitations is with parallel processors, each working on a part of a large problem. Such processors, with relatively slow step times, will generate less heat and can be densely packed. At the same time, physically larger computers with high aggregate speeds can be built without transmission delays becoming a problem.

In practice, though, there are serious obstacles to realizing the benefits of parallelism. Communication among a large number of processors and the cost of the connecting hardware pose major problems. So does the task of coordinating the processors and ensuring that many of them are not idly waiting for others to give them needed results.

To overcome these problems, dozens of different architectures have been proposed. Broadly speaking, they fall into two categories: single instruction/multiple data stream (SIMD) machines and multiple instruction/multiple data

stream (MIMD) machines. In SIMD machines the sequence of instruction is similar to that of a von Neumann device, but the instructions are carried out in parallel on more than one set of data by multiple processors. But not all problems have the necessary structure for SIMD. MIMD machines provide the most parallelism and flexibility, since they are not constrained by the order of instructions in a single stream, but they are harder to synchronize.

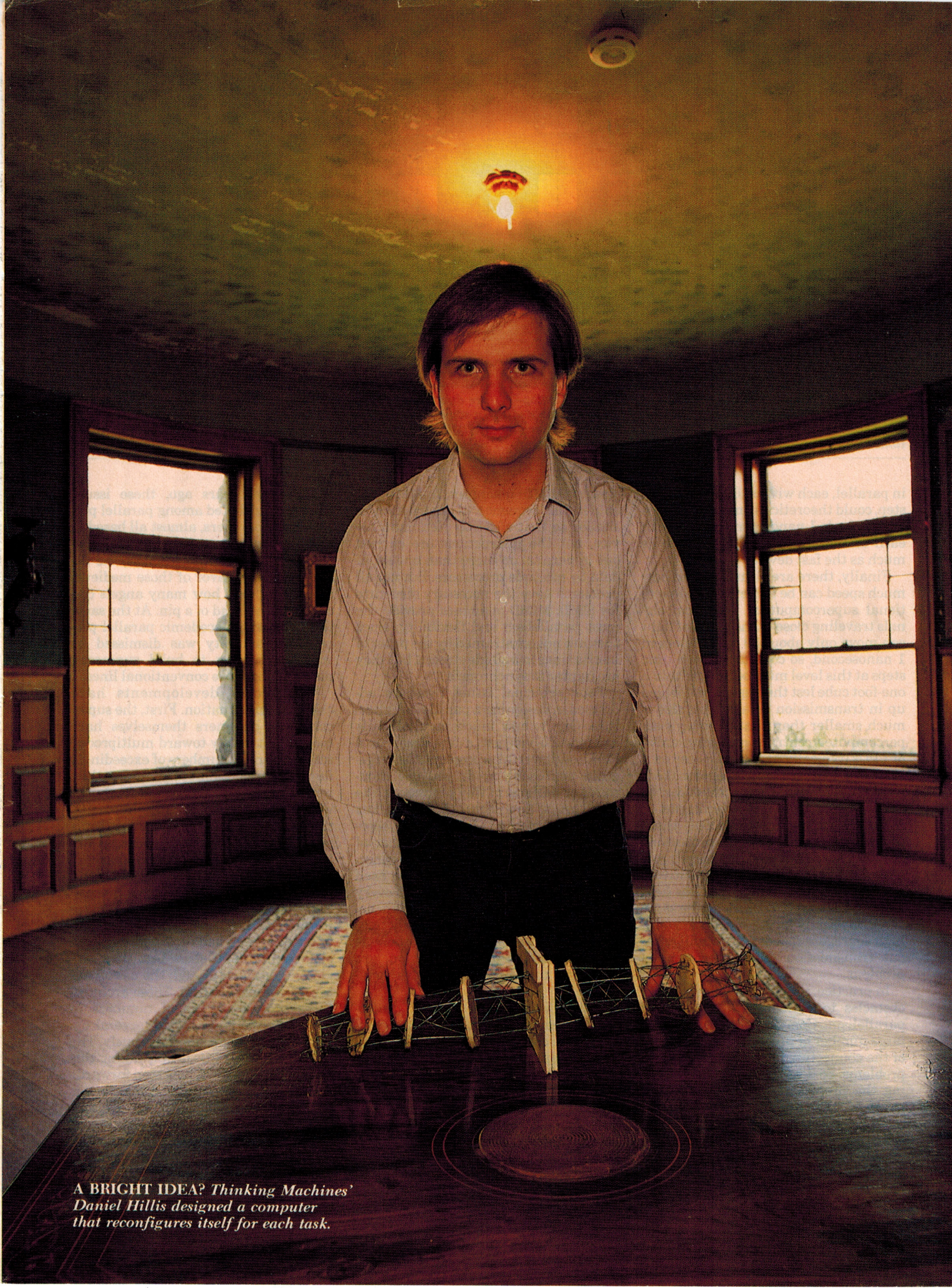
Four principal issues must be addressed in designing a parallel architecture. One is how much parallelism is desired—do you use a few dozen or a hundred powerful processors (the large-grain approach), or do you use thousands or millions of tiny processors (the small-grain approach)? A second key issue is the method of connection, especially critical with a large number of processors. High-performance connections such as crossbars (in which each processor is directly connected to every other one) reduce communication delays but are expensive and highly complex, while simple systems, such as nearest-neighbor-only connection, are cheap but constrain the system's application.

A third issue is how to attach processors to memory—will they share memory, or will each have its own? The final and perhaps most controversial issue is who decides how a problem is to be broken down into parallel parts. If this is decided entirely by the computer and its language compiler—as in the approach known as dataflow—programming is simplified, but the performance will be relatively inefficient. If the programmer must decide, performance can be optimized, but programming may become prohibitively difficult.

Until two years ago, these issues were hotly debated among parallel processing researchers, almost all based in universities. Since none of the machines had been built, though, the debates had the flavor of those medieval arguments over how many angels can dance on the head of a pin. At the same time, the whole academic parallel processor community was dismissed by those continuing to build supercomputers along more conventional lines.

Two recent developments have changed this situation. First, the supercomputer designers themselves have moved tentatively toward multiprocessors with small numbers of exceedingly fast subunits (generally fewer than ten). While such machines properly fall within the parallel processing realm, their continued emphasis on extreme power for each processor distinguishes them from the majority of new parallel machines, which rely on the aggregate power of many less expensive processing elements. Even before this move toward parallel processors, supercomputer designers had incorporated some elements of parallelism into their single-CPU designs through the use of vector processor pipelines. These architectures process strings of numbers in assembly-line fashion with several operations occurring simultaneously. The use of multiple CPUs is now taking this simple parallelism a long step forward.

Also pushing parallel computers is increased research funding. Japan's Fifth Generation and Superspeed Computing projects and the Defense Advanced Research Projects Agency's Strategic Computing Initiative have begun to spend millions of dollars to make parallel processing machines commonplace soon.



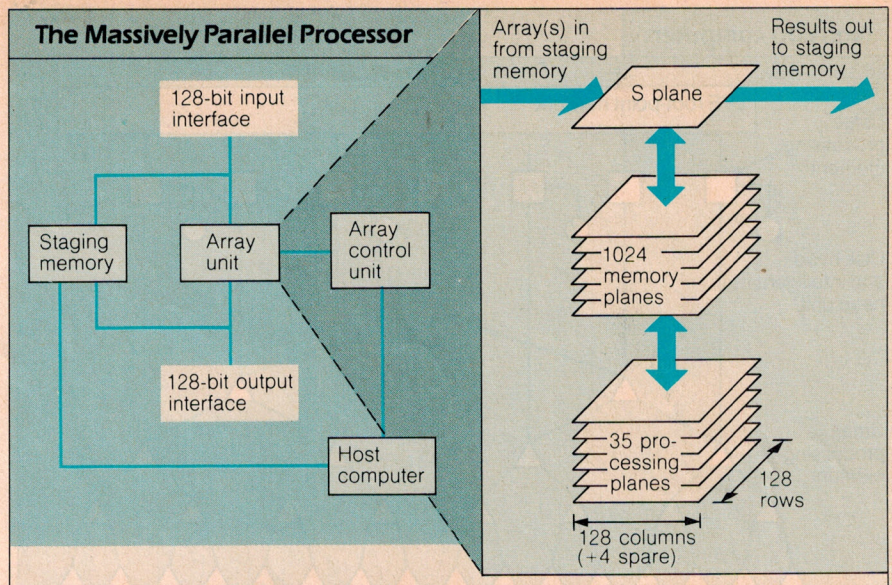
A BRIGHT IDEA? *Thinking Machines'*
Daniel Hillis designed a computer
that reconfigures itself for each task.

Competing with Cray. Conventional supercomputers are the standards against which the new highly parallel processors will be measured. The basic question is whether massively parallel processing will produce more computing power at less cost than the products of Cray and its competitors. The Cray X-MP/48, which has four high-performance processors, is generally considered the fastest of today's supercomputers—with a theoretical maximum speed of 1000 million floating-point operations per second, or MFLOPS. (In practice, such peak speeds are rarely achieved; for most applications, processing speed is about a tenth the maximum.) The X-MP series has a clock cycle of 9.5 nanoseconds.

Yet future parallel processors will be competing not with today's supercomputers but with the even faster machines now under development. The Cray 2, a four-processor machine expected to appear this year, will have a 4-nanosecond cycle and a peak performance of up to 3 billion floating-point operations per second (gigaFLOPS or GFLOPS). In 1986, ETA (St. Paul)—a spinoff of Control Data—plans to release its GF-10, an eight-processor machine with a 10-GFLOPS capability. The Cray 3, with a 1-nanosecond cycle time and computing power comparable to the GF-10, will be introduced around the same time. Future supercomputers, like their present-day progenitors, will cost about \$5–10 million each.

It seems unlikely that any of the massively parallel processors currently under construction will reach, let alone exceed, these speeds, except perhaps in limited-application areas. However, with the future 1-nanosecond machines, the supercomputer designers may be nearing the physical limits of the traditional approach. They may find it very difficult, for example, to go from 1-nanosecond cycle times to 0.1-nanosecond times. In 0.1-nanosecond an electrical signal moves only 3 centimeters—forcing an entire computer to fit within a 1-inch cube! Thus, late in this decade, highly parallel architectures could start to surpass the power of supercomputers having less than a dozen or so subunits.

SIMD is simplest. Since the SIMD approach is the simplest of the parallel designs, it should not be surprising that the first operational large-scale parallel processor was designed along these lines. The Massively Parallel Processor (MPP), built by Goodyear for NASA, is intended for work that has a high degree of inherent parallelism and "localness" (a tendency for parts of the problem to be affected only by adjacent parts). Examples are image processing and hydrodynamic modeling. In con-



Built for NASA by Goodyear Aerospace, the Massively Parallel Processor (MPP) comprises four major modules: the array unit, staging memory, array control unit, and program and data management unit (or host). The array unit consists of 16,384 processors arranged in a 128×128 array (four spare columns are available if needed), with each processor connected to its four nearest neighbors. Each processing element works on 1-bit data and is composed of 35 1-bit registers. Also associated with each processing element are 1024 bits of memory, resulting in 1024 memory planes (each 128×128 bits) for the entire two-dimensional array. Data are stored and reformatted as necessary by the staging memory and then fed over a 128-bit-wide path to the array unit's S plane (an input/output memory facility). After 128 clock cycles, the S plane is full; its entire contents can then be stored in any available memory plane in one clock cycle. Likewise, a full memory plane's contents can be shifted to the S plane in one cycle for output to the staging memory. The array control unit sends the array's processors their identical instructions and coordinates their operation. The front-end host handles user and program interactions with the system.

trast, AI applications are often characterized by interactions of distant parts of a large problem.

The MPP is based on a nearest-neighbor network—its 16,384 processors, arranged in a 128×128 square array, are each connected only to their four neighbors. The edges of the array can be connected to the opposite edges to form a cylindrical or donutlike surface.

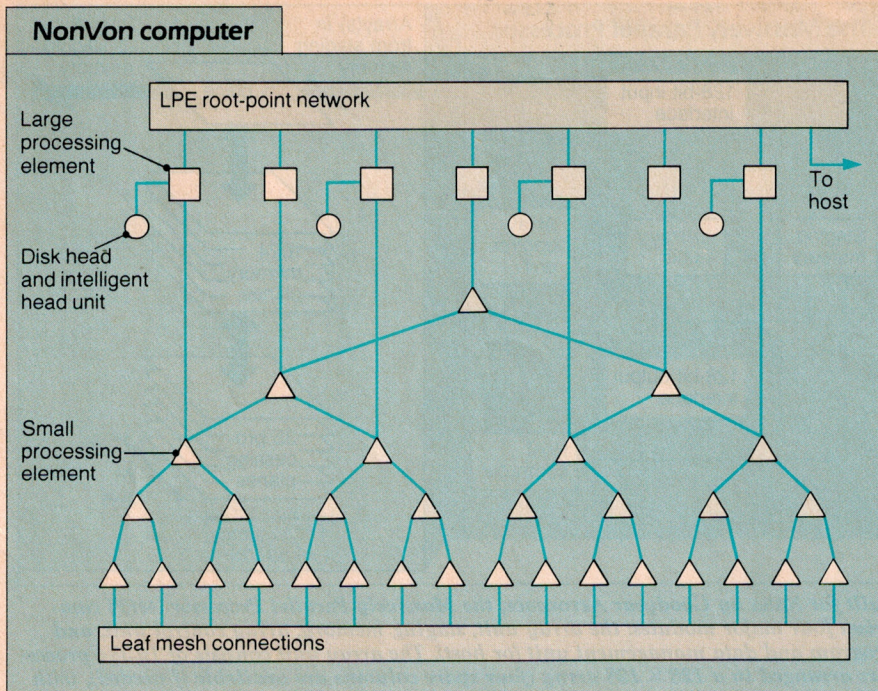
The individual processors, which are small and simple, are combined in groups of eight on custom-designed complementary metal oxide semiconductor (CMOS) VLSI chips. Each processor is attached to 1 kilobit of memory on a standard RAM chip. The processors themselves consist of an arithmetic unit, a logic unit, and a few shift registers. Operating with a 100-nanosecond step time (10 million steps per second), each processor can add, for example, some 400,000 8-bit integers a second or do 13,000 32-bit floating-point multiplications. This gives the MPP an overall maximum rate of some 200 MFLOPS, approaching that of the fastest existing conventional supercomputers. It also makes the MPP the most powerful of the available highly parallel systems.

Collectively, the processing elements make up the array unit (ARU), the main component of the computer. The ARU processors get their identical instruc-

tions from a separate "array control unit," which broadcasts the control signals and memory addresses to all processing elements. This unit also controls input/output operations and performs scalar operations that don't require the computing power of the array. A conventional DEC VAX-11/780 acts as the front-end interface for the whole machine, accepting programs and data from the users and outputting the MPP's final results.

The array continuously performs simultaneous operations on a plane of 16,384 bits. For some applications, the performance of the MPP has been impressive. For example, in a single operation the MPP groups pixels from Landsat images according to their spectral characteristics—in effect, dividing up an image into many segments with approximately constant characteristics. On a VAX-11/780, this grouping algorithm took seven hours, while on the MPP it took only 18 seconds, or less than a thousandth as long. Since current supercomputers are considered to be one or two hundred times as fast as a VAX, the limited-application MPP appears to be 5–10 times as fast as existing supercomputers, even though its clock rate is ten times slower.

In theory, the MPP could be scaled up to have 100 times as many processors,



The NonVon computer uses both large processing elements (LPEs) and small processing elements (SPEs). The LPEs connect to each other over a high-bandwidth root-point network and connect to SPEs via treelike networks. Every LPE and SPE has its own local memory. SPEs at the bottom of the trees are connected to their nearest neighbor SPEs. Still a prototype, NonVon will theoretically grow to 8000 processors.

providing enormously faster speeds than conventional supercomputers. For many applications that can live with nearest-neighbor connections, such as fluid simulations, some computer vision applications, and image processing, the MPP's simple design may prove a significant competitor. But many applications aren't suited to the limitations imposed by this architecture.

Cellular automata. The Connection Machine, now under construction by Thinking Machines, tries to overcome these limitations by setting up a network of simple processors that can be easily reconfigured to reflect the actual structure of a given problem. The machine was originally designed for artificial intelligence applications, in which large bodies of information, such as the rules for a sophisticated expert system, have to be searched repeatedly and rapidly. (Thinking Machines now believes

that the Connection Machine will prove equally effective in processing many types of scientific problems.)

In AI problems each piece of information can be represented as a node in a network, with the relations among the pieces represented by connections between the nodes. The Connection Machine tries to embody this structure in hardware. Each tiny processor has associated with it a small piece of memory, and each processor can form connections with any other one through the physical (hardwired) network. In this system a search is, in effect, the process by which connections are formed—that is, the process by which one cell finds the address or location of another cell.

For example, suppose cell RED wants to call cell BLUE. RED sends out a message wave that propagates to all cells in the network. When the wave, which contains RED's address, reaches BLUE,

the BLUE cell sends its own address back to RED. The RED cell then cancels the still-spreading message wave with a faster cancel wave. The cancel wave travels faster than the message wave because the machine uses built-in delays to slow the initial message wave.

Messages are transferred from cell to cell, and each cell alters the instructions appropriately as it passes them along. Thus a message with an address "up 2, across 5" could be given to the cell above, with the address altered to "up 1, across 5." And since pieces of information can be moved from cell to cell, those that are closely related logically can be stored in neighboring cells, thus minimizing delays.

The machine is based on the concept of cellular automata, tiny entities that communicate with their neighbors in some prescribed fashion and that have become familiar to computer users through such games as "Life." Says Daniel Hillis, the machine's principal designer, "The main advantage of the Connection Machine is its flexibility, the way in which the networks can be made to reflect the problem."

The machine under construction, to be completed this year, will contain 64,000 small processors. Even though its cycle time will be only 1 microsecond, or 10 times slower than the MPP, the Connection Machine will be similar in performance to the Goodyear machine—about 10 billion 8-bit integer operations per second. It contains about 1000 custom VLSI circuits with nearly 100,000 gates each.

The physical interconnection of the network is a variant of the "n-cube," or "hypercube," in which each processor is physically joined to a given number (n) of nearest neighbors. However, the path of data through the network is altered by the address-to-address connection formed during the running of a program, so that a "virtual network" of arbitrary topology can be constructed.

The Connection Machine is not itself a complete computer; it needs a conventional front end to feed it a serial stream of instructions. As a result, it is a SIMD machine, carrying out single instructions on a large body of data through parallel processing.

The basic idea of embedding tiny processors in a memory, in effect creating an intelligent memory, is not unique to the Connection Machine. A related idea is embodied in the NonVon, under development at Columbia University. NonVon, like the Connection Machine, is composed of several thousand (and ultimately a million or more) simple processors, each with a tiny memory.

The structure of NonVon, though, is quite different, allowing the currently planned version to run MIMD programs

Representative parallel computers

Machine	Developer	Number of processors	Cycle time (nanoseconds)	Maximum speed	Memory	Year operational
MPP	Goodyear Aerospace	16,384	100	6.5 BIPS ¹	Dist.	1983
Connection	Thinking Machines	64,000	1000	10 BIPS	Dist.	1985
NonVon	New York University	8000	1500	16 BIPS	Dist.	?
iPSC	Intel	32-128	100	2-8 MFLOPS ²	Dist.	1985
Butterfly	Bolt Beranek and Newman	to 128	?	to 200 MIPS ³	Shared	1984
Sigma-1	Japan's National Electrotechnical Lab	256	100	100 MFLOPS	Dist.	1986
Cedar	University of Illinois	32	100	10 MFLOPS	Shared	1985

¹Billion instructions per second

²Million floating-point operations per second

³Million instructions per second

as well as SIMD. NonVon uses two types of processing units—large and small. The large processing elements (LPEs), each of which has its own memory, connect to each other by a high-bandwidth two-stage interconnection called a root point network. Each LPE sends out a stream of commands to a tree of small processing elements (SPEs), each with its own, smaller memory. The SPEs at the bottom of the tree are also interconnected in a nearest-neighbor network. The LPEs in the current design are 32-bit microprocessors (Motorola 68020s), while the SPEs are implemented on custom VLSI chips.

Like the Connection Machine, NonVon is aimed primarily at AI applications that rely on associative memories capable of being searched rapidly. An associative memory is one in which the entries are addressed by their contents, rather than by a specified location. For example, it can search for all entries containing the word *red* without examining each entry in turn. In such a search, a query from an LPE can be processed simultaneously by large numbers of SPEs, with the replies returning within a few machine cycles. NonVon's LPE-tree-SPE structure allows multiple searches to go on simultaneously in different parts of the machine, even though each SPE remains permanently linked to a single LPE.

NonVon is not as far along as the Connection Machine. Last December a 64-processor prototype began running. According to David Shaw, NonVon's chief designer, its capability is 40 million instructions per second (MIPS). "The next step is to build an 8000-processor machine with a rate of 16 billion instructions per second," he says.

Big is beautiful? MPP, the Connection Machine, and NonVon are all small-grain machines in that they aim to incorporate a million or more processors, thus breaking down problems into tiny pieces (a NonVon SPE, for example, has only a quarter kilobyte of memory). The small-grain approach targets problems with an extreme degree of parallelism. Most other parallel designs use a large- to medium-grain architecture, with prototypes having a few dozen to a few hundred processors and with ultimate goals of a few thousand.

Intel's iPSC computer, the first parallel processor to be widely marketed, is a leading example of this latter approach. The iPSC is being sold in three sizes, with 32, 64, and 128 processors each. The computer is broadly based on Caltech's Cosmic Cube machine, developed by Charles L. Seitz and Geoffrey C. Fox. (The Cosmic Cube was among the first machines to use the hypercube connection, which has since been incorporated in modified form by such com-



KEITH SHELTON

HORIZONTAL PROCESSING. Ed Slaughter, general manager of the Intel Development Operation, believes the iPSC computers will give parallel processing credibility.

puters as the Connection Machine.)

Compared with the processors of small-grain machines, the individual Intel processors are much more powerful. Each single-board processing unit contains an 80286 CPU, an 80287 numerical processor, 512 kilobytes of CMOS dynamic RAM, and 64 kilobytes of ROM. Memory capacity alone is roughly 1000 times greater than for the small-grain processors.

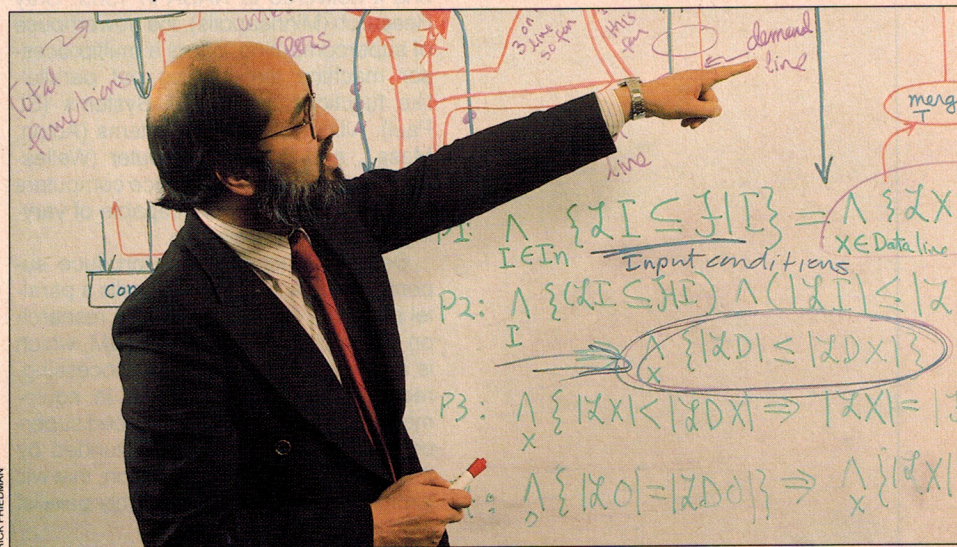
This larger memory is needed for the iPSC's main target applications—physical-science simulations. These currently depend heavily on floating-point operations, in contrast to AI symbolic applications. A single floating-point number requires 64 bits of storage, and the multiplication of long strings of such numbers requires considerably more. AI applications rarely call for numbers with many decimal places—8-bit integers are generally enough.

Interestingly, the dependence of scientific applications on floating-point

operations may diminish as small-grain parallel machines arrive on the scene. With such massively parallel computers, the detail now provided by very large or very small floating-point numbers could instead be provided by breaking the problem into tiny pieces, explains Jim Bailey, director of marketing at Thinking Machines. "With parallel machines, the scientists will become increasingly involved with the relationship between the number of processing cells in their applications and the number of interesting elements that they're trying to understand," he says. "And as the ratio between the number of cells and the number of interesting things under study gets closer to 1:1, the capability provided by floating point becomes less of a requirement."

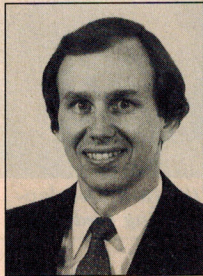
Floating point will still be necessary for simulations in which there are far fewer "interesting" items than processing cells, says Bailey. For example, although a computer model of the planets

GOING WITH THE FLOW. MIT's Arvind says the dataflow technique can coordinate the numerous processors in parallel machines and can simplify programming.



RICK FRIEDMAN

Parallel machines take on supercomputers



Schmidt



Gaertner

"The acceptance of parallel computers rests in part on the development of appropriate software. For this to occur, programmers must undergo a cultural transition."

Gary Schmidt
Manager, Parallel
Processing Applications
Bolt Beranek and Newman

"Eventually, every customer for Cray supercomputers will be a customer for a Cray parallel processing machine."

Robert Gaertner
VP, Communications
Cray Research

Parallel processors—computers able to work on different parts of a problem concurrently—are appearing in the marketplace. They don't yet have the power or speed of the most advanced supercomputers working at peak operations, but they portend significant competition because of their potential for handling problems too complex for conventional machines.

The supercomputer market suggests the demand that could develop for parallel processors. Expected supercomputer sales for 1985 are \$425 million, with revenue growth of 30% a year anticipated to yield \$1.5 billion by 1990, says Jeffry Canin, senior technology analyst at Hambrecht & Quist (San Francisco).

A number of companies have produced, or plan to introduce, parallel processors available for purchase. Industry observers differ about who is selling "true" parallel machines, but a reasonably wide net would take in Intel Scientific Computers (Beaverton, Ore.), Denelcor (Aurora, Colo.), Floating Point Systems (Beaverton, Ore.), ELXSI (San Jose), and Thinking Machines (Cambridge, Mass.), which plans to deliver a product this year funded in part by a \$3 million contract from the Pentagon's Defense Advanced Research Projects Agency (DARPA). Bolt Beranek and Newman (Cambridge, Mass.), which was awarded a \$20 million contract from DARPA, is also seriously considering putting a machine into commercial production, and Goodyear Aerospace (Akron, Ohio) is open to negotiated sales for a parallel computer similar to the one it delivered to NASA in 1983. Cray Research (Minneapolis), the powerhouse of supercomputing, offers a multiprocessor machine that carries out parallel-like functions, while ETA Systems (St. Paul), Alliant Computer Systems (Acton, Mass.), and Encore Computer (Wellesley, Mass.) intend to introduce computers over the next two years capable of varying degrees of parallelism.

Several Japanese firms produce supercomputers, but none yet sells a parallel machine, although intensive research on this subject is under way. IBM, which is also investigating parallel processing, recently donated \$30 million in equipment and services to an advanced supercomputing center at Cornell, funded by the National Science Foundation, that will explore the capabilities of highly parallel machines.

Two forces are prompting the commercial development of parallel processors. "We are running up against fundamental physical barriers in designing von Neumann machines," points out David Shaw, director of the NonVon Supercomputer Project at Columbia University (New York). "Traditional supercomputer architects are now dealing with speed-of-light limitations and exotic cooling technologies, but their efforts are reaching the point of diminishing returns." And Roy Coppinger, marketing manager for Intel, says "both the scientific community and corporate R&D departments worry about what to do when the problems they deal with become too large for conventional supercomputers. These people need greater computational power at an affordable cost."

Initially, parallel computers will be applied to simulations and other types of modeling problems. According to Jim Bailey, marketing manager at Thinking Machines, many such applications—for example, electronic circuit design, molecular modeling, oil reservoir modeling, and airflow analysis—consist of large numbers of small elements simultaneously in flux. Although usually forced into the mold of sequential operations, these problems are particularly amenable to processing that more closely matches their inherent parallelism. Parallel computers could yield substantial economic payoffs by reducing the time needed to design and test a product or manufacturing process, such as a new drug, airplane, or automobile.

In addition, says Bailey, there are "a whole range of applications that can't be handled easily or at all by existing supercomputers that are ideal for parallel processors." These include image processing, many aspects of artificial intelligence, and the autonomous land vehicle and naval battle management system being developed under DARPA's Strategic Computing Initiative.

Vendors of parallel computers invariably cite the relatively low costs of their machines. The average price of a supercomputer is \$5-10 million; most parallel models, albeit with considerably less power, cost under \$1 million. As the technology matures, says Thinking Machines' Bailey, parallel computers will take over "a large portion of the expanding supercomputer market."

—Dennis Livingston

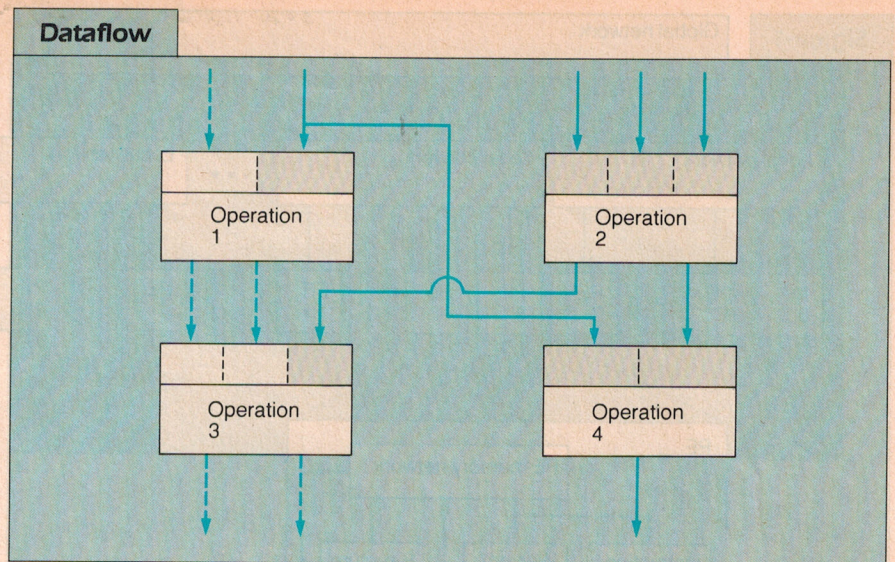
might require a billion cells to convey the size of the solar system, this huge model contains only nine interesting objects. But even for floating-point applications, he says, "I think we will find that the higher the granularity, the more appropriate the computer is for scientific applications."

Indeed, says Bailey, assigning a multithousand-cell problem to a single-CPU computer is like trying to keep thousands of corks underwater with a large sledge hammer. "In theory, if you just make that sledge hammer big enough, you can hit enough corks at the same time and drive them down far enough so you can get back to them before they bob up again. But that's a very complex algorithm. Having thousands of little hammers each hitting one cork at a time (à la small-grain parallel computers) keeps them all underwater in a much simpler fashion." Despite Bailey's enthusiasm for the small-grain approach, however, many other developers maintain that for scientific applications a large- to medium-grain architecture that employs floating-point operations will prove most successful.

Intel's iPSC, while having a relatively small number of processors, is like many of the small-grain computers in that it employs a distributed memory; each processor has its own memory and can therefore act independently on one part of the problem. How the problem is divided is determined by the programmer, not by the compiler or the computer itself. "We found that for most scientific problems, the best way to break up the problem has already been worked out," says Caltech's Seitz, "so it makes sense to have the programmer directly allocate the parts."

In the iPSC, as in the Cosmic Cube, once a part of the problem or process is allocated to a processor, it stays there. For example, in an astro- or plasma physics simulation, each processor of a 64-processor machine may be assigned one grid block in a $4 \times 4 \times 4$ grid, in turn breaking it down into finer points. During each cycle, each processor would take the field in its region, calculate the motion of the particles, and then calculate the contribution of its particles to the fields of all other grid blocks. It would then send the appropriate field contributions to each of the 63 other processors, while receiving from them their field contributions. Finally, each processor would combine the various contributions to get a new field for its region, to start the process again.

Intel claims that its new computers, sold for \$225,000-\$520,000, are six to 24 times as fast as a VAX-11/780 (of similar price) and a tenth as fast as a Cray 1. The 128-processor model is rated at 8 MFLOPS, or 100 MIPS.



Dataflow architectures permit asynchronous operations to fire whenever the required data (arrows) become available. The operations' output can then trigger other operations, permitting a controlled flow of processing. Here operation 1 has yet to output two data tokens required by operation 3. Operations 2 and 4 have fired, however, because all the necessary data tokens have been provided.

Sharing memory. Intel says a number of AI researchers have expressed interest in its machine, but a system with a relatively small number of separate memories does not lend itself easily to the sort of rapid searches required in AI. One solution to this problem is to use shared memories, in which the processors collectively access a single pool of memory units. The Butterfly machine from BB&N—similar to Intel's machine in other respects—uses this shared memory technique.

The Butterfly has been quietly sold to a few research groups during the past year. This May, BB&N made its first public announcement of the machine. In capability the Butterfly fits into the same niche as the iPSC, having a variable number of processors with a current maximum of 128. Each processor has access to half a megabyte of shared memory. Top machine speed is quoted at 200 MIPS, and the price is reportedly competitive with VAX's.

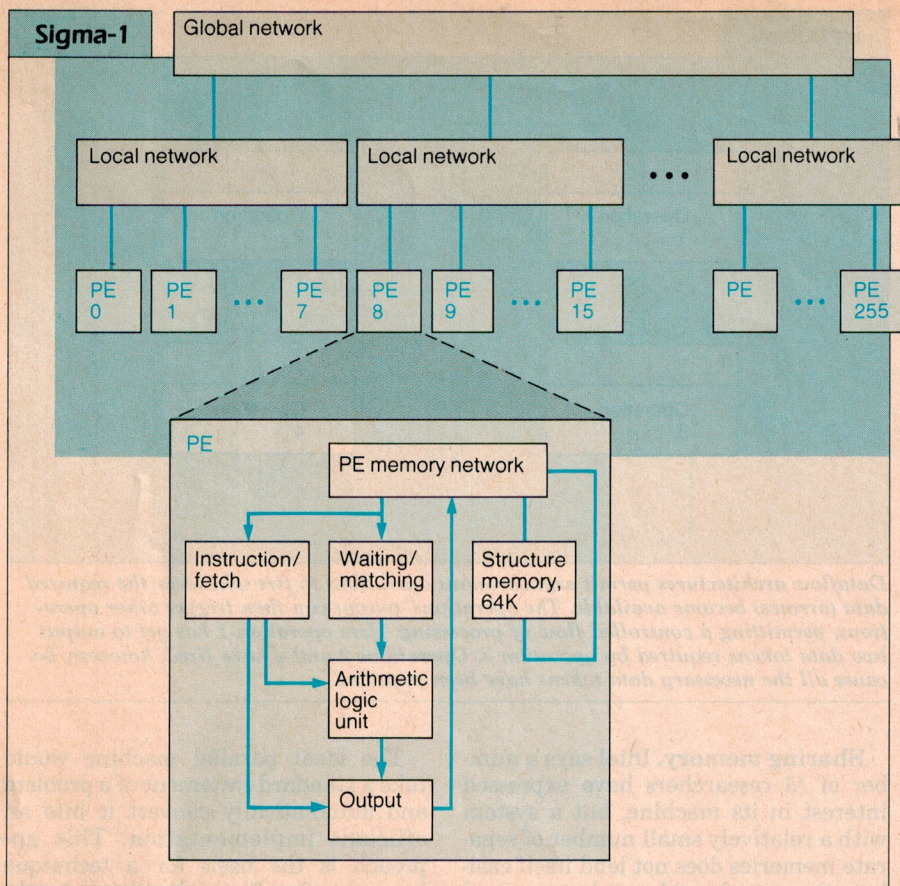
Yet it is not a direct competitor to the iPSC. With its use of shared memory, the Butterfly is aimed at the AI rather than scientific processing markets. In fact, current versions have no floating-point hardware.

Dataflow eases programming. Even if computers such as Intel's live up to their claims, traditional processors like the VAX will not necessarily lose their market. The iPSC, like all the parallel processors described so far, requires that programmers decide how a process is to be split into concurrent parts. This can make programming complex problems an even bigger headache than normal. By contrast, the sequential VAX series can run conventionally programmed problems.

The ideal parallel machine would take a standard statement of a problem and automatically convert it into an efficient implementation. This approach is the basis for a technique known as dataflow, wherein each processor receives tokens, or data packages, from other processors. When a processor has the data it needs, it performs a calculation and sends the results on to another processor. Each processor consists of a storage area with instructions on what to do with the data, a processing unit, and communication ports.

In the original implementations of dataflow machines—typified by the work of Dr. Arvind, an associate professor at MIT—virtually all of the division of a problem among the processors is done by the machine itself. A programmer uses a special language that essentially describes the problem as a map, showing the flow of the data and how they are manipulated at each point. The machine then allocates instructions among the processors, which work as data come to them.

Arvind's group is working on a software package that will emulate a prototype dataflow machine. However, a number of full-scale dataflow machines are under construction in Japan. The Sigma-1, now being developed at Japan's National Electrotechnical Laboratory, is directly based on Arvind's designs and is furthest toward completion. It is scheduled to be operational by mid-1986. The Sigma-1 will use 256 processing elements with a 100-nanosecond step time to achieve 100-MFLOPS performance on scientific problems. The processing elements are grouped into 32 subunits and connected in a



Japan's Sigma-1 will use dataflow concepts to manage the parallel operation of its 256 processors. Groups of eight processing elements (PEs) are linked via local networks, which are in turn connected by a global network. Each PE (expanded diagram) consists of a waiting and matching element that accepts data from other PEs. When all the appropriate data are available, the PE's arithmetic logic unit performs the appropriate calculation, under control of the instruction unit, and outputs the results. These travel to the next designated PE, which uses them for its own calculations.

multilevel communication network.

The Sigma-1 will be a good test of what many consider the key problem of dataflow designs: how to handle large arrays of data, a critical need in scientific applications. Difficulties arise because dataflow uses concurrency on a fine-grain level—the level of individual instructions—so data have to be moved about considerably. If entire arrays must be moved, the result is an intolerable communications overhead. If arrays are broken up, however, it is hard to put them back together. Arvind's basic solution is to leave the arrays in one place, forming a memory that is shared by all processors. This is the solution implemented in the Sigma-1, but a complex communications structure is required to keep the message flow organized and efficient.

Is it possible to strike a balance between dataflow and non-dataflow designs? Yes, according to University of Illinois researchers working on a computer project called Cedar. As in non-dataflow machines, the Cedar programmer must decide how the problems are to be divided into subtasks. But because there will be a queue of subtasks that

are automatically distributed among the processors finishing their previous jobs, the programmer will not have to decide in advance the exact sequence of processing. Once inside a given processor, the chunk of the problem is handled by sequential processing, with each processor accessing a shared memory. The prototype under construction, a 32-processor machine, will have a performance of 5–10 MFLOPS.

And a cast of thousands. The machines described here are simply those closest to realization; dozens of other ideas are being worked on in university, industry, and government labs throughout the U.S., Europe, and Japan. The Reduction Machine of Gyula Mago at the University of North Carolina, for example, aims to use millions of small processors arranged in a treelike structure to carry out operations in a language called FP, developed by computer pioneer John Backus. There is also work on various types of hybrid machines, such as the Rediflow machine of Robert Keller at the University of Utah, a machine that combines reduction and dataflow ideas.

Equally significant, researchers in

some AI fields such as machine vision are discovering that they don't have to buy expensive parallel computers to get the benefits of concurrency for their applications. In many cases, special-purpose processors can do the job just as fast and at a fraction of the cost. In machine vision, the most computationally intensive processes are often simple algorithms, such as edge finding, performed on all the pixels of an image. Custom VLSI chips and custom array processors can be built to expedite such operations. For example, Robert Berger and his colleagues at Carnegie-Mellon developed a special-purpose image processor with 64 subprocessors to perform a number of elementary procedures on images. It operates at a rate of 10 million pixels per second—far faster than any general-purpose computer.

The fates of the various parallel processing architectures will become much clearer during the next 18 months as researchers actually start to use and compare the machines. The Intel iPSC machine will give many users their first parallel programming experience; some researchers may then start replacing VAXs with Intel's product. Comparison between the iPSC and the Butterfly machine will give an indication of the relative merits of shared versus distributed memory. Moreover, the Cedar prototypes and the Sigma-1 will start to test the value of dataflow and the strengths of different implementations of this approach for easing programming problems. Finally, the Connection Machine will test the applicability of small-grain concurrency for both AI and scientific applications.

It is quite possible—perhaps likely—that no clear winner will emerge from all this activity. Vision researchers with a firm idea of what they need may find custom VLSI chips the best solution. AI users could wind up liking small-grain designs such as the Connection Machine or NonVon. Scientific simulation users may go with medium-to large-grain designs like the iPSC, although small-grain machines may also succeed in this market by reducing the need for floating-point operations. Those who need easy programming may opt for dataflow machines. At any rate, by 1987 there will be a wide range of choices. And by the end of the decade, successors to the current parallel machines may be giving Cray and the other traditional supercomputer vendors a run for their money. □

Eric J. Lerner is a freelance journalist who writes extensively on computer science and electronics.

For further information see **RESOURCES** on page 72.

RESOURCES

Following are sources for further information about topics covered in the feature articles in this issue.

Parallel processing, p. 20

- "Perspectives on supercomputing." B. L. Buzbee & D. H. Sharp. *Science*, Feb. 8, 1985. Includes an evaluation of the impact parallel architectures may have in boosting performance.
- "Reinventing the computer." Tom Alexander. *Fortune*, March 5, 1984. An overview of the supercomputer market, with a good explanation of the need for parallelism.
- "NSA seeks super parallel processors." Karen Berney. *Electronics Week*, Dec. 17, 1984. Describes the National Security Agency's plans to build a \$12 million research center for developing massively parallel computers.
- "Supercomputers: a strategic imperative?" Dwight B. Davis. *High Technology*, May 1984. Covers the plans of the traditional supercomputer vendors, including their ambivalence toward parallelism.
- "Advanced parallel architectures get attention as way to faster computing." Tom Manuel. *Electronics*, June 16, 1983. Dated, but provides good explanations of several still-current approaches to parallelism.

Cold war in the ocean depths, p. 29

- "Sound searching the murky seas." David Ritchie. *High Technology*, March 1983. Review of sonar surveillance technology.
- "Understanding anti-submarine warfare technology." Randolph J. Steer. *Review of U.S. Military Research and Development 1984*, Kosta Tsipis & Penny Janeway, eds. Elmsford, NY: Pergamon-Brassey's, 1984.
- "Anti-sub warfare escalates." James B. Schultz. *Defense Electronics*, June 1983. Review of U.S. ASW systems and manufacturers.
- "The silent chase: tracking Soviet submarines." Thomas B. Allen & Norman Polmar. *New York Times Magazine*, Jan. 1, 1984.
- "A quiet revolution." Lt. Com. Ralph E. Chatham. *U.S. Naval Institute Proceedings*, Jan. 1984. Implications of quieter Soviet subs.
- "Navy weaves perilous web for Soviet subs." *Defense Week*, July 30, 1984. U.S. Navy research on fiber optics for cabling and sensors.
- "Antisubmarine warfare in the nuclear age." Donald C. Daniel. *Orbis*, fall

1984. Detailed discussion of nonacoustic ASW.
- "Penetrating the sea sanctuary." Edgar Ulsamer. *Air Force*, Sept. 1984. Use of synthetic-aperture radar for ASW.
- "Transparency: impossible or inevitable?" Richard Wohl. *Defense Science 2002+*, Feb. 1984. Prospects for wide-area ASW.
- "Will strategic submarines be vulnerable?" Richard L. Garwin. *International Security*, fall 1983. Trends in strategic ASW.

National labs, p. 39

Technology transfer officials

- Brian Frost, Argonne National Laboratory, 9700 South Cass Ave., Argonne, IL 60439, (312) 972-4929.
- Jane M. Welch, Idaho National Engineering Laboratory, 550 2nd St., Idaho Falls, ID 83401, (208) 526-8318.
- Robert J. Morris, Lawrence Berkeley Laboratory, Berkeley, CA 94720, (415) 254-1818.
- Ray McClure, Lawrence Livermore National Laboratory, Box 808, Livermore, CA 94550, (415) 422-7600.
- Eugene Stark, Los Alamos National Laboratory, Box 1663, Los Alamos, NM 87545, (505) 667-4960.
- Donald W. Jared, Oak Ridge National Laboratory, Box X, Oak Ridge, TN 37831, (615) 574-4193.
- Loren C. Schmid, Pacific Northwest Laboratory, Box 999, Richland, WA 99352, (509) 375-2559.
- Robert P. Stromberg, Sandia National Laboratories, Box 5800, Albuquerque, NM 87185, (505) 844-5535.

References

- Report of the White House Science Council: Federal Laboratory Review Panel*. May 1983. NTIS Publication PB83-255620. National Technical Information Service, 5285 Port Royal Rd., Springfield, VA 22161. This is the "Packard report," which spurred the labs to increase their work with industry. \$8.50.
- Progress Report on Implementing the Recommendations of the White House Science Council's Federal Laboratory Review Panel*. July 1984. Office of Science and Technology Policy, Executive Office of the President, Wash., DC 20506. Two-volume report—volume one is a summary report, and volume two is a

status report by agency.

Development and Diffusion of Commercial Technologies: Should the Federal Government Redefine its Role? March 1984. Staff memorandum, Office of Technology Assessment, (202) 224-8996.

Probable Levels of R&D Expenditures in 1985: Forecast and Analysis. Columbus, OH: Battelle Memorial Institute, Dec. 1984. A 20-page booklet shows how much federal money goes into industrial R&D.

"Government research at your fingertips." *High Technology*, Oct. 1983. Federal Laboratory Consortium.

Preventing midair collisions, p. 48

Minimum Operational Performance Standards for Threat-alert/Collision Avoidance System. #RTCA/DO-185, vols. 1 & 2. Radio Technical Commission for Aeronautics, 1425 K St., NW, Suite 500, Wash., DC 20005.

System Safety Study of Minimum T/CAS-2. Dept. of Transportation/Federal Aviation Administration. DOT/FAA/PM-83/36. FAA Public Information Office, 800 Independence Ave., Wash., DC 20591.

"Automation in the skies." David Allison. *High Technology*, Nov./Dec. 1982.

"Automating air-traffic control." H. Toong & A. Gupta. *Technology Review*, April 1982. Covers rehosting computer project.

"Piedmont 727 will test threat alert/collision avoidance device." Philip J. Klass. *Aviation Week*, Nov. 12, 1984.

"FAA alters its collision avoidance plan." Philip J. Klass. *Aviation Week*, Nov. 15, 1982.

Platinum, p. 54

"The miracle metal platinum." *National Geographic*, Nov. 1983. A popular account of platinum's history and uses.

"Platinum complexes of vitamin C show anticancer potential." *Chemical & Engineering News*, Sept. 17, 1984. Technical overview.

"Cisplatin and cancer." *Journal of Chemical Education*, Dec. 1977. An account of how the first platinum drug was developed.

"Platinum group metal catalysts in industrial organic synthesis." *Chemistry in Britain*, April 1984. Platinum catalysis for the technically minded.