SCIENCE

Multis: A New Class of Multiprocessor Computers

C. Gordon Bell

Multis: A New Class of Multiprocessor Computers

C. Gordon Bell

Computers so closely reflect electronic technology that the four generations of computers are named, accordingly, the vacuum tube, transistor, integrated circuit, and microprocessor generations (1). Typically, one technology is used until the limit of work it can handle is reached or another technology supersedes it. For example, integrated circuits contain as many as 2000 transistors on a single silicon chip; and microprocessors contain large-scale and very large scale integrated (LSI and VLSI) circuits with as many as 2 million transistors per silicon chip. Each technology has its own characteristic cost, speed, power dissipation, packing density, and reliability; and the different ways in which the technologies have been applied have resulted in a variety of computer classes based on price (Fig. 1).

When a new technology becomes available, there are usually three ways to make use of it, two of which generate new computer classes:

1) The technology can be used to increase the performance of an existing class of computers while maintaining the cost and selling price, thereby increasing the effectiveness for current users. For example, mainframe computers were established in 1950 with the Univac I, which cost between $300,000 and $5 million. The best known family of mainframes was introduced by IBM in 1964 with the System 360, which was based on transistor technology. In the early 1970's IBM introduced the 370 series, which used integrated circuits; and most recently the 43xx–380x series, which makes use of high-performance integrated circuits, was introduced.

2) A new, lower cost class of computers with the same performance as a previous computer can be produced, which will result in new applications for computers. Minicomputers and personal computers are the best known products of this design path. The first minicomputer, the PDP-8, which was the result of second-generation technology, was introduced in 1965 by Digital Equipment Corporation. By 1972, 91 companies had formed to build minicomputers with the third-generation technology, integrated circuits (2). Minicomputers typically sell for $10,000 to $100,000 and superminicomputers sell for $100,000 to $500,000.

3) A new computer class can be produced by combining parts in a new way. The supercomputer and various microprocessor-based computers including workstations and multis emerged in this fashion. In 1964, the supercomputer class was introduced with the CDC-6600, although large computers, including IBM's Stretch, had been built earlier. Seymour Cray's CDC-6600 contained about a half-million densely packed, Freon-cooled transistors connected with discrete circuits. Since the CDC-6600, Cray has designed nearly all of the supercomputers, which have made use of various forms of parallel computation to execute a single instruction. In Cray's latest designs, speed is obtained by processing vectors at over 500 million floating-point operations per second. Cray's supercomputers sell for $4 million to $20 million.

In 1971, a single chip processor, the Intel 4004 microprocessor, was introduced and used in a wide range of applications, from calculators to controllers for microwave ovens. In 1975, Altair introduced the first home computer, based on Intel's 8080 microprocessor. Today, microprocessors have the features necessary for building high-speed computers with virtual memory that compare favorably with minicomputers. This high-performance component of negligible cost has permitted the introduction of many new classes of comput-

ers (2), including lap and home computers ($200 to $1500), personal computers ($1000 to $10,000), multiple-user (shared) microcomputers ($6000 to $20,000), workstations ($10,000 to $60,000), and microprocessor-based multiprocessors, or multis ($20,000 to $500,000). It is with this newest class of computers, the multis, that we will be concerned.

Multiprocessors, Multicomputers, and Multis

Multiprocessors are computers that contain two or more processors capable of independently executing instructions and gaining access to programs and data held in a common memory. A processor is the part of the computer that carries out computational work by retrieving instructions from memory and performing operations on data that were also retrieved from memory. Thus information in a multiprocessor can be freely used by and exchanged among all processors.

In contrast, multicomputers consist of interconnected, independent computers, each of which has a processor, that communicate by passing messages through fixed links or a switch such as a local-area network. Intel recently introduced a series of multicomputers, called the Intel iPSC Family of Concurrent Supercomputers, which have 32, 64, or 128 computers connected in a hypercube configuration. Within a model, each computer is connected to five, six, or seven other computers through message-passing links that can transfer data at a rate of 10 megabits per second.

Since a multiprocessor can be partitioned into independent computers and use common memory for intercommunication, it is a more general machine than a multicomputer and, indeed, can simulate one. Although all mainframe manufacturers sold multiprocessors with two to four processors before the fourth generation of computers was developed, the multiprocessor structure has not been used in the low-cost classes of computers.

Multiprocessors are generally symmetric; that is, any processor in them can operate on any job within the memory. The Burroughs B5000, introduced in 1961, was a dual symmetric multiprocessor (3). Some multiprocessors, however, are asymmetric; that is, one master processor performs the operating system functions and some applications work while all the other processors only do applications work. The disadvantage of

asymmetric processing is that all the operating-system functions are performed by a single processor, which can create a bottleneck in that processor. IBM developed both symmetric and asymmetric dual processors, including the System 370 and System 308X series computers (4). The IBM Attached Processor Scheme, however, is asymmetric and, in addition, requires the master processor to perform the input-output function.

In the past, uniprocessors had a cost advantage over multiprocessors because the cost of switching and cabling is proportional to the number of processors and memories in the system (5). As a result of program sharing, a multiprocessor with N processors requires less memory than N independent computers, but the multiprocessor's memory must be faster and larger than a uniprocessor's. Unfortunately, a larger memory that is k times faster costs more than k times as much per bit.

Several multiprocessors have been built that contain a single, central switch. The advantage of a central switch is low cost, because the cost of cable is directly dependent on the number of processors and memories. Two single-switch computer systems with 16 processors have been built: one for high-performance military use (6) and the other for experimentation in parallel processing (5).

The introduction of cache memory (7) in the late 1960's led 20 years later to the design of microprocessor-based multiprocessors, or multis. Cache memory is a small, high-speed memory that stores frequently used instructions and data. Placed between the processor and the large, main memory, the cache memory

permits a trade-off between expensive high-speed memory, and large, cheap low-speed memory. Since the content of each cache is associated with a processor, extra logic must be added to the system to delete "cached" copies of data or instructions that have been modified by another processor. If a processor uses stale data, incorrect calculations can occur.

The Structure of a Multi

Multis are based on advances in microprocessor technology and the cache memory (Fig. 2). They use a single set of wires called a bus for all communication

Summary. Multis are a new class of computers based on multiple microprocessors. The small size, low cost, and high performance of microprocessors allow the design and construction of computer structures that offer significant advantages in manufacture, price-performance ratio, and reliability over traditional computer families. Currently, commercial multis consist of 4 to 28 modules, which include microprocessors, common memories, and input-output devices, all of which communicate through a single set of wires called a bus. Adding microprocessors together increases the performance of multis in direct proportion to their price and allows multis to offer a performance range that spans that of small minicomputers to mainframe computers. Multis are commercially available for applications ranging from real-time industrial control to transaction processing. Traditional batch, time-sharing, and transaction systems process a number of independent jobs that can be distributed among the microprocessors of a multi with a resulting increased throughput (number of jobs completed per unit of time). Many scientific applications (such as the solving of partial differential equations) and engineering applications (such as the checking of integrated circuit designs) are speeded up by this parallel computation; thus, multis produce results at supercomputer speed but at a fraction of the cost. Multis are likely to be the basis for the next, the fifth, generation of computers—a generation based on parallel processing.

between processors, memories, and input-output devices. This bus and computer structure was pioneered in Digital Equipment Corporation's single-processor PDP-11 Unibus, which was introduced in 1970 (8). In a multi, the cache memory associated with a processor services approximately 95 percent of the processor's requests for memory. Since only 5 percent of a processor's requests reach the common bus, the requirements for the bus bandwidth are reduced by a factor of 20. This allows up to 20 times as many processors as are available in computers without cache memories. In addition, because all processor requests for primary memory appear on the common bus, each cache can independently monitor the bus and delete any data it contains that has been modified in primary memory by another processor (9, 10).

Standard microprocessor buses such as Intel's Multibus II, Motorola's VME, Texas Instrument's Nu Bus, and the proposed IEEE 896 Futurebus provide
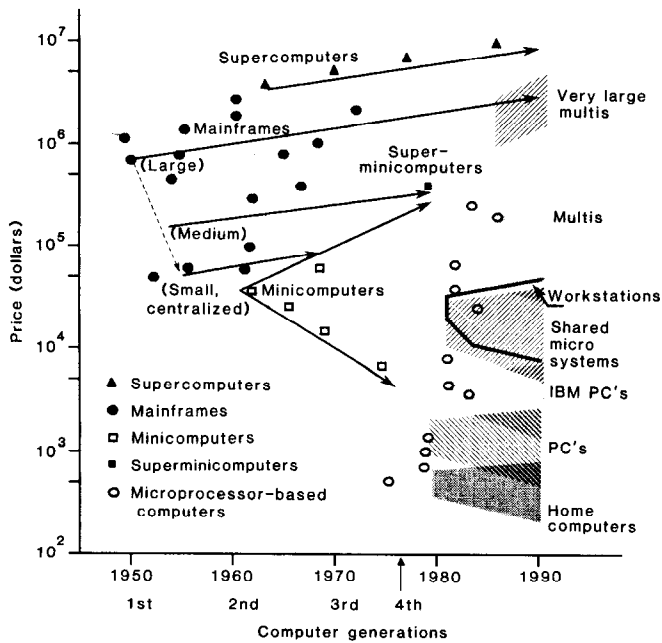
Fig. 1. The price of classes of computers plotted against time illustrates the clustering of classes such as mainframes, supercomputers, minicomputers and superminicomputers, personal and home computers, shared micros, workstations, and multis. The emergence of new classes of computers coincides with the advent of new technologies, as marked by the generations of technologies. Shaded areas identify microprocessor-based systems.

for multiple processors and operate at a rate of about 10 million transactions per second. A typical bus with time-shared address and data lines will deliver data at 20 megabytes per second or allow access to 5 million four-byte memory values per second. If separate address and 64-bit data lines are provided and simultaneous requests for memory are permitted, a large multi can be constructed to deliver data at 100 megabytes per second or allow access to 12.5 million 64-bit memory values per second.

System performance is linearly proportional to the number of processors until the bus is saturated with requests for access to the memory (11). Because of this linear proportionality, multis provide the means for producing a range of compatible computers without a specific processor design and specific technology for each model (12). With multis, a client can select the computer that meets his needs by selecting the appropriate number of processors or memories instead of having to select from different family members.

Current microprocessor-chip sets (13) such as the National 32032 and Motorola 68020 have the key characteristics of past mainframes and minicomputers including: 32-bit addressing, virtual memory control, complete instruction sets, including floating-point data, and performance levels per single-chip set that equal those of a minicomputer. The gap between mainframe-processor and microprocessor performance will continue to close owing to the difference in the rate of improvement of the underlying technologies. Minicomputers and mainframes are based on bipolar integrated circuits of transistor-transistor logic (TTL) and emitter-coupled logic (ECL), respectively. The speed of bipolar circuits has only increased by 15 percent per year over the last 10 years (2), and it has been estimated that the speed of the largest single processor will improve by only 23 percent per year to reach 80 million instructions per second by 1990 (14). Microprocessors, however, are based on metal-oxide semiconductor (MOS) technology, which has increased in internal speed by 40 percent per year for the last 10 years. The speed of TTL and MOS logic are now about the same. High-performance minicomputers use additional hardware for increased parallelism and faster floating-point arithmetic to outperform microcomputers, but the gap in their performance should close in the next few years.
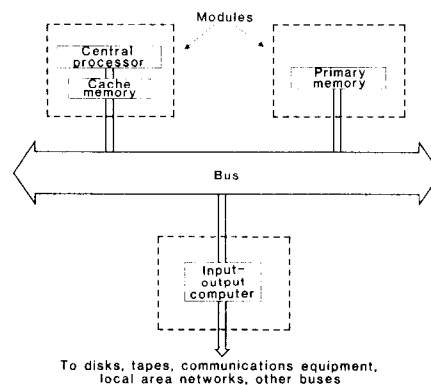
The microprocessor's small size per-

Fig. 2. Multis, multimicroprocessor computers, are organized around a single, uniform bus that connects central and input-output processors and common memory. Each processor requires a cache memory.

mits the construction of physically small multis, a key factor in their high performance. Consideration of signal propagation and packaging leads to a roughly cube-shaped system with 10 to 20 modules. The processors (both central and input-out) and memory occupy about half the volume, and the height is proportional to the bus width and, roughly, the bus-bandwidth capacity. The length (number of modules) and depth (the area of each module) of multis increase in proportion to the bus capacity. Amdahl's Rule (15) describes how memory size and input-output rate need to grow in proportion to processing rate.

Because a multi is built from a number of modules, it has an inherent redundancy. Typically, a multi consists of four module types: processor, memory, and two types of in-out controllers for disks and terminals. From these components, multis of 10 to 20 modules are built.

The inherent redundancy provides multis with greater reliability (probability that the system is operational), availability (fraction of time the system is available for use), and maintainability (the time to repair) than traditional computers of the same size. Indeed, the increase in these characteristics is greater than an order of magnitude.

All parameters can be adjusted at any time during the computer's lifetime by selecting the appropriate number of each type of module. With the appropriate software, various modules can be marked as faulty and taken out of service, which allows the system to function even in the presence of failed modules. Maintainability is increased even more by the practice of having spare modules within the individual computers. The owner should be able to maintain a multi by simply replacing faulty modules.

Manufacturing, Cost, and Longevity

Hardware for multis is inherently simple because of the few module types and the one explicitly defined interface standard, the bus. A multi consisting of 20 modules of four types can be designed with less than one-fifth the effort of a 20-module minicomputer, because in minicomputers each processor module is different and all modules communicate with one another in different ways. Conventional minicomputers require 2 or 3 years of design (16, 17).

Since the cost of product tooling is also proportional to design cost, the manufacturing cost is reduced by building only four module types, each of

464

which can be tested independently. The cost of manufacturing also decreases as more modules are manufactured, following a traditional manufacturing learning curve.

The product life of a multi is determined by the bus speed. Higher speed means greater longevity because it enables the bus to accept more and faster processors before the bus becomes saturated. The use of cache memory further increases longevity by allowing memory speed and size to be increased independently.

The Proof of the Pudding

Several multis have already been developed and introduced. Masscomp has introduced an asymmetrical dual processor for laboratory use. Unidot is offering a dual processor that uses National microprocessors connected by Intel's Multibus. Areté provides a quadruple processor that uses a proprietary, 16-megabyte-per-second bus to connect Motorola 68000 microprocessors. Sequent has introduced the Balance 8000, which can contain as many as 12 National 32032 microprocessors connected by a bus that can transmit 27 megabytes per second. Though not a multi per se, Elexsi's six-processor is a multiprocessor implemented with ECL technology and a bus that can transfer data at 320 megabytes per second.

Introduced in 1983, the Synapse "N + 1" computer is a relatively large multi that uses the Motorola 68000 microprocessor with special memory-management hardware. The Synapse structure has one more component (bus, power supply, processor, memory, in-out controller, and in-out device) than is required for operation. Designed for fault-tolerant transaction processing, the Synapse system is composed of as many as 28 arithmetic or in-out processors that communicate with as many as four memory modules for a total of 32 megabytes of memory. All modules are connected with full-duplexed buses that transfer data at 32 megabytes per second. The cache memories and main memory have control circuitry to determine the location of the "real" data when multiple processors write to the same memory location. This scheme minimizes the bus traffic caused by writing (18) compared with simple write-through schemes that always write in memory (9, 10). The processing rate is 11 to 14 million instructions per second, a rate comparable to state-of-the-art uniprocessor mainframes.

Encore's Multimax is designed for high-performance processing and high input-output data rates; it is equipped with the Nanobus, which transmits 100 megabytes per second. The Multimax permits up to 20 central processors (on ten modules) or 10 input-output computers to address 32 megabytes of memory on eight 4-megabyte modules. A single module controls requests for the bus and provides centralized services including timers, system initialization, and diagnostics. Each module, including memory, has a computer that is used for stand-alone and on-line diagnostics and maintenance. The bus and system architecture permits 1024 processors to address 4 gigabytes of physical and virtual memory. Terminal and workstation access is provided by one or more local-area networks (Ethernet or the IEEE 802.3). The operating system is UNIX 4.2 redesigned for operation on a multi.

Figure 3 compares the performance and price of two multis with various minicomputers. The multis compare favorably with the traditional approach of building a family of products from separate technologies. A single multi is inherently a family that covers a traditional product line family at every price and

Fig. 3. Performance is plotted against price for two multis (A and B) and several conventional computer families. C, D, and E are families of minicomputers; F is a family of mainframe computers. The cost-effectiveness of a multi is achieved by its use of the same components to cover a range of functions as opposed to the individual design, semiconductor technology, and packaging required for each member of a traditional computer family.

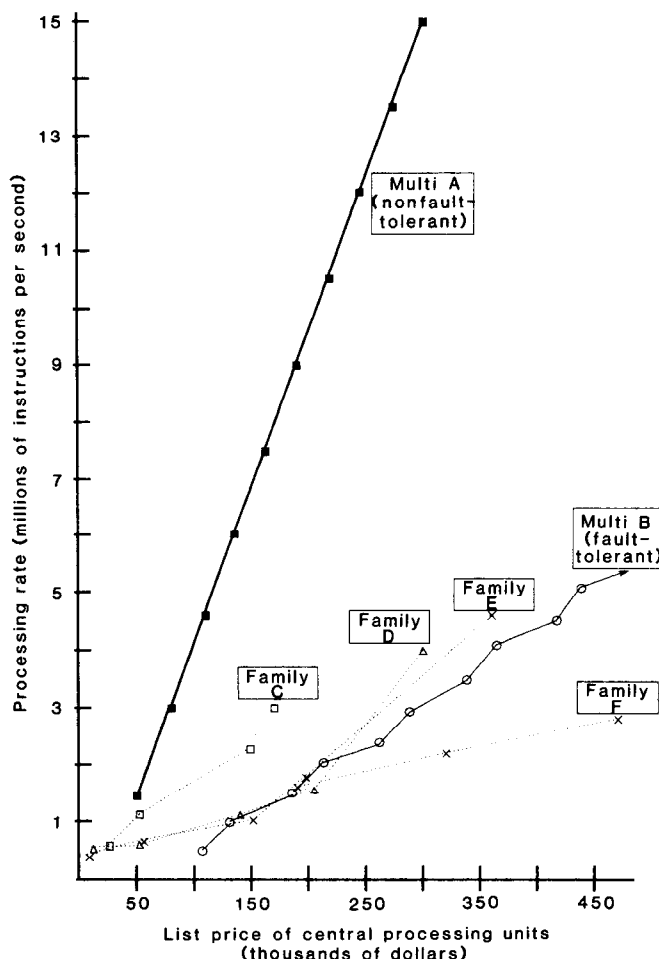performance point. Furthermore, the multi can be expanded to cover a wide range of requirements.

Work Throughput and Speedup

Two characteristics specify the performance of a multiprocessor computer: throughput and speedup. Throughput is the number of jobs completed per unit time. Speedup is the number of times faster a single job can be completed by using multiprocessors. For uniprocessors, any speedup provided by new technology translates directly into higher throughput. Since a multi can be expanded by the incremental addition of processors, both its throughput and speedup can be improved without changing the underlying technology.

Traditional uniprocessors operate on a stream of independent jobs. Dividing these jobs (19) among the independent processors in a multi results in improvement in throughput. This form of parallel processing can be applied to a number of transactions; and, furthermore, it is transparent to users.

All modern batch and time-sharing operating systems (such as UNIX) are mul-

programmed so that the uniprocessor is shared between independent jobs. A job runs to completion or until its time allocation is exhausted, whereupon the processor goes on to the next job. In a multi, each processor can be assigned to a different job, thereby exploiting the independent nature of the work load. In a multiprogramming environment, a multi can perform even more useful work because a given processor can be assigned to a job and need not be switched among all jobs, thereby decreasing overhead time (switching time).

Most commercial, on-line applications take the form of requests for information or action. Each request represents a transaction on a database. Airline-reservation systems and electronic funds-transfer systems are common examples of such transaction processing. These systems are a particular form of a multiprogrammed system in which work (the transactions) is divided into a set of jobs, each of which accomplishes a given function. The work is organized the way it would be in a job shop. Processors are assigned to independent jobs, and the work progresses from processor to processor. In many transaction-processing systems, a number of processors or computers carry out redundant computations or database transactions to insure the integrity of work.

Speeding Up a Single Job

In principle, very large multis with several hundred microprocessors could surpass the performance of supercomputers. The computational power of multis can be harnessed to speedup the time to perform single applications in three different ways: pipeline processing, concurrent processing of a data set, or general parallel processing.

Pipeline processing is accomplished by connecting a series of jobs to carry out a larger job. Operating systems such as UNIX allow a single job to be structured as a number of independent processes with interprocess communication. Data produced by one process are piped to the input of another process. As long as each process in the pipe has data to manipulate, each process can be executed in parallel. A pipelined job typically has three or four stages executing concurrently, for example: inputing a file, computing, and outputing one or two files. UNIX encourages parallel processing with this mechanism whereby various processes of a single job operate in a pipelined fashion.

The second means of speeding up a

single job is concurrent processing of a data set. The data set, such as a file, is broken into N independent parts and processed by N independent copies of the program that simultaneously execute with little or no intercommunication between the program copies. In applications that employ data-set concurrency, the speedup is linearly proportioned to the number of processors used. This proportionality has been demonstrated using Cm*, a 50-processor multiprocessor, for a number of computationally intensive jobs, including the checking of VLSI circuit designs, solving partial differential equations (20, 21), and the simulation of various physical systems.

General parallel processing for arbitrary applications is the subject of research in alternative computer architectures, operating systems, languages, algorithms, and applications. The speedup potential of algorithms for parallel processing can be predicted by their decomposition function (how finely and at what overhead cost work can be partitioned among the processors) and their access function (the contention for shared data). Decomposition and access functions have been measured for more than a dozen applications (22), and the speedup for those functions was found to be either proportional to N (linear with the number of processors), the square root of N, or log N, or there was no speedup. The existence of multis should greatly accelerate the understanding of parallel processing and hasten its application in the workplace. If so, multis could supplant conventional high-performance uniprocessors.

In addition to the work assigned to a computer, its operating system may also be viewed as a collection of jobs that are candidates for speedup. Transactions in files and databases and communications processing can be done in parallel. For example, the UNIX operating system has been adapted for parallel processing in multis. The degree to which this restructuring is possible determines the number of processors that can effectively be used, since a computer often spends 25 to 50 percent of its time executing operating-system functions. This very large fraction illustrates why early multiprocessors based on the asymmetric principle were not effective. In those systems, the master processor did both operating-system functions and user-assigned jobs, and all the other processors performed only user-assigned jobs until they required help from the master, creating a bottleneck in the master while increasing throughput by only a factor of 2 through 4.

Summary

Multis, a new class of computers, are based on microprocessor technology. Now multis contain as many as 30 processors and rival mainframes in performance, but they are an order of magnitude cheaper to use for traditional computing than mainframes. By 1990, multis with hundreds of processors may be built; however, unless the multis' ability to process instructions in parallel is developed, the current rate of increase in computer speed (15 percent per year) will not be surpassed.

Parallel processing is the basis of the Japanese Fifth Generation Computer project and the Defense Advance-Research Projects Agency's Strategic Computing Program (23). The development of multis is quite likely to advance the development of parallel processing.

References and Notes

1. R. M. Burger, R. K. Cavin III, W. C. Holton, L. W. Sumney, *Computer* 17, 88 (October 1984).
2. C. G. Bell, *ibid.* p. 14.
3. W. Lonergan and P. King, *Datamation* 7, 28 (May 1961).
4. D. P. Siewiorek, C. G. Bell, A. Newell, *Computer Structures: Principles and Examples* (McGraw-Hill, New York, 1982).
5. W. Wulf and C. G. Bell, *AFIPS Conf. Proc.* 41, 765 (1972).
6. "The safeguard data-processing system," *Bell Sys. Tech. J.*, special supplement (1975).
7. J. S. Liptay, *IBM Syst. J.* 7, 15 (1968).
8. C. G. Bell *et al.*, *AFIPS Conf. Proc.* 36, 657 (1970).
9. S. J. Frank, *Electronics* 57, 164 (1984).
10. J. R. Goodman, in *10th Anniversary Symposium on Computer Architecture* (Association for Computing Machinery–Special Interest Group on Computer Architecture, Royal Institute of Technology, Stockholm, 1983), pp. 132–139.
11. The number of instructions per second is proportional to the number of times the processor gains access to the memory per second. The proportionality constant can vary widely among instruction sets. A complex instruction-set processor such as the VAX-11 can gain access to the memory a dozen times with a single instruction whereas a load-store architecture for scalars typified by the Cray designs gains access to the memory less than once per instruction. Thus the number of times a processor gains access to the memory per second is more representative of a computing structure's capability than the number of times per instruction set since it removes the variance introduced by the instruction set. The multi architecture can make use of any type of processor.
12. W. Y. Stevens, *IBM Syst. J.* 3, 136 (1964). This article introduced the concept of a computer family, which is composed of a set of processor implementations. Each member of a family is capable of executing programs written for other family members, and the set of family members spans a range of price and performance. Thus, although the manufacturer develops software for one machine, the client can move to a compatible machine with higher performance whenever it is needed.
13. A microprocessor is usually a set of several chips that includes the microprocessor, a control unit to convert virtual memory addresses to physical memory addresses (the memory management unit), a control unit for floating-point arithmetic, various interrupt-access controls, and the clock-timing control.
14. H. Gerola and R. E. Gomory, *Science* 225, 11 (1984).
15. In the late 1960's Gene Amdahl observed that a processor that handled 1-million instructions per second would require 1 megabyte of memory (4). Furthermore, this would require 1-million bits per second of in-out bandwidth. With extensive multiprogramming and the virtual memories

used for large programs, a balanced system could require at least 4 megabytes of memory per million instructions per second.

16. Digital Equipment Corporation's VAX-11/780 Project began in April 1975 and the first delivery was in March 1978.
17. T. Kidder, *Soul of a New Machine* (Little, Brown, Boston, 1981).
18. The simplest way to ensure cache memories do not contain stale data is for processors to write through the cache memory each time a processor issues a write command. All cache memories monitor the bus for data being written back to memory. If the cache memory contains a copy of data that has been modified, it deletes the stale data. This simple write-through scheme requires much higher bus bandwidth, however; and the system's performance is limited by the ability of a cache memory to monitor the write-through commands from all the other cache memories.
19. In a modern operating system a user may initiate and control several jobs. A single job is often composed of a set of independent processes that, in turn, may each be composed of several tasks. Whereas jobs, processes, and tasks are all candidates for parallel execution, I will use the term job to represent generic computational activity.
20. G. C. Fox, in *IEEE Computer Society Proceed-*
ings, *Compcon 84* (IEEE Computer Society, New York, 1984), pp. 70–73.
21. A. K. Jones and E. Gehringer, *The Cm* Multiprocessor Project: A Research Review* (Carnegie-Mellon University, Pittsburgh, 1980).
22. D. Vrsalovic, D. P. Siewiorek, Z. Segall, E. Gehringer, *Performance Prediction and Calibration for a Class of Multiprocessor Systems* (Carnegie-Mellon University, Pittsburgh, 1984).
23. E. A. Torrero, Ed., *Spectrum* 20 (November 1983).
24. I thank H. Burkhardt III, D. Schanin, R. Moore and S. Frank for their contributions to this article and D. P. Siewiorek and G. Bell for their assistance in preparing the article.