'MegaFrame' aims at mainframe performance

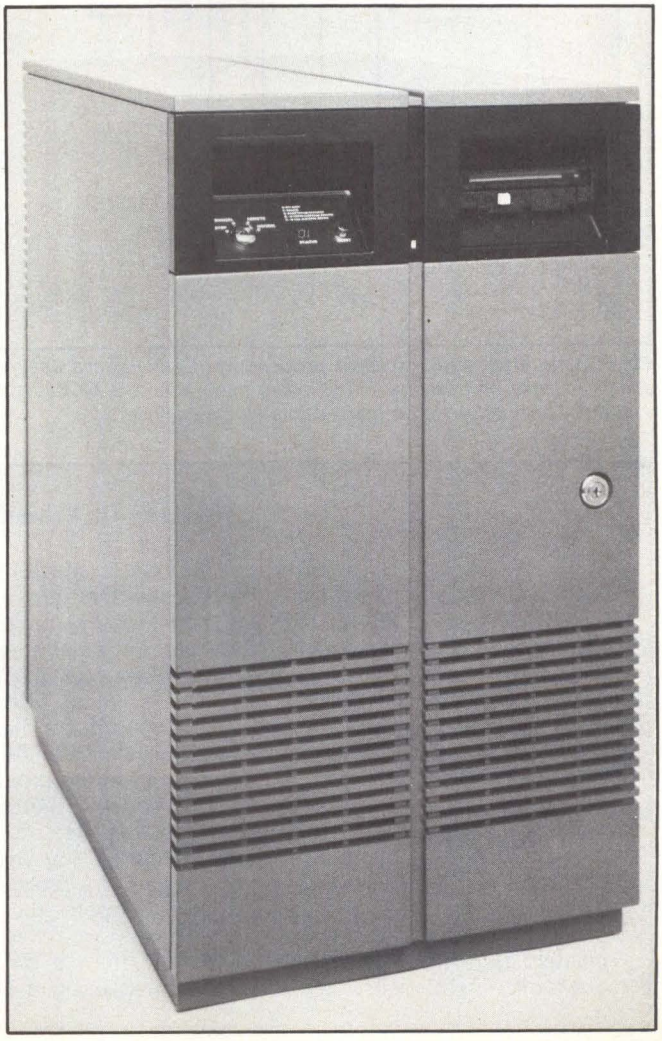
JONATHAN HUIE, RICHARD LOWENTHAL and STEVEN BLANK, Convergent Technologies Inc.

System based on Motorola MC68000 family and Intel 186 processors allows field upgrades to 128 users

The distinctions between microcomputers and minicomputers have become blurred almost to the point of disappearence. Now Convergent Technologies Inc. hopes to extend that blurring to mainframes with its "MegaFrame" system.

The MegaFrame is based on Motorola Inc.'s MC68000 family processors and Intel Corp.'s 186 processors, and it sells for less than $20,000 in a basic eight-user configuration. It can be upgraded in the field to accommodate 128 users without software modification. Its architecture allows both Convergent Technologies' CTOS proprietary operating system and a virtual-memory version of UNIX to share resources, letting users run applications at workstations or terminals without knowing which operating system is in use.

Convergent attempts to transcend the limitations of shared-logic machines through the use of multiple parallel processors.



Limitations of shared logic

Shared-logic architecture has dominated computer system design for 30 years. A drawback of this architecture is its inability to grow as users are added. New users draw against a single CPU with finite computing power. System growth thus has been constrained by three architectural bottlenecks: terminal and communications I/O, disk and file bandwidth and—most important—finite application-processing power.

Mainframe vendors recognized the first system bottleneck was terminal I/O. Terminals interrupting a CPU one character at a time drastically slowed the system. A partial solution was provided by dedicating a

front-end processor to off-load communications overhead from the main CPU.

The second bottleneck, file I/O, involves file access, which requires that the CPU spend time handling the disk and file system rather than executing main code. Dedicating back-end processors to off-load the file-processing overhead from the main CPU helps.

Finally, many large-scale minicomputers and mainframes are limited to a fixed amount of processing power in a single CPU executing the application code, in turn limiting the amount of application processing available to a user.

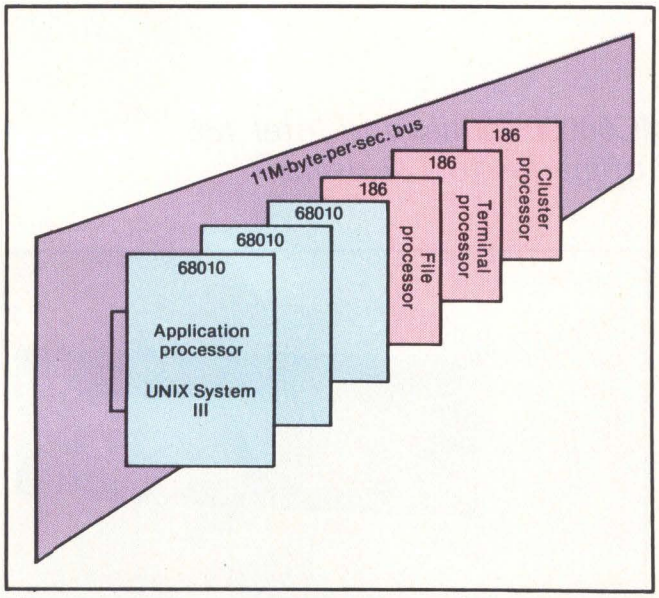


Fig. 1. One enclosure, multiple processors. *Convergent's design allows multiple processors—in this case, three Motorola MC68010s and three Intel 186s—to be housed in a single enclosure.*

Improving on shared logic

The MegaFrame reflects Convergent's belief that dependence on a single CPU is an inherent weakness of all shared-logic computers from IBM Corp.'s 3083 to the new supermicros. The MegaFrame uses multiple concurrent VLSI processors to permit expansion of file, communications and application processing power. As many as 36 independent processing units, each containing a CPU and memory and most having I/O interfaces, are linked by a high-speed, 32-bit-wide asynchronous bus (Fig. 1). The bus can be extended across multiple enclosures, each with a six-slot backplane to a maximum of six enclosures or 36 slots (Fig. 2). Each enclosure supports as much as 200M bytes of integral mass storage.

The MegaFrame's hardware and software is entirely modular, designed to provide OEMs with multiple entry points and upgrade paths. The systems support a mix of dumb terminals and Convergent workstations and high-speed intelligent terminals.

The key to this flexibility is the 11M-byte-per-sec. asynchronous backplane that lets all the system's processors operate in parallel and permits hardware and software to be bundled as subsystems. The processor boards can communicate without using the interrupt structure of conventional single-processor systems. The hardware provides a "doorbell" interrupt, enabling one processor to pass requests to another, separate hardware mailboxes for local and system bus memory access and dual- and triple-ported shared memory. Each processor board gains the attention of another as if it were accessing memory rather than interrupting an external processor. Bus traffic between CPUs consists of request and response blocks to and from processor boards and DMA transfers to and from the disks.

Processing elements operate in parallel

The MegaFrame consists of three main processing elements: the file, the application and the cluster

HOW DEMAND PAGING WORKS

Demand paging is a memory-swapping process based on the concept that only those program segments required for execution at a given time need reside in main memory. The set of segments, or pages, that resides in main memory for any process is called the process working set.

As a process executes, it may need to reference a page that is not in main memory. When this happens, a page fault occurs, and the operating system must bring the required page from the disk into memory, removing an unneeded page from main memory to make room. The MegaFrame's virtual address space—that is, the total space available in main memory and on those segments of the disk that can be paged into main memory—is 4M bytes per processor.

On some virtual systems, each process has a maximum working set size, limiting the total number of pages of the process that can reside in physical memory at once. Such systems use "paging against the process," in which the least-used pages of a process are discarded to make room for new pages. The MegaFrame avoids taking pages away from a process' resident set, instead using a system-wide paging technique called "paging against the system." When a page fault occurs, the system scans all its processes, not just the one executing, to find a page to discard.

The MegaFrame uses the Berkeley UNIX demand-page virtual-memory system integrated into UNIX System III. Under this system, pages allocated to a program can be thought of as arranged in a circle. When a page fault occurs, a pointer advances around the circle, selecting for replacement the first page that has not been referenced during a given number of memory accesses.

processors. Optional elements include the SMD and terminal processors.

The heart of the MegaFrame is the application processor. Running both the UNIX kernel and UNIX applications, it contains a 10-MHz Motorola 68010 CPU, memory-management hardware to support a two-level virtual-paging scheme and 512K to 4M bytes of dual-ported, error-correcting RAM. The memory-management unit provides a high-speed, two-level paging scheme with 4M bytes of virtual address space per board and no wait states. The system handles multiple-processor addressing—as many as 16 CPUs are supported—via an extended address. When addressing off-board memory, the processors issue a 5-byte address. The appropriate CPU recognizes its number and uses the address as input to its map. The entire system, including the MMU and all data and address paths, is designed to accommodate the Motorola MC68020 32-bit processor when it becomes available.

The file processor, functionally equivalent to a back-end database processor, runs the UNIX file system while the application processor is executing applications; it also runs data-management tools such as ISAM programs and a relational database-management system. It contains an 8-MHz Intel 186 processor with 256K to 768K bytes of triple-ported error-correcting-circuitry RAM and the WD-1010 LSI Winchester disk drive controller. The high-speed system bus and memory allow direct memory access to and by other processor boards.

The file processor supports as many as three 5¼-in., 50M-byte Winchester drives (100M- and 145M-byte drives will be supported later) and a removable, 5M-byte cartridge in the first MegaFrame enclosure. Additional file processors support as many as four drives in each expansion chassis (23 drives total), off-loading disk and file system overhead from the first file processor. A maximum of 6 file processors are supported.

As many as 6 file processors and 16 terminal processors run under CTOS.

The cluster and terminal processors, the functional equivalent of front-end communications processors, are dedicated to running dumb terminals and Convergent workstations and intelligent terminals. Both are based on the 8-MHz Intel 186 processor, contain 256K to 768K bytes of ECC RAM and run communications-oriented products under such protocols as SNA, 3270, X.25 and 2780/3780. The terminal processor contains 10 RS232 ports, four supporting synchronous or asynchronous operation and six supporting asynchronous only. Each RS232 line operates as fast as 19.2K bits per sec. As many as 16 terminal processors are supported. The cluster processor controls two RS422 ports and can run terminals at 307K bps and workstations at 1.8M bps.

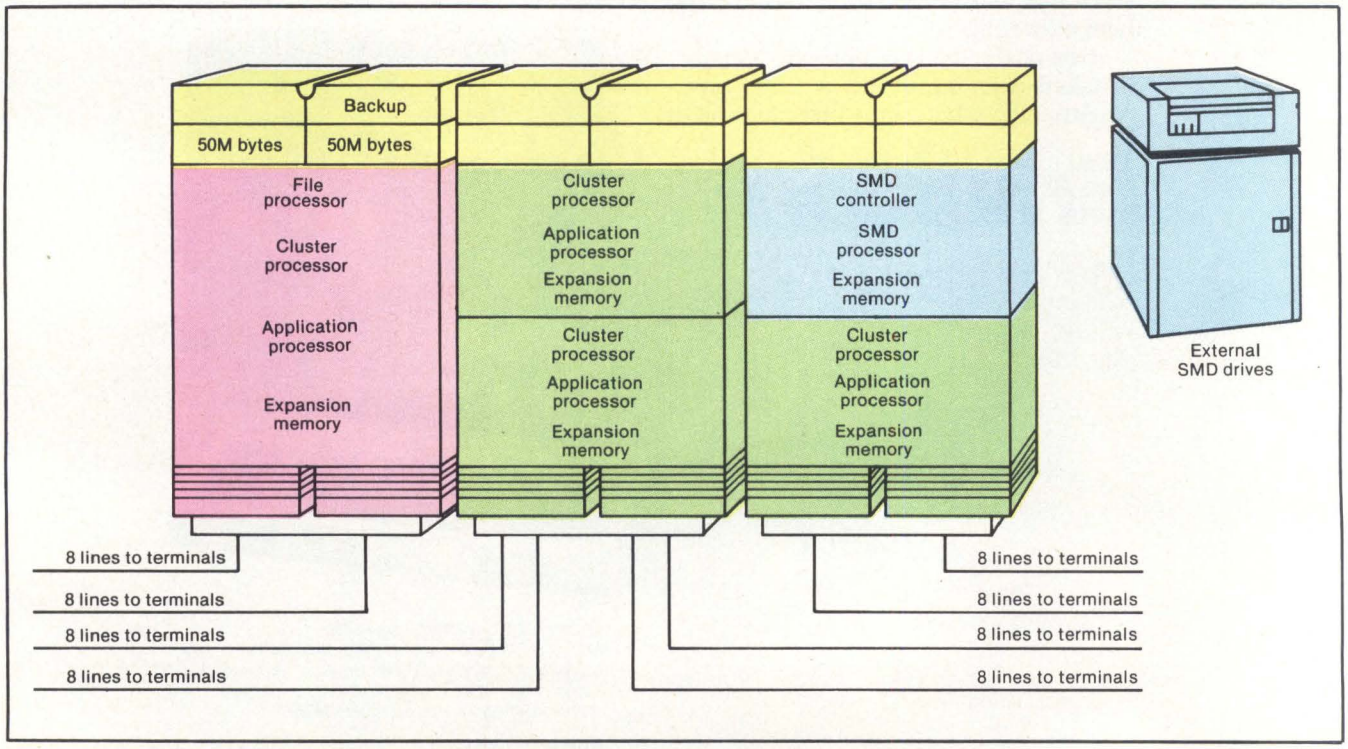


Fig. 2. A system to serve 64 users. *A master chassis (left) and two expansion chassis, plus external mass storage, can be assembled into a system that supports 64 users. Such a system can be expanded to accommodate a maximum of 128 users, 28M bytes of ECC RAM and 1.5G bytes of integral Winchester disk storage.*

This processor also supports three RS232 ports—two synchronous or asynchronous and one asynchronous-only serial-printer interface.

Several other hardware options are available for expansion. An SMD controller handles as many as six 600M-byte SMD drives, memory boards add memory in 500K- or 1M-byte increments, and a Multibus adapter allows incorporation of as many as six interfaces for custom equipment in each enclosure.

Multiple operating systems for multiple processors

Convergent combined its proprietary CTOS message-passing operating system with a virtual-memory version of UNIX fully compatible with UNIX System III, with System V availability planned in the near future. UNIX and CTOS run concurrently and are virtually transparent to a user and the application. The virtual-memory hardware on the MegaFrame's processor—an MMU with a high-speed, two-level paging scheme and 4M bytes of virtual address space per board—permits a demand-page virtual memory system closely approximating the global LRU algorithm.

As many as 6 file processors and 16 terminal processors run under CTOS. Each application processor, to the maximum of 16, runs its own copy of the UNIX kernel. To facilitate message-passing, Convergent has distributed UNIX functionally across multiple processors. Basically, UNIX is distributed in the following manner:

- a copy of the UNIX kernel runs on each application processor in the system,
- UNIX terminal-handling codes such as TermCap, TermLib and Curses use a separate processor running the CTOS operating system,
- the UNIX file system is built on the CTOS file system, which is running on the file processor, and
- all processors and applications communicate over the 11M-byte-per-sec. bus.

Each parallel processor communicates via short messages. Disk data are transferred via DMA into the memory of other processors. The only connection between the UNIX kernel and the other boards is the UNIX inter-CPU communication agent (Fig. 3), which provides request and response blocks to the kernel and all processes. Software running on the application processor communicates with the file system, paging and terminals via the ICC.

Convergent has decentralized UNIX's tree file structure and distributed it across multiple file processors. CTOS also offers multiple-sector data transfers, information kept in memory on open files, asynchronous process execution and redundant volume control structures.

The MegaFrame off-loads UNIX terminal-handling chores from the main processor to multiple independent front-end processors, each running its own operating system. This allows addressing one device, character, line or page at a time without main-CPU intervention. □

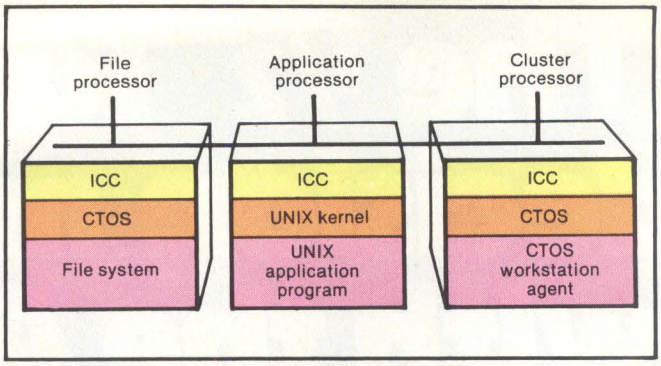


Fig. 3. MegaFrame software organization. *To an application running on a MegaFrame, the computer appears as totally UNIX System III compatible. For example, if an application written in the C programming language were to require some file-system service, the application would make a system call that would be handled by the UNIX kernel. Then the inter-CPU communication module translates UNIX systems calls into CTOS messages, which are passed to the file processor. The file processor retrieves the file and passes it back through the ICC on the application processor, which responds to the systems call.*

Jonathan Huie is manager of software engineering, **Richard Lowenthal** is manager of hardware engineering, and **Steven Blank** is marketing manager at Convergent Technologies Inc., Santa Clara, Calif.