1994 COMPUTER PRESS ASSOCIATION WINNER ♦ BEST COMPUTER NEWSLETTER

MICROPROCESSOR REPORT

THE INSIDERS' GUIDE TO MICROPROCESSOR HARDWARE

VOLUME 9 NUMBER 7 MAY 30, 1995

Intel's P6 Bus Designed for Multiprocessing

First Target Is Servers, But Low-Cost Features Included for PCs

by Linley Gwennap

*This article covers recently disclosed details of the system interface for Intel's P6 processor. Previous articles provide an overview of the P6 (see **090201.PDF**) and a detailed description of the P6 microarchitecture (see **090202.PDF**).*

Intel's P6 combines a unique two-chip processor design with a new high-bandwidth bus. Because the two P6 chips implement a processor with 256K of secondary cache, the P6 system bus is designed to connect efficiently to a long-latency memory subsystem and up to three other P6 processors. The Pentium bus, on the other hand, serves primarily as a connection to a fast L2 cache, with main-memory bandwidth a secondary concern, and supports no more than one other processor.

These differing goals are reflected in a completely revamped system interface for the P6. The new bus is demultiplexed, fully pipelined, and supports split transactions; in the best case, it can sustain its peak bandwidth of 528 Mbytes/s. Cache consistency is maintained in a multiprocessor environment, with data "snarfing" to improve performance. Small voltage swings allow the bus to support up to eight devices at 66 MHz, with the possibility of higher bus speeds in the future.

One objective of the P6 processor is to penetrate the high-end server market, where Intel has been weak. The high bandwidth and MP support are ideal for these servers. In addition, Intel has incorporated error detection and correction features to provide the reliability needed for high-end servers. In short, the new bus, combined with forthcoming chip sets from Intel and others, will simplify the design of powerful four-processor systems that, according to Intel, can deliver up to 1,400 TPS for online transaction processing (OLTP).

In time, the P6 will also be used in mainstream PCs, so many of these bus features can be simplified to reduce the cost of uniprocessor implementations. These simplifications reduce performance somewhat but still leave adequate bandwidth for a single P6 processor.

Grouping Signals Enables Pipelining

The P6 bus consists of several groups of signals, including arbitration, address, data, and response signals. Each group conducts its business independently and in parallel with the others, allowing bus transactions to be overlapped. Transactions can be fully pipelined, much like instruction execution in a pipelined processor.

Figure 1 shows how this pipelining works. The first group of signals handles arbitration. In the figure, arbitration for the first transaction ("A") occurs in cycle one (t_1). During cycle two, all processors analyze the arbitration results and agree on which will be the master for the next transaction. At t_3, the master asserts the address on the request bus, followed by supplemental information in the next cycle. By this time, arbitration for the next transaction ("B") is already under way on the arbitration bus.

At t_6, the target device can signal an address parity error. In the meantime, bus devices have been checking to see if the address hits in their caches; at t_7, these devices use the snoop signals to indicate a hit, in which case data may be returned by the snooping device rather than by the original target. If there are no snoop hits, the target device uses the response bus at time t_9 to indicate whether this transaction has completed successfully; if so, data is transmitted on the data bus starting in that same cycle.

Note that, by this time, the arbitration bus is already up to transaction D. At full speed, a 32-byte read takes 12 cycles to complete but uses the data bus for only 4 of those cycles. With the other buses forging ahead, the data bus can be fully utilized for long periods of time; in this way, three or four transactions can be in progress at once. The P6 bus supports up to eight transactions at once, which can occur if devices throttle the bus to extend the transaction latency, as described later.

Designed for Memory, Not Cache

Because the bus returns the requested word first,

the latency to that critical word is 9 bus cycles, not 12, and can be as little as 7 cycles in a uniprocessor system. In any case, this latency would be intolerable for a secondary cache bus. The P6, however, relies on a separate bus that connects the CPU to the secondary cache, both of which are contained in a single package. Thus, the P6 bus design is optimized for a high-bandwidth memory subsystem with relatively long latency.

At full speed, there are six cycles between the time the address is driven and the time the corresponding data is returned. At 66 MHz, this allows about 90 ns for a memory access. Given some time for buffering and overhead, 60-ns DRAMs are typically required to satisfy this latency.

Once the first word is returned, subsequent values must follow on each 66-MHz (17-ns) cycle to maintain the maximum bandwidth. Thus, a high-performance P6 system must implement either a synchronous DRAM memory system that can sustain 66-MHz bursts, or a four-way-interleaved 64-bit-wide memory system built from standard DRAMs. Of course, low-cost implementations can use standard noninterleaved DRAM memory, but they then could not deliver the full bus bandwidth.

Split Transactions Allow Slow Devices

Despite Intel's claims, the P6 bus is not a pure split-transaction bus, in the sense that the vast majority of bus transactions complete under the control of a single master. Split-transaction buses, such as Sun's XDBus *(see **070301.PDF**)*, initiate a request and then free the bus for other devices to use; the responding device later arbitrates for the bus and returns the data as a separate transaction. In a multiprocessor system, this arrangement improves bus utilization over simpler buses like Pentium's.

The P6 approach improves utilization by pipelining the transactions instead of splitting them. The P6 approach eliminates the need for memory controllers to be bus masters, simplifying their design, while allowing high bus utilization. With arbitration, address, and data fully overlapped, the P6 bus can sustain 100% utilization of the data bus. The protocol does require one dead cycle between reads and writes to turn the bus around, so the bus reaches its peak bandwidth only during a long sequence of read (or write) transactions. With a realistic transaction mix, the bus could achieve 90% utilization.

One advantage of a split-transaction bus is that a slow device does not hold up the entire bus; other transactions can occur during an arbitrarily long latency period. Intel solved this problem by allowing some transactions on the P6 bus to be split, although these are the exception and not the rule. If a device will take significantly more than six cycles to respond, it can defer its response (at time t_9 in Figure 1). In this case, the device must eventually rearbitrate for the bus before finally returning the requested data to the original requester.

Once a transaction is deferred, it does not count against the limit of eight pending bus transactions. Each P6 processor, however, has a limit of four outstanding transactions, including deferred requests from that processor.

Round-Robin Arbitration

Table 1 lists the signals used by the P6 processor, grouped by function. In addition to the signals listed here, the 387-pin package includes 21 reserved ("no connect") pins and 16 test pins.

The arbitrate group includes BREQ[3:0] and BPRI. During arbitration, a processor can request the bus by asserting its BREQ[0] output, and it receives requests from the other processors on BREQ[1:3]. These pins are cross-coupled on the bus so each processor can read the status of the other three. The processors use a symmetric round-robin prioritization that allows each CPU to access the bus in turn.

To speed uniprocessor systems, the P6 supports bus parking. If no other device requested the bus in the previous arbitration cycle, the current bus owner can immediately assert its next request, ignoring any arbitration requests in the current cycle. This method reduces latency by two cycles. Bus parking can occur in an MP system; a device whose arbitration signal is ignored in this way can gain the bus the following cycle, a small penalty.

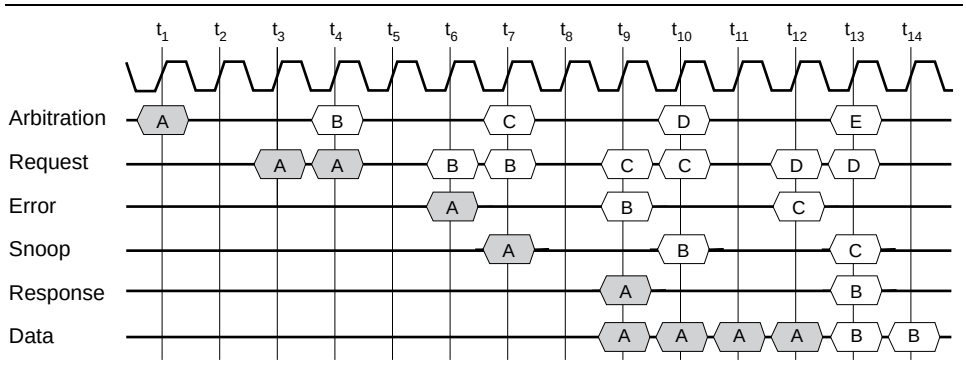

Figure 1. Each group of signals on the P6 bus handles a different transaction phase. This design allows transactions to be overlapped and fully pipelined, sustaining 100% utilization of the bus.

The BREQ signals allow up to four processors to arbitrate among themselves. The BPRI signal is asserted by other devices to obtain the bus. Because memory controllers do not need to master the bus, BPRI is typically used by I/O bridges. In a system with only one I/O controller, that chip can drive BPRI directly. If there are two or more nonprocessors that need to master the bus, they must arbi-

	Signal Name	Type	Description
Arbitrate	BR[3:0]#	I, O	Bus request (one per CPU)
	BPRI#	I	Bus priority request
	BNR#	I/O	Block next request (depipeline)
	LOCK#	I/O	Bus lock
Request	ADS#	I/O	Address valid strobe
	REQ[4:0]#	I/O	Request type
	RP#	I/O	Request parity
	A[35:3]#	I/O	Address (36-bit physical space)
	AP[1:0]#	I/O	Address parity
Snoop	HIT#	I/O	Snoop hit—unmodified line
	HITM#	I/O	Snoop hit—modified line
	DEFER#	I	Defer transaction response
Response	RS[2:0]#	I	Response status
	RSP#	I	Response parity
	TRDY#	I	Target ready for write/snarf
Data	DRDY#	I/O	Data ready
	DBSY#	I/O	Data bus busy
	D[63:0]#	I/O	Data (64-bit)
	DEP[7:0]#	I/O	Data ECC or parity
Error	AERR#	I/O	Address parity error
	BERR#	I/O	Bus error
	BINIT#	I/O	Initialize bus engines
	IERR#	O†	Internal CPU error*
	FRCERR	O	Master/checker (FRC) error*
	THERMTRIP#	O†	CPU shutdown due to overtemp

	Signal Name	Type	Description
APIC	PICCLK	I†	APIC clock*
	PICD[1:0]	I/O†	APIC data*
	LINT0/INTR	I†	Interrupt*
	LINT1/NMI	I†	Nonmaskable interrupt*
Control	BCLK	I†	Bus clock
	INIT#	I†	Initialize CPU (but save data)*
	RESET#	I	Reset CPU*
	FLUSH#	I†	Flush all processor caches*
	PWRGOOD#	I†	All power inputs are good
	STPCLK#	I†	Stop internal processor clocks*
	SMI#	I†	System-management interrupt*
PC	FERR#	O†	Floating-point error*
	IGNNE#	I†	Ignore numeric error*
	A20M#	I†	Mask address bit 20*
Test	TCK	I†	JTAG clock*
	TDI, TDO	I, O†	JTAG data in, data out*
	TMS, TRST#	I†	JTAG test mode select, data ready*
	BPM[3:0]#	I/O	Breakpoint and performance monitor
	PRDY#, PREQ#	O, I†	Debugging use only
Power	VID[3:0]#	O	Voltage request; selects VCCP value
	VCCP	I	47 pins (2.9 V)
	VCCS	I	28 pins (3.3 V)
	VREF	I	8 pins for GTL (1.0 V)
	VSS	I	98 pins for ground
	PLL1, PLL2	I	PLL decoupling capacitor

Table 1. Most of the P6 pinout is consumed by the P6 system bus. Nearly every group of signals on the bus is protected by parity or ECC. 16 test and 21 reserved pins not shown. # indicates active-low signal *same definition as in Pentium †3.3-V signals (others use GTL+ levels)

trate among themselves using a side bus to determine which will drive the next transaction. Requests using BPRI take priority over any processor requests for the bus, but once BPRI stops being asserted, the processors return to their original round-robin sequence.

There is no central arbiter in this model. All processors monitor the arbitration signals and independently decide (and hopefully agree) on the next bus master. While this approach requires more logic in each processor than does a central arbiter, it speeds arbitration and simplifies system design. In the common case of a single I/O controller, that device needs no arbitration logic, since it always wins when it requests the bus.

Request Phase Takes Two Cycles

After winning arbitration, the bus master asserts ADS (address strobe) along with REQ[4:0] (request type) and the address. In addition to the standard 64K I/O address space, the bus supports a 64G memory address space. The P6 uses a new page-table model that creates physical addresses of up to 64 bits, although the hardware supports only 36-bit physical addresses.

Request types include basic reads and writes for memory and I/O, as well as an invalidate-memory transaction that causes other processors to flush an address from their caches so a single processor can have write access to that address. Memory requests can handle up to 32 bytes, matching the line length of the P6's cache, while I/O requests can be 1–4 bytes. Using the byte enables, memory requests can transfer fewer than 8 bytes. Cache consistency is maintained for memory requests.

Additional request types are deferred reply and system functions. A deferred reply indicates that a device that previously deferred a transaction is now ready to return the requested data. System functions include flush, synchronize, halt, shutdown, and interrupt acknowledge. These functions are the same as in the Pentium bus *(see **070502.PDF**)*.

After the initial request information is sent, REQ[4:0] and some of the address lines are asserted on the following cycle to transmit extended request encodings, byte enables, transaction ID, and bus debug information. The transaction ID is used to match the initial request with a deferred reply. Although the initial P6 devices support a maximum of eight pending bus transactions, six bits are reserved for the transaction ID, allowing future implementations to support up to 64 pending transactions.

Data Snarfing Supported

Once a transaction has been initiated, it can complete in a number of ways. Three cycles after the address is driven along with ADS, the receiving device may signal an address parity error; Intel calls this the error phase. The P6 will retry the transaction once; if the address error persists, it will signal a machine check to software.

Unless there is an address error, the following cycle

For More Information

Intel has not announced the P6 processor. For more information, contact your local Intel sales office or check the World Wide Web at *http://www.intel.com*.

is the snoop phase. During this cycle, three signals can be asserted. If any processor finds a clean copy of the requested data in its cache, it asserts HIT, informing the master to mark the data as shared instead of exclusive; other aspects of the transaction proceed normally.

If another processor finds a dirty copy, it asserts HITM. In this case, that processor must respond to the request by driving its dirty data onto the bus, ensuring that the requester receives the most up-to-date copy. The P6 protocol requires the original target device (usually the memory controller) to "snarf" the data as it is transferred, reading it from the bus as it goes by and updating its own copy of that data. These cache-to-cache transfers take place with the same timing as memory-to-cache transactions.

The DEFER signal will defer the completion of the transaction, as noted above. If none of the three snoop signals is asserted during the snoop phase, the transaction completes normally.

The next cycle is called the response phase. The target device indicates its response by asserting RS[2:0]. For a normal read response, the target device returns data on the data bus starting in the same cycle; the transaction may occupy the data bus for up to four cycles. A different encoding of RS[2:0] indicates a write response; in this case, the timing is the same as for a read, but the master device transfers the data. If the data bus needs to be turned around (i.e., a write following a read or vice versa), the response phase is delayed by one cycle.

Two other encodings of RS[2:0] are used only if DEFER had been asserted. A normal deferral indicates that the target device will complete the transaction in the future. The device may instead request that the transaction be retried from the start. Finally, the protocol allows the device to signal a hard failure, which may cause software to take corrective action.

Many Data Integrity Features

To increase its applicability to high-reliability servers, the P6 bus includes many features to protect data integrity. The 64-bit data bus is covered by eight ECC bits, allowing single-bit error correction and double-bit error detection. The 33-bit address bus (bits 2:0 are not part of the address bus but are handled by the byte enables) is protected by two parity bits. The other request signals, as well as the response signals, are protected by their own parity bits. If any parity errors are detected, the transactions can be retried, preventing a transient bus error from crashing the system.

If a hard error does occur, the P6 supports an extended version of Pentium's machine-check architecture. The error is signaled to software via a maskable interrupt; once the interrupt is acknowledged, several registers provide specific debug information. The P6 even has a built-in thermal sensor, implemented in silicon on the CPU die, that disables the internal processor clock if the chip is operating at a dangerously high temperature (junction temperatures of roughly 130°C). In this situation, the processor also asserts THERMTRIP and awaits a reset signal.

Low-Cost Implementations Possible

The P6 bus is designed to deliver enough bandwidth for four processors. Intel's tests show that, even with a heavy transaction-processing load, bus utilization does not exceed 65% with four P6 processors. Future P6 processors will operate at higher CPU clock speeds, increasing the demand for bus cycles, but this increase will be mitigated somewhat by Intel's plan to increase the secondary cache from 256K to 512K.

A single P6 processor, however, does not need anything like 528 Mbytes/s of bandwidth. In fact, one P6 is unlikely to use more than 20% of that capacity. Thus, uniprocessor systems may choose to throttle the bus, reducing bus bandwidth but with little impact on system performance. Throttling allows the use of slower devices, reducing cost.

The P6 bus protocol allows slow devices to extend the transaction time in several ways. After winning an arbitration, a device can take more than the minimum two cycles to assert ADS. During the response phase, the receiving device can hold off the data phase (for example, if its buffers are full) by waiting to assert target ready (TRDY). Once it asserts TRDY, it must be able to receive a complete burst of data at the full bus rate.

When sending data, a device can insert wait cycles between words by not asserting data ready (DRDY) on each cycle. A device can issue the ultimate brush-off by deferring a request, giving it plenty of time to calculate its response. To further simplify the bus interface, a device can assert BNR to block the next request, essentially depipelining the bus.

For example, a uniprocessor system could use a single-bank DRAM memory subsystem to reduce cost. This subsystem, however, could not supply or receive data at the full 66-MHz bus speed. By using DRDY, the memory controller could insert enough wait states to allow the DRAM to operate at its own speed. Because the bus returns the critical word first, such an implementation might not significantly reduce the performance of a single P6 processor on typical PC applications.

Low-cost implementation may also disable ECC

and parity checking on the bus, simplifying chip sets and eliminating the extra main memory required to store the ECC bits. This change has no performance impact but does reduce system reliability.

Chip Sets Difficult to Design

As with Pentium, the first chip sets for the P6 will come from Intel. Figure 2 previews what Intel calls the P6 PCIset, previously called Orion. (The company has not yet officially announced this product.) This high-performance chip set is intended for servers; it requires seven chips for a basic implementation, although four are simple multiplexers.

The I/O side is fairly simple: a single chip (PB) provides a complete P6-to-PCI bus bridge that allows the PCI clock to operate at any ratio relative to the P6 bus. The PCIset does not include any additional I/O functions; standard products are available to add graphics, networking, and other functions to the PCI bus. A second PB is optional; the two PB chips will arbitrate between themselves before asserting BPRI on the P6 bus. With two PCI buses, the peak system I/O bandwidth is 264 Mbytes/s.

For pin-count reasons, the memory controller is split into two chips, DP and DC. Because DP handles the data path and DC the control signals, the two chips together place only a single load on the bus. The two chips implement a four-way interleaved DRAM subsystem capable of sustaining the full bandwidth of the P6 bus. Four MIC chips, which are essentially four-way multiplexers, connect the memory controller to the DRAM.

The current PCIset does not support EDO DRAM, which would allow two memory banks to supply the full bus bandwidth. EDO support will be added in a future product. The P6 bus is also a good match for Rambus DRAMs, which have a long latency but high bandwidth. Intel has not committed to supporting Rambus parts.

The PCIset supports up to four P6 processors. While this is literally true for any P6 chip set, this design can actually supply data fast enough to keep up with four CPUs, particularly since a second memory subsystem and a second PCI bus can be added. It can also be used in uniprocessor systems, but Intel will probably release a second, low-cost chip set tuned for P6 PCs.

With each successive processor generation, it has become more difficult to design these chip sets. A basic 486 chip set can be designed in about six months by a few good engineers, as demonstrated by several startup companies in the past few years. Pentium chip sets are more complex, but several vendors are now shipping products.

The P6 forces chip-set designers to cope with a highly pipelined bus, split transactions, multiprocessor cache consistency, data snarfing, and GTL+ signal levels (see sidebar below). Some of these factors can be ignored in a uniprocessor-only design, but by the time the P6 reaches the PC mainstream in 1997, multiprocessor-ready desktops may begin to replace the current OverDrive design as the upgrade strategy of choice. Although all of this could make life difficult for smaller chip-set vendors as they look toward the P6 generation, we expect the major chip-set makers to support the P6.

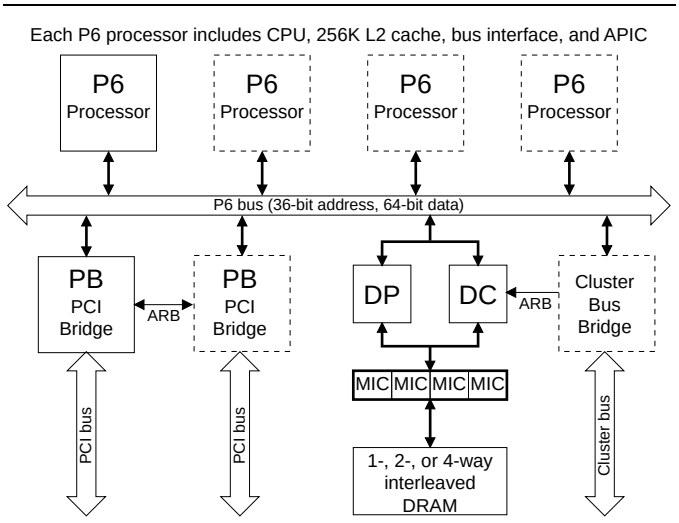

Figure 2. The P6 PCIset connects up to four P6 processors with PCI and a four-way-interleaved memory subsystem. The memory controller is split between the data path (DP) and data control (DC) chips. The MICs are essentially multiplexers for the DRAMs.

One opportunity for third parties is at the very high end. For systems with more than four processors, the P6 requires a cluster design, as the bus has neither the arbitration lines nor the electrical ability to handle so many CPUs. Such systems would have two or more P6 buses, each with up to four processors.

The buses would be connected by P6-to-P6 bridge chips that detect transactions to memory or I/O addresses on remote buses. These cluster controllers would then defer the request on the originating bus while passing it along to the appropriate remote bus. After obtaining the data, the cluster controller would perform a deferred reply to return the data to the original requester. Figure 2 shows an optional cluster controller.

Intel has no current plans to produce such a cluster controller, but Corollary is said to be working on one, and other vendors of multiprocessor Pentium chip sets are probably considering it.

Crashing the Glass House

Intel believes that the P6's high-performance bus will enable high-end servers that cost much less than comparable RISC-based systems. This lower cost is not derived from any technical advantages: the major next-generation RISC processors, due in the same timeframe as the P6, all implement similar system interfaces with higher sustainable bandwidth than the P6, with UltraSparc leading the way at 1,333 Mbytes/s *(see **090703.PDF**)*.

The cost advantage of a P6-based design lies in its

GTL+ Speeds Signals

At the physical level, the P6 bus operates with a modified version of GTL *(see **070301.PDF**)*. Intel licensed the basic GTL technology from Xerox but increased the supply voltage to 1.5 V, from 1.2 V in the original design, to improve noise margin. The P6 design also uses a 1.0-V reference and termination resistors of 50 Ω, both slightly different than Xerox's implementation. Intel may have patents that involve some of these changes. The company refers to its design as GTL+ to distinguish it from standard GTL.

This design is quite different from the Pentium bus, which works at CMOS levels. In general, the smaller voltage swings of GTL+ should allow a higher bus speed than the Pentium bus, which reaches 66 MHz. But much of the extra margin is used to accommodate up to eight devices, including four processors, on the P6 bus. This configuration creates a physically long bus that has many stubs, an unfriendly electrical environment. To further complicate matters, many P6 systems will ship with one or more empty sockets, allowing upgrade processors to be added later.

Thus, Intel is specifying the new bus at 66 MHz, the same as its predecessor. It could probably run faster in smaller configurations, but only the largest configurations need greater bandwidth. It is possible that, as hardware designers become more experienced with the P6 bus, they will discover ways to push the clock speed to 80 or even 100 MHz, speeds that will be required to keep pace with third-generation P6 processors, expected to reach 300 MHz in 1997 or so.

The current implementation takes several steps to allow for high-frequency operation. All signals have a full cycle to flow across the bus; no logic functions are performed during the transmission cycle. This technique maximizes the flight time for signals. GTL+ has better noise immunity than CMOS and offers controlled edge rates, which help reduce settling time even with stubs and empty sockets on the bus.

leverage of the PC infrastructure. PC vendors interested in the server market can use the same PCI peripherals, disk drives, and other components that they already purchase in high volume. Compaq is already using this business model with 486 and Pentium servers, and other large PC vendors may follow with the P6.

Such newcomers, however, will find themselves on foreign turf. The decision to buy an expensive MP server is far different from the decision on a $2,000 PC. Traditional server vendors such as HP, IBM, and Digital have the direct sales force and the high-level corporate contacts needed to sell these pricey systems. These vendors also offer the costly support that customers demand for these systems, along with proven high-end operating systems and application software.

Compaq or any other PC company will take some time to duplicate this infrastructure. But it isn't impossible: Sun's recent success in penetrating the corporate server market may serve as a model for these aspirants. Over time, the P6 will come to play a significant role in the high-end server market.

Edging Out CPU Competitors

The P6 bus also offers Intel another advantage over its direct competitors, particularly AMD and Cyrix. These companies will offer processors compatible with the Pentium pinout just as Intel is deploying P6. Even though the K5 and M1 cores offer significant design improvements over Pentium, this performance advantage is reduced on many applications because of the limited speed of the Pentium bus.

For their next designs, these competitors are undoubtedly considering adopting the P6 bus, just as they have used Intel's other buses in the past. Intel is currently withholding key technical information about the bus from the public and may never release the complete bus specification, much as it has kept Pentium's Appendix H secret. Even if competitors reverse-engineer the bus, Intel may bring system-level patents to bear against system vendors that use these competing products.

These issues may convince competitors to define their own buses. But if they do, they give up compatibility with the broad range of chip sets and motherboard designs that will become available for the P6. The best move would be for Intel's competitors to unite behind a single alternative bus for P6-class processors. Together or separately, these companies may be forced to get out of Intel's slipstream and race on their own. ♦

