MICROPROCESSOR REPORT
THE INSIDERS' GUIDE TO MICROPROCESSOR HARDWARE

Volume 10, Issue 14 October 28, 1996

Digital 21264 Sets New Standard

Clock Speed, Complexity, Performance Surpass Records, But Still a Year Away

by Linley Gwennap

Brushing aside pretenders to the performance throne, Digital's Jim Keller demonstrated at last week's Microprocessor Forum why Alpha continues to be king. The company is pushing microarchitecture to new levels, introducing innovations in branch prediction, datapath design, and system interfaces as part of the forthcoming 21264. These features will not only propel the 21264 to unprecedented levels of performance, they will also set standards for future designs.

Previous Alpha processors, while delivering leadership performance, have generally pushed hardest on the "speed" button while relying on simpler CPU cores than their competitors. This tradition goes back to the design of the original instruction set, which ruthlessly eliminated any instruction that might reduce clock speed. The 21264 does not disappoint on clock speed: at 500 MHz, it is faster than any other announced CMOS microprocessor and twice as fast as most. Yet the new chip adopts an aggressively out-of-order CPU core that exceeds the capabilities of current chips.

The 21264 adds a high-bandwidth system interface that channels up to 5.3 Gbytes/s of cache data and 2.6 Gbytes/s of main-memory data into the processor, feeding its demanding CPU core. Digital's Dick Sites laid the groundwork for this interface through his research into performance bottlenecks (see MPR 8/5/96, p. 18), summarized in his rallying cry, "It's the bandwidth, stupid!" While other processors do well on programs, including many of the SPEC95 benchmarks, that fit into large L2 caches, the 21264 will excel even on code that frequently accesses main memory.

Which is not to say the 21264 does poorly on SPEC95: Keller projects scores in excess of 30 on SPECint95 and 50 on SPECfp95 (base). These scores more than double those of any existing microprocessor. One caveat: the chip has not yet taped out, and Digital projects system shipments no sooner than 4Q97. Even with more than a year to catch up, however, Digital's competitors have nothing on the books that will come close to the 21264's performance in that timeframe.

Four Instructions Per Cycle

The core of the 21264 is an highly out-of-order processor with a peak execution rate of six instructions per cycle and a sustainable rate of four per cycle. The processor can keep up this pace on either integer or floating-point code. Up to 80 instructions can be in process at once, more than in any other microprocessor. Registers are renamed on the fly, with 41 extra integer registers (80 total) and 41 extra floating-point registers available.

As Figure 1 shows, instructions are decoded and then assigned to either the integer or floating-point queues. Each cycle, all instructions that have their operands available arbitrate for access to the function units. Instructions that have been in the queue the longest have priority. After arbitration, an instruction can be issued to each of the function units. Instructions that have dependencies and are waiting for data are bypassed in favor of instructions that can execute right away, creating opportunities for instructions to execute in an order different from the one specified by the program.

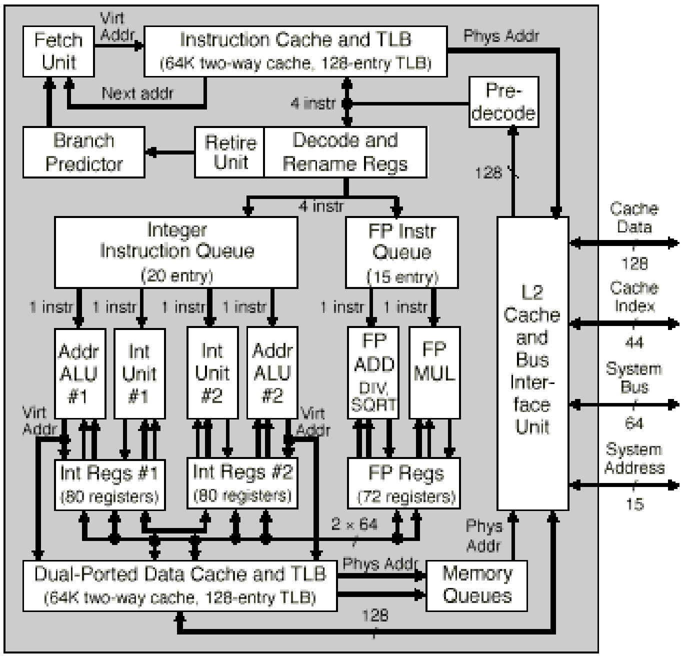


Figure 1. The 21264 issues instructions out of order from both the integer and FP queues into six function units grouped in three "clusters." Unlike the 21164, the new processor has two large primary caches on the chip.

The 21264 includes four integer execution units: two general-purpose units and two address ALUs. The latter pair executes all load and store operations (for either integer or FP) and can also perform simple arithmetic and logical operations. The general-purpose integer units execute arithmetic and logical operations plus shifts and branches. One integer unit has a multiplier; the other handles the new Alpha motion-video instructions (see sidebar).

Clustered Integer Units

This array of function units can execute four integer instructions per cycle for most instruction mixes. The downside is that a standard implementation would require an integer register file with eight read ports and six write ports. Digital discovered that the physical width of this enormous register file caused the entire datapath to be distended, increasing the cycle time beyond the tight target.

Instead, the 21264 duplicates the integer register file, with each copy having four read and six write ports, reducing the datapath width significantly. As Figure 1 shows, each register file services a general-purpose integer unit and an address ALU in a grouping Keller calls a "cluster." The register files are kept synchronized to ensure correct execution, but it takes an extra cycle to write data from one cluster's ALUs to the other's register file, due to the physical distance between them and the minimal cycle time. Digital's simulations showed a 1% performance degradation from this penalty, a small price to avoid degrading the cycle time.

The instruction queue understands the difference between the two clusters and naturally issues instructions in an efficient manner. As instructions arbitrate to be issued, their operands will be available in one cluster a cycle earlier than in the other. Thus, a string of dependent instructions will tend to be issued in sequence to the same cluster.

In theory, common integer instructions, such as adds, can be issued to any of the four integer units. A completely general design, however, would have required complex issue logic that would have jeopardized the cycle time. Instead, the 21264 preassigns simple integer instructions to either the general-purpose units or the address ALUs as they enter the instruction queue. Preassignment does not determine the cluster to which an instruction can be issued.

In this way, all instructions can be assigned to either the address ALUs or the general-purpose units as they enter the queue. This method reduces the number of instructions potentially arbitrating for each ALU, speeding the arbitration logic. Digital's simulations showed this simplification causes only a 1-2% performance loss from the fully generalized case without increasing the cycle time.

With only two function units, the floating-point unit is organized more traditionally, with a single physical register file. Since FP loads and stores are executed in the integer unit, the 21264 can sustain four FP instructions per cycle given the right mix of 50% memory references, 25% multiplies, and 25% other FP operations.

Both floating-point units are fully pipelined, with a four-cycle latency for add, multiply, and most other operations. A double-precision divide takes 16 cycles, while a double-precision square root requires 33 cycles; these operations are not pipelined.

Pipelined Primary Caches

The on-chip cache architecture is completely revamped from the 21164 (see MPR 9/12/94, p. 1). That chip has tiny 8K primary caches, their size limited by the need to access them in a single 2-ns cycle. These caches are backed by a 96K secondary cache, fully pipelined but carrying a six-cycle latency, that is also on the CPU chip. Finally, a large tertiary cache must be added externally for reasonable performance.

The miss rate of the 8K primary caches is relatively large, causing frequent accesses to the six-cycle L2 cache. In addition to waiting for these accesses to complete, the 21164 bears the overhead of moving data back and forth between the primary and secondary caches.

In contrast, the 21264 has two 64K primary caches on the chip. These large primaries, which are two-way set-associative, have a much better hit rate than the 21164's small L1 caches and avoid the overhead of that chip's two-level on-chip cache. The downside: the new chip's primary caches generally take two cycles to access.

The problem is not in the cache array itself; the latency through the 64K arrays is only one cycle, allowing accesses to be fully pipelined with ease. The cache-array access, however, takes nearly a full cycle, leaving no time to move the address or the data any significant distance across the large die. Moving instructions from the cache to the instruction decoder, for example, adds a cycle to the instruction latency.

HP's PA-8000 processor (see MPR 11/14/94, p. 1) is the only other recent microprocessor to use multicycle primary caches, although HP took this path to allow external primary caches at 180 MHz. As other processors approach 500 MHz, they will run into the same problems Digital has. Some may stick with tiny primary caches, as Exponential did(see cover story), but we expect most will ultimately adopt the multicycle strategy, as the cost is relatively small.

Assuming the primary caches are fully pipelined, as they are in the 21264, extending the latency to two cycles has two significant effects. A multicycle data cache extends the load-use penalty, that is, the number of cycles an instruction must wait if it requires data from an immediately preceding load instruction. This delay would sap the performance of an in-order processor, but an out-of-order processor simply executes nondependent instructions while waiting for the data cache. Digital estimates adding a cycle of data-cache latency costs about 4% in overall performance. Although this penalty sounds significant, the alternatives are a much lower hit rate on the primary caches or a decrease in clock speed, both of which would cause a greater performance loss.

Advanced Branch Prediction

A longer instruction-cache latency creates a different problem. As Figure 2 shows, the mispredicted branch penalty is at least seven cycles in the 21264, including two cycles to access the instruction cache. Because instructions typically spend 4-5 cycles in the instruction queue, the average mispredicted branch penalty is more than 11 cycles. Thus, the extra cache cycle extends the branch penalty by only 10%; the impact on overall performance is about 1%, according to Digital.

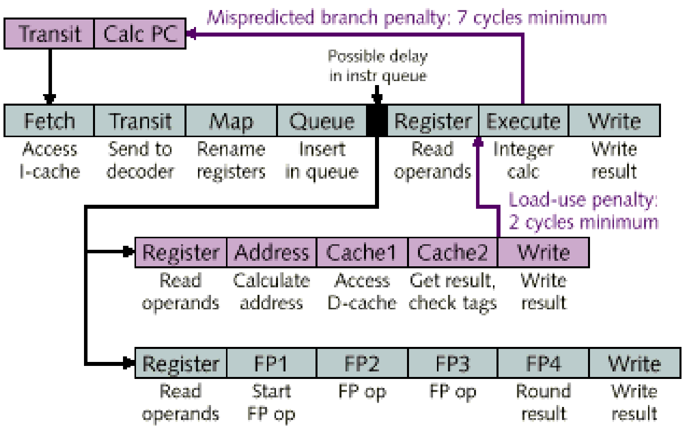


Figure 2. The basic 21264 pipeline is seven stages for a simple instruction and nine stages for a load or store.

The impact would be larger except for the great lengths to which the 21264 goes to avoid mispredicted branches. The chip uses a method originally developed by Scott McFarling of Digital's Western Research Lab (WRL) and published in a 1993 paper (on the Web at www.research.digital.com/wrl/techreports/abstracts/TN-36.html).

Earlier research developed a variety of methods for predicting branches (see MPR 3/27/95, p. 17), some more accurate than others. This accuracy is not universal, however: different algorithms work well on different types of branches. McFarling realized branch prediction could be improved by using a hybrid method that combined two different algorithms, picking the better algorithm dynamically.

As Figure 3 shows, the 21264 combines a two-level table, which Digital calls the local predictor, that is indexed by the program counter with a single-level table, the global predictor, indexed by a global history of all recent branches. A third table observes the history of both predictors and chooses the better algorithm for each particular situation.

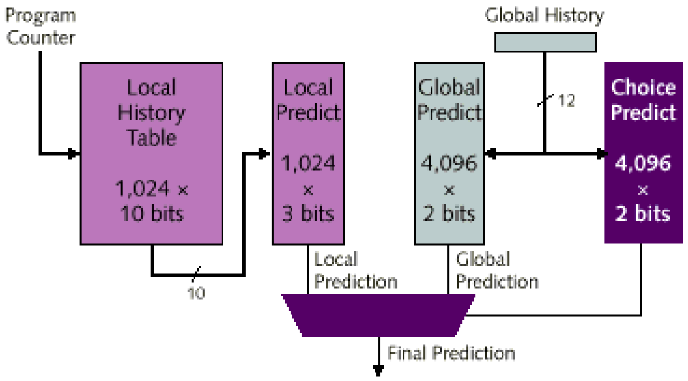


Figure 3. The 21264 combines two branch-prediction methods, dynamically choosing the more accurate on a per-branch basis.

Digital says this combination of algorithms reduces the number of mispredictions to between 7 and 10 per thousand instructions on the SPECint95 benchmarks. At 11-12 cycles per mispredicted branch, the impact of mispredictions on performance is about 0.1 cycles per instruction. As the 21264 averages about 0.5 CPI on SPEC95, the impact of mispredicted branches is significant but not overwhelming.

Other vendors typically quote prediction accuracy in terms of mispredictions per branch, not per instruction. Assuming SPEC95 has one branch every six instructions, the 21264 would have a success rate of about 95%, the best value reported on SPEC95 for any commercial microprocessor. The misprediction rate will be worse, however, on many business applications, which typically have more branches and branches that are more difficult to predict than the technically oriented SPEC95 programs.

The cost of this accuracy is about 35K bits of storage for all the requisite branch-history information, which consumes about 2% of the processor's total die area. This figure does not include the predicted target addresses; as described below, these are stored in the instruction cache on a per-line basis, adding another 48K bits.

Autonomous Fetch Unit

Another potential problem with extending the instruction-cache latency is handling taken branches. If the processor waits until the instructions are fetched and decoded to detect a branch and redirect the instruction-fetch stream, it would induce a multicycle penalty for each predicted-taken branch. Instead, the 21264 takes advantage of the fact that the cache array produces a result in a single cycle.

Each line in the instruction cache holds four instructions along with a "next-line" predictor and a set predictor. These two fields are immediately sent back to the cache inputs to start the next access. If the line contains a branch that is predicted taken, the predictor fields point to the cache line and set of the predicted target; otherwise, they simply point to the next sequential address.

Assuming the prediction fields are correct, a taken branch has a zero-cycle delay, since the target group of instructions is fetched immediately after the branch. The predictors are initialized to point to the sequential address; they are updated with the branch target address when a branch is predicted taken. The fields are controlled by the branch prediction unit, resulting in a high degree of accuracy. This design allows instruction fetching to proceed fairly autonomously, feeding four instructions per cycle into the decode unit with little external intervention.

Like other chips, the 21264 contains a return-address stack to predict the target address of subroutine returns. This stack is significantly larger than in other microprocessors, allowing accurate prediction through 32 levels of subroutine calls. This stack is located near the instruction cache and is accessed in parallel, so the result can be directed to the cache address input with minimal delay.

One downside of this arrangement is that instructions are always provided in an aligned group of four. If a predicted-taken branch is not in the fourth slot, some issue slots are wasted. Similarly, if a branch target is not the first instruction in a group of four, issue slots are wasted. Instruction misalignment is a common problem that keeps the chip from sustaining four instructions per cycle. Recompiled code will typically align branch targets, but arranging branch instructions properly is much more challenging.

Along with the predictor bits, the instruction cache contains three predecode bits per instruction. These identify the target function unit of the instruction: general-purpose integer, address ALU, or FP unit. The preassignment of integer arithmetic operations occurs as instructions are predecoded and loaded into the instruction cache; this common technique simplifies the later decode stage.

Double-Speed Data Cache

When designing a processor with a large die and a cycle time of 2 ns, keeping signals within a small area of the chip is imperative. As Figure 1 shows, the chip functions are arranged in groups such that each communicates with only one or two others, allowing them to be arranged as near neighbors. One exception to this rule is the data cache, which must work with both integer clusters, the FPU, and the system interface. As a result, it takes two cycles to get an address to the data cache, access the cache array, and return the data to the requester.

The data cache is dual ported, able to supply two independent 64-bit results per cycle. Unlike some processors, which use a banked structure to simulate dual porting, the 21264 pipelines its data-cache array to produce two results per cycle, eliminating bank conflicts and allowing any combination of two load/stores per cycle. This pipelining is achieving by starting a new access on each half-clock cycle, operating the data cache at a frequency of 1 GHz.

In parallel to the data-cache access, virtual addresses are sent to the fully associative 128-entry data TLB and the tag array. These structures, which have a slightly longer access time than the cache data array, are duplicated to support the dual porting. After translation, memory addresses are logged in the memory reorder buffer, where they are held until the associated instruction is retired. This structure checks for multiple accesses to the same physical address; if detected, such pairs of instructions must be executed in program order to avoid errors.

Up to eight cache miss requests are held in the miss queue, waiting for access to the external cache. Cache lines displaced from the data cache are held in an eight-entry victim buffer. Both the primary and secondary caches are nonblocking, continuing to service other requests while these transactions are pending.

Flexible External-Cache Interface

Although its 128K of on-chip cache ranks the 21264 among the leaders in this category, the powerful CPU core has a voracious appetite for data that often overwhelms the on-chip caches. Unlike its predecessor, which has a single 128-bit bus to the external cache and main memory, the new chip includes a dedicated 128-bit L2 cache bus and a separate 64-bit system bus, as Figure 4 shows. Both can operate at speeds up to 333 MHz.

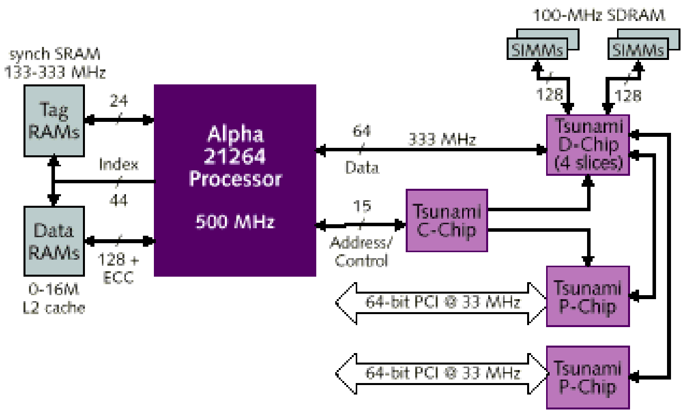


Figure 4. The 21264 connects directly to its external cache via a 128-bit data bus. The Tsunami chip set converts the 333-MHz 64-bit system bus into a more reasonable interface to SDRAM main memory and two 64-bit PCI buses.

The external cache is controlled directly by the processor and can be as large as 16M, although practical systems will probably implement 1M-8M. The cache consists of synchronous SRAMs with a 2.0-V HSTL (high-speed transistor logic) interface and can be configured to operate at various speeds, including 2/3, 1/2, 1/3, and 1/4 of the CPU clock speed. These configurations allow the system designer to trade cost for performance.

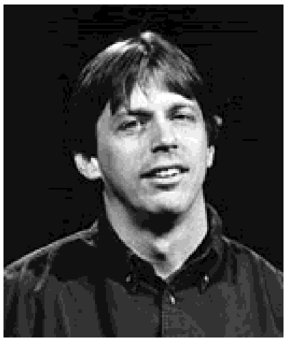
The lowest-cost 21264 systems will use 133-MHz synchronous SRAMs. These parts are expected to become inexpensive once Intel's Klamath processor debuts early next year, as Klamath will use similar SRAMs. Because Klamath is expected to have a 64-bit cache interface, however, the Alpha chip will be able to get 2.1 Gbytes/s, twice the bandwidth of the Intel processor, from these SRAMs.

For better performance, the 21264 can use SRAMs at 200 to 250 MHz; the latter parts are not available today but are likely to be by the time the 21264 begins shipping. At that top speed, the cache could deliver 4.0 Gbytes/s. These parts, of course, are far more expensive than slower SRAMs.

Along with a consortium of leading SRAM vendors that includes Motorola and Samsung, Digital has specified a new "dual-data" SRAM. This part is similar to a 5-ns SRAM but internally fetches twice as much data on each access. A small amount of logic sends this data across the output pins in two transactions. Using both edges of a 167-MHz clock, these parts can produce data at 333 MHz, increasing the 21264's cache bandwidth to its maximum 5.3 Gbytes/s.

This figure still lags the record set by the PA-8000's 5.8 Gbytes/s of cache bandwidth, but the HP chip has absolutely no on-chip cache and relies on this bandwidth to service all the CPU's demands. Digital estimates even the midrange half-speed cache will supply enough bandwidth for the 21264 to achieve its SPEC95 performance goals, although the quarter-speed cache could degrade integer performance by 5% and FP performance by 20% or more. The 333-MHz cache will be most useful for future high-speed versions of the 21264.

While these cache options have a major effect on bandwidth, the latency to the cache does not differ much. The dual-data SRAMs actually have the same latency as a 200-MHz part because they share the same internal memory array. With any of the faster SRAMs, the load-to-use latency of an L2 cache access is 12 CPU cycles; only two of these cycles are due to the SRAM itself. With the 133-MHz SRAMs, the latency is 14 cycles. Because even this low-cost cache supplies enough bandwidth for SPECint95, the relatively small increase in latency keeps performance on this metric close to that of the faster caches.



Digital's Jim Keller explains how the 21264 will set new standards for bandwidth and performance. (photo by Michael Mustacchi)

High-Bandwidth System Bus

The simplest way to increase the bandwidth of the system bus is to make it 128 bits wide. This decision, however, would increase the package size, and thus the cost, of the processor as well as all other chips that connect to the system bus. Instead, Digital chose to extend its high-frequency CPU design to the system level, running the 64-bit system bus at speeds up to 333 MHz. At this speed, the bus can sustain 2.0 Gbytes/s, in excess of the memory bandwidth of any current microprocessor except IBM's P2SC (see MPR 8/26/96, p. 14). Using similar divisors as the external cache, the system bus can operate at lower speeds in cost-sensitive designs.

As Figure 4 shows, this high-speed "bus" is really a point-to-point connection, simplifying system design. Digital’s Tsunami D-chip demultiplexes the 64 data bits into a 256-bit-wide SDRAM subsystem that can operate at a relatively leisurely 100 MHz; four D-chips are required. The D-chips also connect to two PCI bridges, preventing them from loading the high-speed system bus. A separate C-chip connects to the processor’s address bus and controls the remainder of the system-logic chip set.

A low-cost system can be configured with two D-chips and one P-chip, providing a 128-bit memory system and a single PCI bus. A large system, in contrast, could use eight D-chips to support a 512-bit memory system.

The system bus uses a split-transaction protocol that allows up to 16 memory references to be in progress at once, maximizing the utilization of this bus. Multiple processors cannot be added to this bus, however, as that would violate the point-to-point design. Instead, each processor must have its own connection into the chip set. Tsunami supports a second processor; a four-processor version could be created by adding 16 more pins to each D-chip and 30 more pins to each C-chip. Note that, in an MP system, each 21264 retains its full 2-Gbyte/s channel to memory, although the bandwidth of the memory system itself must still be shared among the processors.

Impressive Transistor Count

The 21264 initially will be built in the same process as the 500-MHz 21164: Digital’s 0.35-micron six-layer-metal CMOS-6. The transistor count is a whopping 15.2 million; although most are in the large caches and branch predictor, the CPU core contains about 6 million transistors. This figure surpasses the 4.2 million transistors in the P6 CPU core, the most complex processor currently shipping.

Digital expects the 21264 will reach 500-600 MHz in this process, slightly faster than the current 21164. In adding complexity, the designers avoided any feature that would slow down the pipeline relative to the 21164 and modified some of the critical timing paths for a small speed boost. The high clock speed and transistor count combine to drive the maximum power dissipation to a scorching 60 W at 500 MHz.

Because the chip has not yet taped out, the die size is not final. Keller expects the die to be about the size of the original 21164, which weighed in at 298 mm2. The 21264 will be packaged in a 588-pin ceramic PGA to accommodate the two large buses and their associated address and control lines. Although most other vendors are turning to lower-cost BGA-type packages at such high pin counts, Digital felt such a move would be too risky.

The MDR Cost Model estimates the 21264 will cost about $300 to manufacture, topping all announced microprocessors. Of course, this is only one component of the overall system cost: Digital’s performance estimates require adding megabytes of fast SRAM, a set of 6-10 system-logic ASICs, gobs of 100-MHz SDRAM, and a big fan. Clearly, leadership performance will come at a leadership price.

Help is on the way. A shrink to Digital’s 0.25-micron CMOS-7 process (see MPR 9/16/96, p. 11) could begin shipping as early as mid-1998. This move could reduce the manufacturing cost to well below $200. More important, the clock speed could reach an eye-popping 800 MHz, with corresponding increases in performance. Power dissipation could rise slightly, however, as the CPU core voltage will not drop enough to counter the steep rise in clock speed.

Filling a 0.35-Micron Die

If the 21264 can deliver on its performance and schedule goals (and Digital has a solid track record in this regard), it could nearly double the performance of every other shipping processor when it appears late next year. Along with a strong design effort, a key factor in this performance lead is that the company has taken better advantage of IC process technology than its competitors have.

The 21264 will probably be the first high-end processor optimized for a true 0.35-micron process. To take full advantage of a new process technology, a microprocessor vendor must redesign its core to consume the maximum available transistor budget. Simply shrinking a previous-generation processor provides a performance gain due to higher clock speeds, but the per-cycle throughput is not increased.

Most current high-end processors, including the 21164, the PA-8000, UltraSparc, and Intel's P6, were initially designed to fill a 0.5-micron die. The R10000 uses a nominal 0.35-micron process but has a transistor density similar to that of a 0.5-micron chip. These processors can't compete with the 21264 on transistor budget, giving the Alpha chip a leg up in the performance race. The 21264 is expected to deliver more than twice the performance of the 21164 using the same 0.35-micron process, showing the advantage of a larger transistor budget.

The 21264 may also turn out to be the last high-end CPU optimized for a 0.35-micron process. The next wave of new high-performance CPU designs--UltraSparc-3, the MIPS H1, the PowerPC G4, and Intel's Merced--is expected to crest in 1998. Each of these chips is likely to consume a large 0.25-micron die with a bigger transistor budget than the 21264's. If any of these future chips takes full advantage of this transistor budget, it could pressure even an 800-MHz 21264 for the performance lead.

Digital's design team has done an excellent job in completing the 21264 design less than two years after the 21164's. Most other vendors are taking three to four years between major high-end CPU designs. To protect Alpha's performance lead beyond 1998, however, another new (or at least enhanced) core may be required. Digital's designers are already working on the 21364, designed to secure Alpha's performance lead sometime in 1999.

Setting New Standards

The 21264 is a strong design that will dominate all other processors in performance when it first appears. In addition to delivering SPEC95 performance well in excess of any current microprocessor, the 21264 sets the pace for system bandwidth as well. This bandwidth will ensure that the high performance of the 21264 core is actually delivered on a wide range of technical and commercial applications. Other processors will be challenged to match this bandwidth or suffer the consequences of an unbalanced design.

The 21264 CPU core does a fine job of making performance tradeoffs. Compared with the 21164, the new chip adds enormous complexity aimed at increasing the number of instructions executed per cycle. But not a single bit of clock speed was sacrificed to achieve these gains. Despite exceeding the out-of-order capabilities of all other announced microprocessors, the 21264 is slated to deliver twice the clock speed of any non-Alpha CPU in a comparable IC process.

To achieve this clock speed, Digital has added pipeline stages where necessary, extending the data-cache latency and the mispredicted branch penalty. The out-of-order design covers much of the former problem, while the advanced two-algorithm branch predictor handles the branch problem. As other processor designs approach the clock speed of Digital's CPUs, we expect they will adopt many of these same techniques, including multicycle caches and advanced branch-prediction methods.

The cost/performance tradeoff was consistently made in favor of performance at any cost. Maximum performance will require an expensive system design around the costly 21264 chip. The 21264 will support lower-cost systems using slower SRAMs and a less expensive memory subsystem, but these will not deliver the same level of performance.

Digital is addressing a similar problem today with the 21164PC (see page 4), a derivative of the 21164 designed for systems that sell for as little as $2,500. Systems using the 21264, even with low-cost memory chips, will not come close to that price point. Presumably, the company (perhaps in partnership with Samsung) is working on a 21264PC. This device would be a 0.25-micron derivative that could support a low-cost chip set and commodity memory chips. Such a device, however, is unlikely to appear before 2H98.

In the meantime, the 21264 will break SPEC95 records and maintain Digital's pre-eminent performance position. HP has threatened to surpass the 21264's performance when its PA-8500 debuts in 1H98 (see page 18), but a 0.25-micron 21264 should quickly subdue this threat. Digital may not be able to deliver another three years of unbroken performance leadership, but the 21264 should maintain Alpha at or near the top through 1998, keeping Digital's workstation and server customers happy.

Alpha Adds Video Instructions

For only the second time, Digital has made a significant modification to the Alpha instruction set. After adding byte loads and stores last year, the company announced a new set of motion-video instructions that will be incorporated into both the 21164PC (see page 4) and the 21264.

In typical minimalist fashion, Digital added only a few instructions to handle specific multimedia applications, rather than the more extensive modifications seen in Intel's MMX and Sun's VIS. The key changes are instructions for motion estimation and a few parallel 32-bit computations. The main focus of the new extensions is video encoding, as the current design is adequate for most other multimedia functions. We will provide a detailed description of the new Alpha instructions in our next issue.

Price & Availability

Digital has not yet announced price and availability for the 21264. It expects the chip to be in volume production by 4Q97 at speeds of at least 500 MHz. Contact Digital Semiconductor (Hudson, Mass.) at 800.332.2717 (508.628.4760 outside the U.S.) or access the Web at www.digital.com/info/semiconductor.

Website Administrator
Updated: 28 april 1997 15:14 mn