OBLIQUE PERSPECTIVE

Breaking Moore's Law

Divided, the P6 Stands; United, It Likely Would Flail

by John Wharton, Applications Research

The returns are in: the Intel P6 appears to be a most formidable device. Its main liability (in the view of some analysts) may be the partitioning of the design onto two die. If Intel had packed the same functions onto one chip, they reason, the conventional advantages of monolithic processor designs would have applied.

For sure, the P6 microarchitecture *(see **090201.PDF**)* represents a new milestone in the evolution of microprocessor design. I know of no other microprocessor that partitions the decoding, execution, and retirement functions in quite the same way, or that has as much headroom for future performance improvements.

But these are topics for a future column. A more mundane milestone set by the P6 is that it is the first mainstream microprocessor in which more than one die—a CPU and a second-level cache (L2C)—have been housed in a single package. Computer revolutions always begin with new packaging technology, and future generations (I think) will remember P6 as the first microprocessor to break the die-area bottleneck implicit in Moore's Law. The benefits of this are many.

Moore's Law Restated

Following the 1994 Intel shareholders meeting, a microprocessor newsletter columnist asked Intel Chairman Gordon Moore how his law was doing, and whether he saw any impending limits to future technology advances. His answer (best I can recall) was that he wasn't even sure what "his" law *was* anymore, that he couldn't recall ever having stated the law that bears his name.

Dr. Moore does have a fondness for tracking semiconductor technology trends as a function of time, such as the cost, size, or speed of a given logic function; the number of transistors on a given class of circuit; or the total acres of silicon processed by the industry each year. Regardless of what metric he plotted, Dr. Moore found, on semi-log graph paper the result was always a straight line.

This observation might be called "Moore's Meta-Law": Every aspect of the microelectronics industry improves by a consistent percentage each year. Figure 1 illustrates what's perhaps the meta-law's most popular corollary: the number of transistors on a leading-edge microprocessor doubles every two years or so.

The Intel 4004, introduced in 1971, contained 2,300 transistors. The original Intel Pentium processor, introduced in 1Q93, consumed 3.1 million. Plug in these endpoints, and the average compound growth rate is about 40% per year, for a doubling of transistor count every 25 months. Every new Intel microprocessor product announcement has fallen—with uncanny uniformity—along a line between the 4004 and Pentium. The one thing that hasn't changed in 25 years, though, is that every one of these processors was built on a single die.

Break #1: Getting Ahead of the Curve

The P6 CPU contains 5.5 million transistors. The L2C contains either 15 or 16 million transistors, depending on which Intel source you consult. (Funny, not too long ago the Intel i860 earned its 15 minutes of fame as the first microprocessor to contain a million transistors. These days, a million transistors can get lost in the rounding error.) The crosshair above the line in Figure 1 (labeled "P6/CPU+L2C") plots the P6's total of 21 million transistors against the product's expected 4Q95 introduction date.

Extrapolate the growth curve of Figure 1 and you find that the P6 chip set contains three times as many transistors as Moore's Law would have predicted. Put another way, it will be late 1998 (the circle to the right of the top crosshair) before it will be possible to build a microprocessor of P6's complexity using a conventional monolithic design. By splitting the P6 functions between

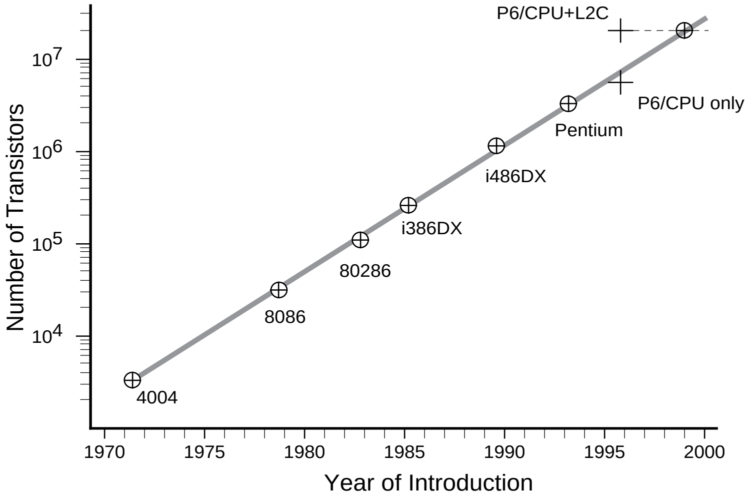

Figure 1. The number of transistors in Intel's leading-edge microprocessors has followed a remarkably consistent growth curve—until now.

two separate die, Intel has taken a three-year head start on Moore's Law.

Break #2: Eschewing the Bleeding Edge

A second thing to notice from Figure 1 is that the P6 CPU die in isolation is somewhat less complex than what history would have predicted. The crosshair below the line (labeled "P6/CPU only") shows the transistor count of just the CPU. By 4Q95, two and a half years will have transpired since Pentium first shipped. In that time, device complexity should have grown by 2.5×. Instead, the P6 CPU transistor count increased just 75%.

From a chip designer's perspective, this suggests the P6 CPU is significantly less aggressive than it might have been. At 8K each, the on-chip instruction and data caches are no larger than those of the now-mature Pentium, and they are collectively no larger than the unified I/D cache in the (486-based) IntelDX4.

Indeed, at a March dinner presentation, P6 architect Dr. Robert Colwell told *Microprocessor Report* subscribers that some of his early goals for the part had been trimmed back a bit. The on-chip instruction cache, reorder buffer size, and instruction decoder sophistication were all supposed to have been greater, but were downsized (Colwell said) when transistor budgets got tight.

Figure 1 implies that by conventional standards, raw transistor count shouldn't have been a problem. Doubling the size of the on-chip I-cache would only have added 400,000 transistors, and the part would still have been buildable. It's more likely that CPU complexity was intentionally constrained for business reasons. Even a 4% reduction in die area would likely increase manufacturing yield and lower die cost by 10%.

This conflicts with the conventional wisdom of microprocessor design, which is to pack as much cache onto the CPU as is technologically feasible, in order to minimize the impact of first-level cache misses. In the case of the P6, however, Intel's designers could get away with relatively small on-chip caches because putting a large, fast, tightly-coupled L2C within the same package as the CPU greatly ameliorates the cache-miss penalty.

Put elsehow, combining two die within one package made it possible for the P6 CPU lag the technology curve defined by Moore's Law by nearly a year, with minimal effect on performance. On the day P6 is introduced, Intel should have no more trouble building the part than it had with earlier processors a full year after introduction.

Break #3: System Revenue Hegemony

A third break (to Intel) is that partitioning the P6 as it did migrates new system-logic functions from the PC motherboard onto an Intel product. Intel has long followed the strategy that adding features to a chip enhances its perceived value to buyers, increasing the amount they should be willing to pay for the part. Early microprocessors contained just a CPU. The i486DX augmented its integer core with an FPU, cache, write buffers, parity circuits, and other system logic. Pentium further enhanced the cache and added system-integrity functions, while the P54C included power-reduction circuits and an APIC-compatible interrupt controller for glueless multiprocessing.

To hit a particular end-user price point, system designers must work within a fixed component-cost budget. Intel sees each dollar spent on support logic as a prize to be won; the less OEMs spend on cache SRAMs and glue, the more they can afford to give Intel.

At the chip level, the P6 provides a "front-side" system bus and a separate "back-side" bus just for the L2C. The i960 M-series embedded processor family (introduced in 1990) has a similar system architecture—not surprising, since the same Intel engineer defined the interfaces of both products—as does the NexGen Nx586, MIPS R4000, and various other CPUs. In each case, though, cache RAM sales go to competing chip vendors.

As long as the CPU and off-chip caches are packaged separately, system designers have the option of implementing cache functions externally with whatever approach costs the least. And since SRAM technology is one area in which Intel is not an industry leader—the company builds its SRAMs on fab lines optimized for CPUs and random logic, and still uses six transistors in each memory cell when the rest of the industry needs only four—the off-chip cache is one budget item Intel would likely have been forced to leave on the table.

Putting the second-level cache array into the same package as the CPU solves this problem. With P6, whatever dollars an OEM would otherwise have budgeted for cache SRAMs will now flow directly into Intel's coffers.

Break #4: Socket-Compatible Scalability

A question raised by many P6 watchers concerns whether Intel plans to offer a version of the P6 with an outboard L2C. Officially, Intel has left this option open, but I don't think it makes sense. Bonding out the back-side bus to external pins would force Intel to redesign the CPU to include I/O buffers, level shifters, and protection circuitry on these pads, would mandate a new, incompatible pinout, and would increase chip-crossing delays. External cache logic would be hard pressed to keep up at even 133 MHz and would stall as the P6 core frequency rose to 200 or 266 MHz.

Moreover, the elimination of the on-package L2C would not necessarily save much. The cost of even a 256K L2C is but a fraction of that of the CPU, about 40% according to MDR's Cost Model. Even though it contains many more transistors, the L2C die is just two-thirds as large, and fabrication costs for a die this size vary with the second or third power of its area. *(For a description of the MDR Cost Model, see* **071004.PDF** *and* **081203.PDF***)*. A

128K L2C with half the area of the current design would cost Intel almost nothing to build. Indeed, if Intel sells "fall-out" parts (L2C die in which half the array is bad), these chips would literally be free.

So retaining a small L2C on-package might give Intel a "K5 killer," with the strategic advantage of preserving the pinout of the original device. PC motherboards could then provide a common socket for a full range of processor implementations, and low-end systems could be upgraded in the field by swapping out the original CPU for a device with a full-size cache—another win for Intel. In multiprocessing configurations, CPUs with a local L2C need never contend for discrete external SRAM. Moreover, as faster processes come on line, L2C speeds will scale to match the CPU; existing sockets will provide a good home for the faster parts, whereas discrete cache would eventually become frequency limited.

Break #5: A Lock on System Differentiation

Another strategic advantage of placing the CPU and L2C in a single package is that it limits a system designer's options for product differentiation. This in turn gives Intel more flexibility in waging its marketing wars.

PC vendors have historically offered a range of product implementations. Even with the same CPU type and frequency, clever system designers could create entry-level, midrange, and high-end products by altering such external elements as whether or not a system contained a second-level cache array, its cache size and configuration, and the main-memory bandwidth.

The P6 eliminates memory-system sophistication as an avenue for product differentiation. At any given core and bus frequency, different P6 systems should all deliver remarkably similar levels of performance. The front-side bus is designed to run at one-half, one-third, or one-fourth the frequency of the CPU core; it may take several of these reduced-frequency bus cycles to post a request to system memory, and several more to absorb returned values; and the P6 transaction-oriented bus allows the initiation and completion of system-bus transfers to themselves be separated in time. As a result, on-package (as opposed to on-chip) P6 performance is tremendously decoupled from off-package accesses.

In other words, there's little a system designer can do to either improve or degrade memory-system performance. Consider two radically different P6-based designs: an entry-level PC that runs directly from slow, noninterleaved, commodity DRAMs, and a high-end server with a fast main memory and an infinitely large zero-wait-state third-level cache. My guess is that both would deliver about the same memory-system throughput. The biggest factor in determining how fast each system would run, then, would be how often each must access the outside world, which depends in turn on the size of the on-package cache.

With a given core frequency, then, the best way for a PC vendor to differentiate P6 system performance levels might be to purchase processors with different sizes of L2C. Intel has said it plans to introduce a version of the P6 with a 512K L2C when the family moves to a 0.35-micron process in 1996. But at an early P6 briefing, Contributing Editor Brian Case noticed that the product code stamped on P6 samples already includes the notation "256K." Historically, Intel has not attached distinguishing suffixes ("SX," "DX," "DX2") to its product codes until there actually existed multiple incarnations of a product among which differentiation was needed. My guess is that even before 0.35-micron devices are in production, Intel may begin offering 0.6-micron P6 products with a different—read "smaller"—L2C. The ability to pair standard CPUs with L2C chips of varying size wouldn't be possible with a monolithic design.

Break #6: Load-Balancing Production

But perhaps the most compelling reason (to Intel) for splitting the P6 design between two devices is that it gives the company unprecedented flexibility in fine-tuning its production capacity to meet demand.

Intel is first and foremost a manufacturing company. The manufacturing cost of a state-of-the-art microprocessor is determined primarily by the cost of the equipment needed to build it, divided by the number of devices that can be sold before the equipment becomes obsolete. To a first-order approximation, marginal fab costs (silicon, dopants, labor) are zero. Since 1991 Intel has sunk more than $6 billion in new "megafab" production lines for 0.6-micron and smaller geometries. For this investment to pay off, Intel must balance supply and demand such that these fabs run at very near full capacity.

Suppose Intel's fab lines have a defect density of about 0.6 defects per cm^2. Then according to the MDR Cost Model, a single 8" wafer would yield about 14 good P6 CPU die, or 32 good die for the 256K L2C. (These are rough estimates; actual defect rates and die yields are a deep, dark secret. For the purpose of this discussion, though, absolute yield is less important than the yield ratio, which depends less critically on defect rates.)

Under these assumptions, Intel must build about 2.3 CPU wafers for every L2C wafer it processes in order for the numbers to come out even. If demand for P6 CPUs is high enough to keep Intel's fabs running at 70% of capacity, Andy Grove would be a very happy man; the remaining 30% would neatly satisfy the corresponding demand for 256K cache chips.

Now suppose the P6 market is slow to develop, and only demands, say, 30% of Intel's capacity for CPUs, and another 13% for the L2C. More than half of each plant would sit idle, depreciating quietly away. Here's where the magic of silicon economics comes in: Intel could then shift cache manufacturing to a larger 512K part. Yields might be significantly lower—as low as six die per wafer, say, one-fifth that of the 256K part—so the entire 70% of capacity not needed for P6 CPUs might have to be given over to L2C production. Still, Intel's total costs (in round numbers) would be no higher than if the same manufacturing equipment sat idle. In effect, the two-times-larger L2C die would cost nothing to build, yet would spur increased interest and command a higher sales price. Memory for nothin', and the chips for free.

Or suppose P6 demand is higher than expected. By reducing the L2C to 128K bytes, its yield would increase to about 122 good die per wafer—four times higher than the 256K flavor. Intel could then allot a full 90% of its wafer starts to CPU chips, and reserve just 10% for the requisite L2Cs.

And if Intel ever finds itself production limited, with the market willing to buy however many P6 CPU die its megafabs crank out? Well, the company is already negotiating with outside SRAM vendors to take over cache-chip production entirely at some point down the road. This would not be possible with a monolithic CPU.

Smart Move

Historically, it has not been cost-effective to put more than one complex IC in a single package. During burn-in and parametric testing, a significant fraction (perhaps 5%) of packaged devices fail. If two chips are packaged together, the chip failure rate would double, and both die would have to be discarded, increasing the cost of wastage fourfold. What now makes dual-chip processors financially practical is Intel's SmartDie program *(see* ***0809MSB.PDF****)*, under which multiple-chip modules can be assembled from bare die that have already survived the rigors of burn-in and testing.

It should be noted, though, that the P6 is not a "multichip module," nope, no-sirree, it's *not* an MCM. Intel seems to be very defensive on this point; at every P6 presentation and briefing to date Intel has made it abundantly clear that the part is merely a two-chip processor inside a dual-cavity package. The difference, as Tom Halfhill noted in the April 1995 issue of *Byte* magazine, is that MCMs have a reputation of being a riskier, more expensive packaging technology, and Intel would rather portray the P6 as a reliable, high-volume part.

Methinks Intel doth protest too much. Perhaps the real reason the company appears to be trying so hard to build a case for its new system partitioning is that if other vendors understood the true benefits of partitioning processors as a two-chip module, Intel's competitors would begin breaking Moore's Law as well. ♦

John Wharton is the editor and primary author of The Complete x86: The Definitive Guide to 386, 486, and Pentium-Class Microprocessors. *Contact MicroDesign Resources for information on this and other Technical Library reports.*