Arthur Revitalizes PowerPC Line

First G3 Processor Improves Integer Performance with Low Cost and Power

by Linley Gwennap

With their first so-called G3 processor, Motorola and IBM have achieved an outstanding combination of high performance and low cost. The chip, code-named Arthur, combines features from the PowerPC 603, 604, and 620 to significantly improve performance, particularly on Macintosh applications. On a clock-for-clock basis, these changes improve Mac performance by more than 90% over the 603e and 40% over the 604e, according to the vendors' Somerset design center. But by using a leading-edge 0.25-micron process, the new design measures just 67 mm^2, much smaller than the 604e and slightly smaller than the smallest 603e.

The new process also boosts clock speed. At the recent ISSCC, a Somerset representative discussed a 250-MHz version of the chip. Unlike Intel's 400-MHz Klamath, however, this clock speed is quite realistic; in fact, it is probably conservative. Arthur is based primarily on the 603e, which already reaches 240 MHz in a 0.33-micron process; the new chip could significantly exceed 250 MHz when it reaches production.

At 250 MHz, Arthur will deliver 10.5 SPECint95 (base), according to Somerset estimates. This score exceeds the integer performance of the fastest PowerPC chip available today, the 604e-225. On the floating-point side, however, the new chip's 7.7 estimate lags that of the 604e slightly. Thus, Arthur may largely displace the current 604e from Apple's products, but the more expensive chip is likely to remain predominant in IBM's workstation lineup due to its FP prowess.

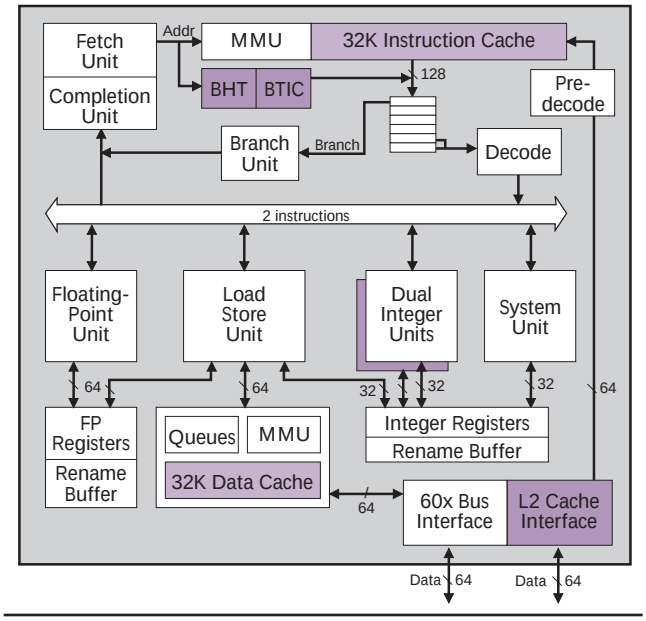

Figure 1. Arthur adds larger caches, a second integer ALU, dynamic branch prediction, and an L2 cache bus to the original PowerPC 603 design.

The vendors expect Arthur (the actual product name will be announced later) to appear in systems by midyear, making it likely to be the first or second commercial microprocessor (possibly after the R10000) to ship in a 0.25-micron process. The new process, called PPC3 by Motorola and CMOS-6S2 by IBM, combines a 0.25-micron transistor with metal layers more typical of a 0.35-micron process *(see* **101203.PDF***)*. The companies plan to ship a true 0.25-micron process in 2H97, about the same time as Intel and other major vendors. If Motorola can deliver on this schedule, it will achieve process parity with both Intel and IBM after lagging in this area for years.

Backside Bus Aids Performance

The biggest performance boost in the new design comes from changes not to the CPU core but to the system interface. Current PowerPC processors use the 64-bit 60x bus to access both the L2 cache and main memory. At CPU speeds in excess of 200 MHz, this bus begins to run out of gas, forcing the CPU to frequently stall while waiting for data.

To fix this bottleneck, Somerset added a new L2 cache bus, or backside bus. This approach is used in the unreleased PowerPC 620 as well as in chips from several other vendors. As Figure 1 shows, the new bus is 64 bits wide. The cache can be configured for SRAMs of various types and speeds, although typical implementations will operate at two-thirds or one-half of the CPU clock speed. In the faster mode, the cache provides up to 1.3 Gbytes/s of peak bandwidth, compared with 533 Mbytes/s for the 60x bus. Considering the 60x processors must share that bandwidth with main memory, Arthur will have three to four times the usable cache bandwidth of its predecessors.

According to Somerset, the backside bus alone increases performance by at least 50% at 200 MHz. At higher clock speeds, the L2 bus grows more valuable, since the 60x bus becomes less adequate for faster CPUs. The new chip supports a variety of cache sizes and speeds, allowing flexible system configurations; the performance impact, of course, will be smaller for the less expensive caches.

Arthur provides address and control signals for the L2 cache and includes the L2 cache tags on the CPU chip. This design limits the external components to just the data RAMs. The on-chip tags support external caches of 256K, 512K, and 1M. By doing the on-chip tag lookup before accessing the external data RAMs, the L2 cache can be two-way set associative, improving its hit rate.

The new bus requires expanding the BGA package to 360 leads, compared with 255 for the 603e. The new package slightly increases manufacturing cost. It also requires new motherboard designs. Both of these are small costs to pay for the extra performance. To support existing motherboards, the vendors will offer Arthur in the same package as the 603e and 604e; this version will lose the performance of the backside bus but still offer higher clock speeds and a faster core CPU, at least for Mac applications.

Core Changes Target Mac OS

Arthur represents the first time Somerset has optimized a new core based primarily on extensive execution of Mac OS applications. The 601, 603, and 604 designs were essentially complete before the first Power Macintoshes shipped; these designs were based mainly on simulations of SPEC and code fragments from the Mac. Since then, Somerset engineers have analyzed a broad variety of Mac (and other) software running on various PowerPC processors to determine the strengths and weaknesses of each design.

Arthur started with the basic core and pipeline of the 603, but the analysis pinpointed several areas for improvement. Although the 603 core can issue two instructions per cycle plus a branch *(see* **071402.PDF***)*, only one of these instructions can perform integer arithmetic; the other must be a load, store, or floating-point instruction. Code analysis found many cases where two successive integer operations could not be executed in parallel because of this restriction.

To solve this problem, Arthur (like the 604) includes a second integer ALU, as Figure 1 shows. This unit can handle basic arithmetic, logical, and shift instructions but does not include a multiplier. With two integer ALUs, the number of situations in which the CPU cannot issue two instructions due to resource conflicts is greatly reduced.

The increased throughput of the CPU core puts more pressure on the instruction cache. The 603 fetches two instructions per cycle from the instruction cache, preventing the chip from sustaining its peak rate of three instructions per cycle for more than a few clock ticks at a time. To remove this bottleneck, Arthur fetches four instructions per cycle from the cache. This change allows the CPU to sustain two instructions plus one branch per cycle in the best case.

Another bottleneck found during code analysis was in TLB miss handling. Mac applications generate more TLB misses than software for AIX, the operating system that had provided most of the traces for optimizing the original PowerPC processors.

To speed the handling of these misses, Somerset added a hardware tablewalk feature to Arthur. This feature, found in many other modern CPUs, allows the processor to directly access the virtual page tables and fetch the missing translation. This change significantly reduces the average number of cycles for handling a TLB miss, resulting in an improvement in Mac application performance of about 6%.

For More Information

Neither Motorola nor IBM has announced pricing, availability, or the actual product name of the processor known as Arthur. A copy of the vendors' ISSCC paper is at *www.mot.com/pub/SPS/PowerPC/library/tech_sum.*

Like the 620, Arthur stores predecoded instructions in its on-chip cache to maintain a short pipeline at high clock speeds. Arthur uses four predecode bits per instruction instead of the seven bits in the 620.

BTIC Provides Zero-Penalty Branching

Code analysis also showed branches to be a problem in the 603. Mac applications, like PC applications, have more frequent branches than technical workstation software. This makes efficient branch handling important, but unlike most modern CPUs, the 603 has no dynamic branch prediction. The penalty for mispredictions can be as little as two cycles, but the 603's static prediction algorithm is incorrect 25–35% of the time on many applications. To sustain high performance on Mac software, this figure needed to be reduced.

Arthur adopts a dynamic prediction method similar to that used in the 604 core. Both chips have a 512 × 2-bit branch history table (BHT) that predicts the direction of branches. This method should reduce mispredictions to about 20%. Surprisingly, the designers did not choose to further reduce mispredictions by increasing the BHT size to 2K entries or by utilizing a more modern two-level BHT *(see* **090405.PDF***)*. These improvements would have had minimal die area impact in a 0.25-micron process, but Somerset claims the performance impact of such a change would have been minimal as well.

The new chip does improve on the 604 in accessing branch targets. The 604 contains a 64-entry branch target address cache (BTAC) that contains the predicted target address of 64 branches. This address is then fed to the instruction cache to fetch the target. In Arthur, the BTAC is replaced by a branch target instruction cache (BTIC) that holds the first two instructions located at the target.

At the end of the first pipeline stage, both the sequential instruction path and the target instructions are available for use. If the branch condition can be resolved in the second pipeline stage, which is frequently the case, the correct set of instructions can be routed through the pipeline regardless of the original prediction. In other words, branches that are resolved early have no misprediction penalty.

PowerPC is a condition-code architecture; for the branch to be resolved early, the condition must be precomputed. For most code, the only time the branch resolution is delayed beyond the second pipeline stage is when a long-latency event (such as a cache miss) delays the instruction computing the condition code.

	PowerPC 603e	Arthur	PowerPC 604e	Exponential x704
Clock Speed	240 MHz	250 MHz	225 MHz	533 MHz
Cache Size	16K/16K	32K/32K	32K/32K	2K/2K/32K
Issue Rate	2 instr + branch	2 instr + branch	4 instr	2 instr + branch
Integer ALUs	one ALU	two ALUs	two ALUs	one ALU
FP Multiplier	32 bits	32 bits	64 bits	32 bits
Branch Pred	static	512 BHT, 64 BTIC	512 BHT 64 BTAC	256 BTB
IC Process	0.33µ 4M CMOS	0.25µ 5M CMOS	0.33µ 4M CMOS	0.5µ 5M BiCMOS
Die Size	78 mm^2	67 mm^2	148 mm^2	150 mm^2
Transistors	2.6 million	6.4 million	5.1 million	2.7 million
Est Mfg Cost*	$30	$40	$60	$90
Max Power	6 W	5 W	20 W*	85 W
SPECint95†	5.5 int*	10.5 int	9.0 int*	12 int
SPECfp95†	4.0 fp*	7.7 fp	8.5 fp*	10 fp
Mac Perf‡	1.0	2.1	1.3	not avail
Availability	3Q96	2Q97	3Q96	2Q97

Table 1. With its variety of improvements, Arthur is more like the 604e in features and performance, but the new chip most resembles the 603e in die size, cost, and power dissipation. †baseline ‡based on a Somerset tool designed to estimate Macintosh application performance (Source: vendors except *MDR estimates)

The zero-penalty branching works only if the branch hits in the BTIC. Because the BTIC has far fewer entries than the BHT, some branches will be correctly predicted but will miss in the BTIC. In Arthur, these branches create a one-cycle "bubble" in the instruction queue. This bubble will not affect performance if there are at least two instructions already in the instruction queue, since the four-word instruction fetch can quickly refill the queue.

Larger Caches Aid Performance

A final performance boost comes from Arthur's larger on-chip caches. The new chip includes 32K of instruction cache and 32K of data cache. This is twice the amount of cache found on the 603e and four times the cache size of the original 603. The 603's tiny 8K caches were notoriously poor for Mac OS software, particularly for 68K emulation; even the 603e's caches cause a significant performance hit at higher clock speeds. Given Arthur's design target of 250 MHz and up, doubling the caches again made sense.

Somerset estimates the larger on-chip caches alone add about 15% in performance for typical Mac applications, compared with the 603e. The changes to the core—the extra integer unit, improved fetch rate, hardware TLB miss handler, and dynamic branch prediction—add another 15% or so. The biggest performance gain comes from the new L2 cache bus; this change raises performance by 50% or more.

These changes make Arthur very similar to the 604e, as Table 1 shows. Both chips have 32K of instruction cache and 32K of data cache; both have two integer ALUs, a floating-point unit, a load/store unit, and a branch unit; both have dynamic branch prediction. The 604e can execute up to four instructions per cycle, one more than Arthur, and has more extensive out-of-order execution to increase parallelism.

Apparently, these extra capabilities are not particularly helpful on many applications. Somerset estimates that, even without using the new L2 cache bus, Arthur is only 6% slower than the 604e on SPECint95 when running at the same clock speed. This comparison indicates that the differences in the two CPU cores have a minimal effect on this benchmark. When tested on Mac OS applications, Arthur is actually 16% faster than the 604e when the changes to the L2 cache and clock speed are neutralized, due to its shorter pipeline and zero-penalty branching.

The 604 core has an important advantage in its double-precision floating-point multiplier. Arthur retains the single-precision multiplier of the 603, which requires extra cycles for DP operations. As a result, the 604 core is 33% faster than Arthur's core on SPECfp95, which makes heavy use of DP multiplication. Even taking Arthur's L2 cache bus and 20% faster clock speed into account, the 604e appears to be a better solution for double-precision floating-point applications.

Like the 603e, Arthur is limited to uniprocessor or simple dual-processor systems by its cache consistency protocol. Thus, the 604e will continue to be used in multiprocessor systems. Apple, however, offers few MP systems, and IBM typically uses its internally designed CPUs for MP systems. For these reasons, we expect Arthur to rapidly displace the current 604e except in IBM's desktop workstation products and a few multiprocessor systems.

Power Low Enough for Notebooks

Although the new chip will exceed the performance of the 604e on most applications, it retains the low power dissipation of the 603e. The maximum power dissipation at 250 MHz is just 5 W, the same as for a 603e-200. Arthur retains the same 2.5-V power supply for the CPU core and 3.3-V I/O as that chip; the power required by the core changes and extra cache is balanced by the decreased requirements of the smaller transistors and shorter traces.

These advantages are gained by the move to the new 0.25-micron five-layer-metal process. The Arthur die measures just 67 mm^2, compared with 78 mm^2 for the 0.33-micron 603e. As Figure 2 shows, the caches consume a large portion of the die, nearly as much as the CPU core. The chip contains 6.35 million transistors on this small die. According to the MDR Cost Model, the manufacturing cost of Arthur should be just $40 in the CBGA-360 package.

The vast majority of the transistors are in the caches. The on-chip L2 tags, level-one data cache and tags, sense amps, and cache-control logic consume about 5.3 million transistors. The CPU core has about 1.0 million transistors, 50% more than in the 603 core but still only about half the complexity of the 604 core.

In 1998, Motorola and IBM will probably move the Arthur design into their next process, called PPC4 by Motorola and CMOS-6X by IBM. This process, a true 0.25-micron

process with tighter metal pitches than PPC3, will further reduce power by moving the core voltage to 1.8 V. This change will also make the die size even smaller and boost clock speeds, perhaps to 350 MHz.

With its modest power requirements, even the initial Arthur chip will fit right into existing Apple notebook designs. The motherboard must be modified to take advantage of the new L2 cache bus, although Apple will probably build at least some notebooks by dropping the chip into existing motherboards and ignoring the L2 cache bus. The PowerPC chips have roughly half the maximum power dissipation of Intel's mobile P55C processors, extending battery life and making Apple's thermal design efforts easier than those of PC notebook vendors.

Head-to-Head With Klamath

A bigger contrast can be drawn with the Klamath processor *(see* **110201.PDF***)*, which dissipates far more power than a notebook system can handle. Arthur should match the performance of Klamath while fitting into a mobile system, allowing Apple to deliver the fastest notebook systems in the world. Although this gap is likely to narrow in 1H98 when the first P6 notebooks appear *(see* **110202.PDF***)*, if Apple executes well it could have a 6–12 month advantage in notebook system performance compared with Intel-based systems.

On the desktop, the power issue is less important. Here, Arthur will compete head-to-head with Klamath. Until the products are introduced this spring, a precise performance comparison is impossible, but it looks as if the two chips will offer similar integer and floating-point performance. Arthur may have an advantage on single-precision floating-point, popular in 3D graphics software. With MMX, Klamath may have an advantage on some multimedia applications. In summary, Arthur can match Klamath but is unlikely to open a significant performance gap.

Arthur's low manufacturing cost, however, lets IBM and Motorola continue to undercut Intel's prices. We expect Klamath to initially debut at a list price of $700–$800. In contrast, Arthur is likely to appear at $400–$500. Apple will pay far less, of course, while even Intel's best customers don't get much of a discount off list. Intel will bring down the price of Klamath over several quarters, but Arthur's cost structure will easily support a price well below Klamath's.

This positioning appears advantageous but is unlikely to boost PowerPC's prospects versus Intel. Without even getting into Apple's problems *(see* **1101ED.PDF***)*, a major failure of PowerPC is that it has never been able to deliver a large performance advantage over Intel. Although lower CPU prices are impressive from a technical standpoint, the cost savings are usually eaten up by higher system margins and component costs. Arthur keeps pace with Intel but doesn't appear to change this basic equation. As noted, Apple's one opportunity for performance leadership will come in the notebook market, but the company must respond quickly when this opportunity knocks.

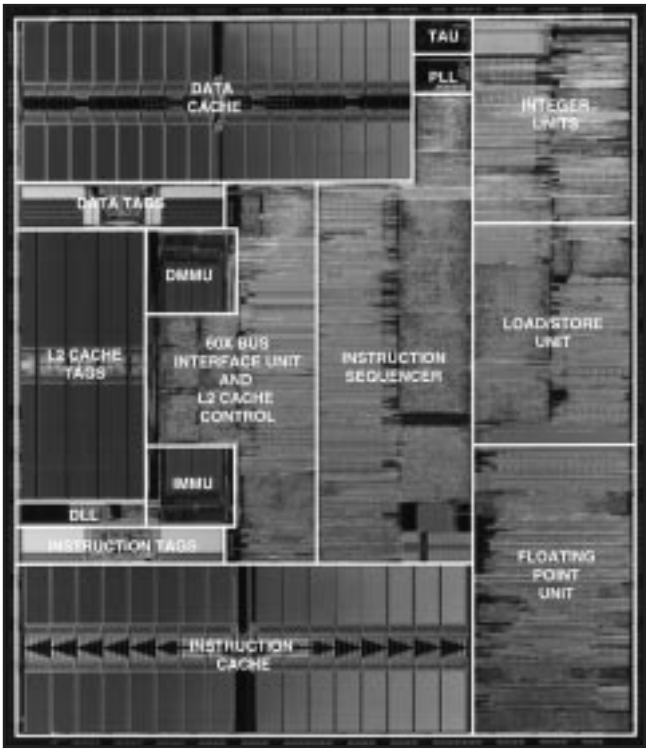

Figure 2. Somerset's Arthur design contains 6.35 million transistors and measures 7.6 × 8.8 mm when fabricated in a 0.25-micron five-layer-metal CMOS process.

Rocking the Mac World

Within the Mac OS world, Arthur is an impressive part that should displace the current 604e and even some 603e products from Apple's lineup. An interesting question is how Apple will position Arthur against the forthcoming 533-MHz x704 from Exponential *(see* **101401.PDF***)*. SPEC95 estimates from the vendors indicate that the x704 may beat the 250-MHz Arthur on integer performance, as Table 1 shows. But Exponential's tiny (2K) primary caches may fare poorly on Mac OS applications. When the two chips are tested in Macintosh systems, Arthur may match or even exceed the application performance of the x704.

Most consumers, however, don't look at benchmark results. The most impressive number on the Exponential chip is 533, its clock speed. We believe Apple will use this number to position x704-based systems at the high end of its line, although users may see little benefit to "upgrading" from Arthur to the x704. If end users don't fall for this positioning, the x704 may have no room in Apple's lineup.

The strong performance of the new Somerset chip may eliminate the need for the Exponential chip entirely. Arthur's ability to achieve this performance with such a small die and low power consumption makes the design truly exceptional. The new chip is reminiscent of QED's MIPS designs: lots of cache, high clock speed, relatively simple CPU. Arthur reminds us that much of the complexity in high-end CPUs is unnecessary for many applications and that there is virtue in simple, fast microprocessors. M