



APPLE'S
DROPPING THE
POWERPC AND MOV-
ING TO X86. MICROSOFT'S
DROPPING X86 AND MOV-
ING TO THE POWERPC, JOIN-
ING NINTENDO AND SONY.
WHICH COMPANY IS RIGHT,
OR ARE THEY ALL RIGHT?
AND WHAT'S THE BEST
CHOICE FOR YOUR
DESIGN?

Go to www.edn.com/050915cs
for tables that detail all of the
benchmarking data.

BY BRIAN DIPERT • SENIOR TECHNICAL EDITOR

Mac (under) the knife: piecing together the PowerPC puzzle

Three notable events in the high-tech industry so far this year have particularly influenced the status of the PowerPC CPU in embedded-system applications and the broader electronics market. Beginning in mid-March, IBM published a series of articles promoting the Apple Mac Mini as an embedded hardware- and software-development platform, both under the Mac OS and with various iterations of Linux and BSD Unix (**references 1 and 2**; see **sidebar** "Linux: a work in progress"). In mid-May, all three next-generation game consoles (Microsoft's Xbox 360, Nintendo's Revolution, and Sony's PlayStation 3), containing various spins of the PowerPC architecture, went public at the E3 Expo. And, in early June, Apple Chief Executive Officer Steve Jobs announced that the company would begin a phased transition away from the PowerPC to Intel x86 CPUs (**references 3 and 4**; see **sidebar** "New PowerPC flavors").

The diverging game-console-versus-computer trends have provoked a vigorous cyberspace and water-cooler debate regarding the future of the PowerPC, both absolutely and relative to its chief 32- and 64-bit competitor, the x86. Is the Mac Mini, as IBM's documentation claims,

a valid development vehicle for PowerPC-based embedded-system designs? What does next-generation game-console developers' embrace of the PowerPC architecture indicate about its cost, performance, power consumption, and other attributes? And, conversely, does Apple's embrace of x86 suggest that the PowerPC is at a crossroads compared with AMD-, Intel-, and Via-supplied CPU alternatives? These are some of the questions that the development efforts, benchmark results, and other issues raised and resolved in this hands-on project attempt to explore.

THE HARDWARE

Following the recommendations outlined in IBM's literature, *EDN* purchased a 1.25-GHz Mac Mini, with the SuperDrive writable-DVD-drive option, for \$553 after rebate (**Table 1**). Stay tuned for a product teardown in a future *EDN* Prying Eyes section. The Mac Mini's specifications are similar to those of a PowerBook that's also in the *EDN* computer pool, giving credence to the oft-

AT A GLANCE

Recent events put an added twist on the long-standing x86-versus-PowerPC debate.

Four hardware platforms, along with multiple subsystem combinations for one of them, give numerous data points for a more complete picture.

SPEC data is, for now, integer-specific and does not reflect multi-processor presence.

Xbench testing provides highly detailed results, some of which are not readily explainable.

The project lives on; keep an eye out for future print articles and blog postings.

touted observation that (simplistically speaking) Apple repackaged an iBook laptop and removed the LCD from it to come up with the Mac Mini (**Figure 1**). The 32-bit G4 (PowerPC 74xx) CPUs in both systems run at 1.25 GHz, with a 167-MHz FSB (front-side-bus) frequency and no L3 cache. The core clock speed of the CPUs in the Mac Mini and PowerBook exceeds that of many PowerPC-based embedded designs. To give the project additional relevance to *EDN* readers, therefore, we also acquired a four-year-old

G4 Power Mac through a winning bid on an Ebay auction.

The dual 800-MHz G4 Power Mac system (code-named Quicksilver) showcased in this article has half the L2 cache of its Mac Mini and Powerbook G4-based counterparts but contains 2 Mbytes' worth of off-CPU L3 cache consisting of synchronous SRAM running at one-quarter the core CPU-clock rate. The cost-cutting L3-cache omission on the Mac Mini and PowerBook is perhaps more understandable when you realize that their SDRAM runs at a higher peak data rate than the L3 cache in the dual G4 Power Mac can support! This first-generation Quicksilver system, which runs its single-data-rate SDRAM at a PC100 speed setting, also came in 733- and 867-MHz single-CPU variants; second-generation Quicksilver systems delivered 800- and 933-MHz (single-CPU) and 1-GHz (dual-CPU) speeds and underwent a minor L3-cache evolution, switching to DDR (double-data-rate) SRAM that ran at half the CPU core clock speed. Follow-on "Mirrored Drive Door" systems marked the end of the line for the G4 Power Mac series; they further evolved the memory subsystem by switching from SDRAM to DDR SDRAM.

This project also harnessed a dual 1.8-GHz G5 Power Mac, at the other end of the performance spectrum

from the dual G4 Power Mac. This second-generation iteration of the platform includes PCI slots along with a four-DIMM architecture; the first-generation dual 1.8-GHz G5 Power Mac supported PCI-X add-in cards and included six DDR SDRAM slots. This system's FSB speed is half the core clock frequency (in this case, 900 MHz), whereas the single-CPU 1.8-GHz G5 Power Mac ran the FSB at one-third the core clock rate, presumably to maximize the test yield for the CPU and other system components. The G5 (PowerPC 970FX) CPU supports 64-bit addresses and data, in contrast to its 32-bit G4 predecessor, as well as a full-duplex FSB. Other enhancements include a larger, faster L1-cache subsystem, faster L2 cache (as with the Mac Mini and PowerBook, not supplemented by costly L3 cache), DDR400 SDRAM, and dual SATA drives in a RAID (redundant-array-of-independent-drives) 1 configuration.

Version 4.2.0a15 of Apple's CHUD (Computer Hardware Understanding Development) tools allows the user to disable nap mode on the PowerPC CPU. (We disabled nap mode for all of the benchmarks in this project.) On the G4 systems, it can also disable the L2 cache. Previous versions of CHUD reportedly could also disable the L3 cache in the G4 Power Mac, but Version 4.2.0a15 appears to



Figure 1 Apple's 1.25-MHz Mac Mini (a), 1.25-GHz G4 PowerBook (b), dual 800-MHz G4 Quicksilver Power Mac with GeeThree's Sweet Multiport installed (c), and dual 1.8-GHz G5 Power Mac (d) underwent intensive benchmark scrutiny during this hands-on project.

TABLE 1 SYSTEM SPECIFICATIONS

	G4 Mac Mini	G4 PowerBook	Dual G4 Power Mac (Quicksilver)	Dual G5 Power Mac
Processors	1.25-GHz PowerPC 7447A Version 1.2	1.25-GHz PowerPC 7457 Version 1.1	Two 800-MHz PowerPC 7450 Version 2.1	Two 1.8-GHz PowerPC 970FX Version 3.0
Front-side bus	167 MHz (32-bit, half-duplex)	167 MHz (32-bit, half-duplex)	133 MHz (32-bit, half-duplex)	900 MHz (64-bit, full-duplex)
L1 cache	32-kbyte instruction and 32-kbyte data, 1.25 GHz	32-kbyte instruction and 32-kbyte data, 1.25 GHz	32-kbyte instruction and 32-kbyte data (per processor), 800 MHz	64-kbyte instruction and 32-kbyte data (per processor), 1.8 GHz
L2 cache	512 kbyte, 1.25 GHz	512 kbyte, 1.25 GHz	256 kbyte (per processor), 800 MHz	512 kbyte (per processor), 1.8 GHz
L3 cache	NA	NA	2 Mbyte (unified), 200 MHz	NA
DRAM	128-Mbyte (Nanya), 256-Mbyte (Hynix), and 1-Gbyte (Micron) DDR400 CL3 SDRAM, and 512-Mbyte (Samsung) DDR333 CL2.5 SDRAM (all run at DDR333 CL2.5 speeds in-system)	1.5-Gbyte DDR333 CL2.5 SDRAM (one 1-Gbyte SoDIMM from Kingston and one 512-Mbyte SoDIMM from Samsung)	1.5-Gbyte PC-133 SDRAM (one 512-Mbyte DIMM from Hynix and two 512-Mbyte DIMMs from Samsung)	3-Gbyte DDR400 CL3 SDRAM (two 512-Mbyte DIMMs from Corsair and two 1-Gbyte DIMMs from Samsung)
Hard-disk drive	100-Gbyte Seagate 4200-rpm and 5400-rpm, 2.5-in. PATA, both with 8-Mbyte cache	80-Gbyte Hitachi 4200-rpm 2.5-in. PATA, 2-Mbyte cache	40-Gbyte IBM 7200-rpm 3.5-in. PATA-100, 2-Mbyte cache	Two 250-Gbyte Maxtor 7200-rpm, 3.5-in. SATA-150, one with 8-Mbyte cache and the other with 16-Mbyte cache, RAID 1 configuration using SoftRAID
Graphics	32-Mbyte AGP ATI Radeon 9200	64-Mbyte AGP ATI Mobility Radeon 9600	32-Mbyte AGP Nvidia GeForce2 MX	128-Mbyte AGP ATI Radeon 9600XT

eliminate this feature. With both the dual-core G4 and G5 Power Macs, CHUD additionally allows the user to disable the second microprocessor. This project's benchmarking exploited CHUD's configuration capabilities to more fully understand the variables that influence perceived system performance. Disabling the L2 cache also potentially enables the CPUs to more closely mimic the performance of reduced-cache and no-cache embedded PowerPC processors from AMCC, Freescale, and IBM, along with CPU cores from these companies and from

Xilinx's Virtex-II Pro FPGAs (see sidebar, "Embedding the Mac Mini").

The Mac Mini's published specifications include a 4200-rpm, 2.5-in. hard-disk drive and PC2700 (DDR333) CL2.5 (CAS, column address strobe, latency=2.5 clocks) SDRAM. Yet, when the system arrived, its System Profiler report revealed a 5400-rpm Seagate Momentus hard-disk drive, along with PC3200 (DDR400) CL3 (CAS latency=three clocks) SDRAM. Initial Mac Mini user feedback included complaints about slow system performance believed to be the fault of the low-speed hard-disk drive, and

it's possible that Apple has decided to spend the extra money for 5400-rpm drives to address those concerns. *EDN* ran benchmarks using both 4200- and 5400-rpm Momentus drives to quantify any system-performance differences between them, employing Bombich Software's Carbon Copy Cloner utility, along with an ADS Technologies' external hard-disk-drive enclosure, to make a mirror image of the in-system drive before doing the drive swap.

Other Mac Mini user-performance grumbles centered on the 256 Mbytes of system memory in the base configuration, so *EDN's* benchmarks tested system-memory capacities of 128 Mbytes to 1 Gbyte. Apple's public-relations contacts confirmed that the DDR400 memory would run at DDR333 speeds in the Mac Mini and that its presence versus DDR333 simply reflected relative market availability and cost at the time of its manufacture. Apple's second-generation Mac Minis, introduced at press time, bumped up the base-system memory to 512 Mbytes at no incremental cost, so there likely was some truth to users' complaints with first-generation units. Both the hard-disk drive and memory replacements done in conjunction with this project required opening the Mac Mini's



CRACKING OPEN THE CASE

Gaining access to the guts of the Mac Mini isn't too difficult, once you obtain the proper tool. It's a putty knife with a paper-thin blade, believe it or not, and the official Apple service manual even documents it. Insert the putty-knife-blade edge in the gap between the bottom assembly and each side of the metal housing, bend it backward to spring loose the internal latches, and don't fear the gruesome-sounding popping noises that result.

For a more detailed explanation (additional research is *highly recommended* before you proceed), search the Internet using keyword "mac_mini.pdf" for the Apple service manual. Other World Computing offers a QuickTime video clip of the procedure at <http://eshop.macsales.com/shop/mac-mini> at multiple resolutions. And PB FixIt offers a number of disassembly, parts-replacement, and reassembly instruction guides in PDF format at www.pbfixit.com/Guide/82.0.0.html.

enclosure, which wasn't a simple feat (sidebar "Cracking open the case").

SOFTWARE AND BENCHMARKS

All four systems showcased in this project ran OS 10.3.9 (code-named Panther). However, the latest Version 2.1 of Apple's free Xcode development tools, whose integrated GCC (GNU C compiler, <http://gcc.gnu.org>) Version 4 we used to compile the SPEC (Standard Performance Evaluation Corp) benchmarks, runs only under OS 10.4 (Tiger). Therefore, code compilation took place on a separate 15-in. PowerBook owned by Eric Nedervold, a veteran Mac OS- and Java-application developer, who also participated in the project.

Xbench (www.xbench.com), a well-known Mac-system-benchmarking utility, tests numerous computer subsystems and generates detailed reports of its findings. This project used Version 1.1.3 of Xbench, which was released in late 2003. Version 1.2, which was also released just as this article went to press, focused on OS 10.4 support as well as compatibility with Mac development systems based on Intel CPUs; neither issue affects this project's parameters. However, because Xbench is a Mac-only program, you cannot directly

compare its results with benchmarks run on x86-based systems.

Therefore, this project also encompasses SPEC CPU2000 Version 1.2 benchmarks, which are by design platform-independent. The SPEC Web site reveals an abundance of published SPEC CPU2000 results spanning multiple CPU architectures, including AMD and Intel x86, Intel Itanium, Hewlett-Packard PA-RISC, Sun Microsystems SPARC, and MIPS. But the only PowerPC results it lists are from IBM, and they come from workstations and servers. Mac-based results are notably absent from the list, and this project fills in some of the missing pieces. SPEC benchmarking on the Mac Mini employed a system configuration with 1 Gbyte of system memory and a 5400-rpm hard-disk drive.

As their name implies, SPECINT (integer) benchmarks test integer performance, and they're based on the C and C++ (for the 252.eon function) languages. (See Table 2 at www.edn.com/050915cs.) The SPECINT suite includes the following functions:

- 164.gzip (reference-time-1400) data-compression utility;
- 175.vpr (reference-time-1400) FPGA circuit-placement and routing;

- 176.gcc (reference-time-1100) C compiler;
- 181.mcf (reference-time-1800) minimum-cost network-flow solver;
- 186.crafty (reference-time-1000) chess program;
- 197.parser (reference-time-1800) natural-language processing;
- 252.eon (reference-time-1300) ray tracing;
- 253.perlbnk (reference-time-1800) Perl;
- 254.gap (reference-time-1100) computational-group theory;
- 255.vortex (reference-time-1900) object-oriented database;
- 256.bzip2 (reference-time-1500) data-compression utility; and
- 300.twolf (reference-time-3000) place-and-route simulator.

Within a single benchmark session, each function ran three times, and the SPEC software used the median score in its report. (Rather than average the three scores, it simply selects the middle one.) SPEC FP (floating point) contains 14 floating-point-intensive functions, written in a combination of Fortran-77 (six functions), Fortran-90 (four), and C (four) languages:

- 168.wupwise (reference-time-1600) quantum chromodynamics;

LINUX: A WORK IN PROGRESS

Taking our cue from IBM's Mac Mini documentation, we installed Terra Soft Solutions' YDL (Yellow Dog Linux) 4.01 on the system on a previously blank 60-Gbyte hard-disk drive. Unfortunately, we didn't get far in our evaluation. We knew that YDL 4.01 wouldn't support the Mac Mini's built-in sound chip or the Broadcom-based WiFi (Wireless Fidelity) transceiver in the integrated wireless module. (That drawback was one of the reasons we didn't spend \$100 extra on this option.) However, one of the primary enhancements of the YDL Version 4.01 over the

Version 4.0 predecessor was supposed to be full support for the Mac Mini's more-than-two-year-old ATI Radeon 9200 GPU (graphics-processing unit).

Postinstallation, the Mac Mini came up in a graphics mode in which roughly 20% of the screen, including the all-important Linux equivalent of the "start" button and program icons, shifted left off the visible desktop. (The Mac Mini was connected to a Compaq TFT5030 display.) Redefining the display from a generic monitor to a generic 1024×768-pixel LCD, we obtained a relatively stable 640×480-pixel GUI, but higher resolutions

were unavailable. The display would still occasionally come up in a stretched and left-shifted mode, but exiting and re-entering X-Windows or, in the worst case, rebooting Linux, would fix it. Attempts to explicitly identify the GPU as a Radeon 9200, therefore using a device-specific graphics driver instead of the default generic driver, resulted in a garbled, illegible output akin to a system's driving a progressive-scan display with an interlaced video signal.

In response to the display problem, Terra Soft's Chief Executive Officer Kai Staat comments, "The Mac Mini has a funky graphics card that is not easy to work

with." The system would also randomly boot up with the built-in Ethernet adapter disabled. We also tried to directly run the Live DVD version of Ubuntu Linux (www.ubuntulinux.org) Version 5.04 and had even less success; the system reset to an open-firmware prompt and froze when we selected the G4-specific build. When we selected the PowerPC generic Ubuntu variant, it complained about the GPU's frame buffer and refused to load X-Windows. Linux on the Mac Mini, we reluctantly conclude, remains not ready for prime time, except perhaps for the operating system's core constituency of patient power users.

- 171.swim (reference-time-3100) shallow-water modeling;
- 172.mgrid (reference-time-1800) multigrid solver in a 3-D potential field;
- 173.applu (reference-time-2100) parabolic and elliptic partial differential equations;
- 177.mesa (reference-time-1400) 3-D-graphics library;
- 178.galgel (reference-time-2900) fluid dynamics (analysis of oscillatory instability);
- 179.art (reference-time-2600) neural-network simulation (adaptive resonance theory);
- 183.equake (reference-time-1300) finite-element simulation; earthquake modeling;
- 187.facerec (reference-time-1900) computer vision (facial recognition);
- 188.ammp (reference-time-2200) computational chemistry;

- 189.lucas (reference-time-2000) number theory (primality testing);
- 191.fma3d (reference-time-2100) finite-element crash simulation;
- 200.sixtrack (reference-time-1100) particle-accelerator model; and
- 301.apsi (reference-time-2600) problem solving regarding temperature, wind, velocity, and the distribution of pollutants.

Unfortunately, Xcode does not include a Fortran compiler, so this article does not include SPEC FP results. Our first stab at compiling the SPECINT routines comprehended two GCC PowerPC-generic speed-tailored optimizations: “O0” (for no optimization) and “O3” (for full optimization). We ran into problems compiling the “eon” routine that a type mismatch in the header files caused. After more than a week of stop-and-go debugging, we created G4- and G5-optimized compilations, after surmounting even more compiler-versus-code incompatibilities.

Keep this information in mind as you peruse the following data and especially when comparing our results with those on the SPEC Web site. We’re not compiler experts, and it’s highly conceivable that by fine-tuning the compiler flags, a power user *might* be able to squeeze a few more percentage points’ worth of performance out of some or all of these chips. (Note that the SPEC license agreement expressly forbids altering *any* of the routines’ source code.) Keep in mind, too, that both the SPEC and the Xbench routines ran on systems with full OS X images loaded—not a stripped-down, text-only-mode Darwin configuration. Specifically, the dual G4 Power Mac, dual G5 Power Mac, and Mac Mini were all running Redstone Software’s OSXVNC server utility, which, according to OS X’s Activity Monitor, added a virtually undetectable incremental load to the system. We did terminate or otherwise disable all unnecessary background functions, however.



EMBEDDING THE MAC MINI

Does IBM’s vision of the Mac Mini as an embedded-development platform hold water? This simple question has a complex answer. Keep in mind, first, that the G4 PowerPC CPU in the Mac Mini has an excess of features compared with most embedded-PowerPC variants; examples include its out-of-order code-execution support, its abundant on-chip cache, and its AltiVec SIMD (single-instruction multiple-data) instruction set. Your code performance profile may differ greatly under the G4 than with the CPU in your final design, even if they run at comparable clock speeds.

Apple achieved the Mac Mini’s compact size at the expense of expansion capability. There’s no industry-standard PCI or equivalent bus connector into which you can plug add-in boards;

the developer note provides some information on the inner workings of the system, but Apple doesn’t detail the pinout and timings of the connector that mates up with the optional Bluetooth-and-WiFi (Wireless Fidelity) mezzanine board (Reference A). This dearth of internal expansion also means, for example, that you’re stuck with the ATI Radeon 9200 graphics chip, and that you can bump up the system’s main memory capacity to only whatever fits on a single DDR SDRAM DIMM.

With respect to external expansion, the Mac Mini supports only FireWire 400—not FireWire 800. You’ll also find only 10/100-Mbit Ethernet support; there’s no Gigabit Ethernet capability. And, turning your attention to software, the sidebar “Linux: a work in progress” details the difficulty we had

getting Linux to run on the Mac Mini, specifically with respect to the graphics subsystem. Linux-distribution providers will clean up such glitches over time, but they’re likely to occur with other operating systems, as well.

What about the next step: taking the Mac Mini directly to production as the hardware foundation for your system design? The allure of a fully debugged, high-volume production board, especially one that in its base configuration costs less than \$500, is compelling. But as a recently published iSuppli teardown report suggests, you’ll be supporting Apple’s profit aspirations; iSuppli estimated the bill-of-materials cost for the entry-level Mac Mini at \$274.69 (\$283.37 including manufacturing costs, Reference B).

Part of what you’re paying for—the Mac OS X operating

system and iLife application suite that come with the Mac Mini—you probably won’t even end up using. The Mac Mini system board isn’t in any sort of an industry-standard form factor (unless you count the fact that it snugly and, likely, coincidentally slips into the single-DIN slot of an automobile sound system). And you’ll be subject to the rapid obsolescence-and-replacement cycles of the PC business; don’t assume you’ll be able to buy the same Mac Mini configuration or, frankly, for that matter, *any* Mac Mini a few years down the road.

REFERENCES

- http://developer.apple.com/documentation/Hardware/Developer_Notes/Macintosh_CPUs-G4/MacMiniG4.
- www.isupply.com/marketwatch/default.asp?id=311.

Finally, note that the SPEC routines are relatively immune to brief interruptions from other contending system tasks, such as mouse movement, both because they run each function multiple times and use the median result, and because each iteration takes a long time to complete. In worst-case configurations, with caches and second CPUs disabled and running nonoptimized code, a single SPEC benchmark iteration ran for *several days*. In contrast, each Xbench cycle takes only a minute or so to complete and comprises numerous tests, increasing the possibility that an interruption might adversely affect one or several of them. **Table 3** at www.edn.com/050915cs offers potential evidence of this corruption. One way to alleviate the situation would be to run each test multiple times to filter out the divergent data.

THE RESULTS

Looking first at the SPEC data, you'll notice a consistent, significant performance improvement when L2 cache was enabled and a similar dramatic improvement when running

O3 code versus unoptimized O0 routines. The only PowerPC 74xx-based platform on which we ran the G4-optimized SPEC routines was the Mac Mini, and it resulted in an unexpected performance *decrease* compared with O3 code. The lack of a speed boost isn't surprising; a fundamental difference between G3 and G4 PowerPC CPUs lies in the G4's AltiVec, which Apple and IBM also refer to as Velocity Engine and VMX, respectively, SIMD (single-instruction multiple-data) instruction-set support. GCC will tap into this instruction set only if it finds explicit array data-type definitions in the C source code. But the root cause of the performance decrease with supposedly G4-optimized code is unclear.

Under SPEC, the Mac Mini and PowerBook numbers are nearly identical; the slight discrepancy is likely due to random run-to-run variance, and a repeat of the experiment might even produce the opposite results. Both systems' CPUs handily beat the PowerPC 7450 in the G4 Power Mac; the G4 Power Mac's L3 cache is insufficient to counterbalance its lower core and FSB speeds and its slower cache and main memory. Note, too, that enabling the second CPU in the G4 Power Mac offered little incremental benefit. Because we ran the SPEC benchmarks in their "speed" rather than "rate" mode, the various functions executed sequentially, not in parallel. Any SPEC-benchmarking-results benefit of enabling the second CPU is likely due to its ability to handle other system functions, leaving one CPU free to focus on executing SPEC code.

Similarly, the dual G5 Power Mac results improved only slightly when we enabled the second PowerPC 970FX CPU. However, the dual G5 Power Mac did much better than its G4 counterpart when executing CPU-optimized SPEC functions. Granted, there was little performance improvement with G5-tailored code, but at least there wasn't the dramatic performance decrease seen on the Mac Mini with G4-optimized routines.

Now, turn your attention to the Xbench data. One of the first things you might notice is its contrast with the SPEC case; enabling the second CPU resulted in a demonstrable improvement in many of the Xbench test results. Enabling the

NEW POWERPC FLAVORS

Several weeks after Apple Chief Executive Officer Steve Jobs announced the gradual conversion of his company's computer-product lines to Intel CPUs, Freescale and IBM introduced PowerPC processors that cast some doubt on the *real* motivations behind Apple's decision. Freescale's latest G4 CPUs, manufactured on 90-nm-process technology, include the single-core MPC7448, with maximum core clock speeds of 1.7 GHz and front-side-bus speeds of 200 MHz, and the code-compatible, dual-core MPC8641D. IBM's latest offerings are a low-power variant of the single-core 970FX (G5 PowerBook, anyone?) running at 1.2 to 1.6 GHz with corresponding power consumption of 13 to 16W, and the dual-core, 2.5-GHz-maximum 970MP.

+ Go to www.edn.com/050915cs to find tables with more data and insights, as well as this hands-on project's associated vendor box. At this URL, you can also post a comment on this article.

+ At the Brian's Brain blog, www.edn.com/briansbrain, visit the "Mac (under) the knife" entries to read the following postings:

- "Accessorize your Mac";
- "Small, silent, and spouse-sanctioned";
- "Future plans";
- "Troubleshooting";
- "Remote control";
- "Interesting lit and other bits," where you'll find the detailed report files generated by SPEC (Standard Performance Evaluation Corp) and Xbench, as well as the script files we used to compile the SPEC code; and
- "What do you observe?" where you can share your comments and peruse the observations that others have posted.

L2 cache also in most cases significantly improved the scores. Whereas the SPEC tests revealed little difference between the G4 CPUs in the Mac Mini and the PowerBook, the Xbench data magnifies the discrepancy, even if you focus only on the CPU-centric scores. Presumably, the CPU and core logic in the PowerBook are fine-tuned for power savings, whereas their counterparts in the Mac Mini are tailored for speed. And you'll notice that switching from a 4200-rpm drive to a 5400-rpm drive in the Mac Mini boosted its hard-disk drive-related test scores.

Not all of the Xbench data is predictable, however. Most baffling are the graphics-related results. Focusing on the Mac Mini, the Quartz Graphics Test numbers are unsurprisingly higher in all cases with the L2 cache enabled, but the results also increase on the 4200-rpm hard-disk drive-equipped system when system memory grows beyond 128 Mbytes. You might think, as we initially did, that this situation occurs because the graphics accelerator employs system memory as its frame buffer, but an ATI spokesperson confirms that the Radeon 9200 GPU (graphics-processing unit) has a dedicated 32-Mbyte video-memory array. Perhaps the poorer results with 128 Mbytes of system

memory are due to DRAM-influenced constraints elsewhere in the system. One outstanding question is why almost all of the Mac Mini Quartz Graphics numbers are *lower* when the system is equipped with a 5400-rpm hard-disk drive. The OpenGL Graphics tests with a 5400-rpm drive versus a 4200-rpm equivalent reveal a similar performance decrease, and the 5400 rpm numbers are generally irregular; in one case, the results were actually *better* with the L2 cache off!

Unfortunately, the memory tests didn't quantifiably expose the often-dramatic system-performance improvements that we perceived when we incremented system DRAM. With 128 Mbytes of memory inside, the Mac Mini was as slow as molasses, both to initially boot and to subsequently toggle between applications (especially when coupled with a 4200-rpm hard-disk drive). Switching to a 256-Mbyte DIMM noticeably improved both attributes, and a further increase to 512 Mbytes made yet another incremental step-up, albeit not as dramatic as its predecessor. The final increase, to a 1-Gbyte DIMM, resulted in no detectable improvement, although, in a heavily loaded system with many applications simultaneously running, the difference between 512 Mbytes and 1 Gbyte of DRAM might have been more noticeable. **EDN**

REFERENCES

- 1 www.edn.com/blog/400000040/post/60000606.html.
- 2 www.edn.com/blog/400000040/post/430000643.html.
- 3 www.edn.com/blog/400000040/post/690000869.html.
- 4 Krewell, Kevin, "Apple drops PowerPC for Pentium," *Microprocessor Report*, June 27, 2005.

ACKNOWLEDGMENT

I'm indebted to Eric Nedervold for his abundant expertise and numerous contributions to this project.

You can reach
Senior Technical Editor
Brian Dipert
at 1-916-760-0159,
bdipert@edn.com,
and www.bdipert.com.

