Cahners
MICROPROCESSOR REPORT
www.MPRonline.com
THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

SPEC CPU2000 Released

SPEC Integer and Floating-Point Benchmarks Updated

By Kevin Krewell {4/17/00-02}

Last December, the nonprofit Standard Performance Evaluation Corp. (SPEC) released a new version of its CPU benchmark—SPEC CPU2000. SPEC is composed of computer manufacturers, system integrators, microprocessor vendors, universities, research organizations, publishers, and consultants interested in producing a method to measure performance using standardized, cross-platform, compute-intensive workloads.

The goal set by the SPEC committee for the SPEC benchmarks is to clearly separate the processor-performance aspects and measure them. System issues—such as graphics, mass storage and I/O performance—are excluded or minimized as much as possible. The SPEC component processor benchmark (CPU2000) has two tests: integer (CINT) and floating point (CFP), focused on the two different types of numerical representations. The letter "C" in both tests stands for "component," indicating the test is focused on one component's aspect of the processor or system.

Both these tests are influenced by some common processor elements: memory interface, compiler technology, branch prediction hardware, and so on. These common characteristics have significant influence on the integer and floating-point benchmarks, and the SPEC CPU numbers, therefore, measure more than just the raw throughput of the integer and floating-point units. This is as it should be; a benchmark that fits in a level-one cache, doesn't branch, and tests only the performance of the processor core will not be a good test of delivered performance.

One of the least understood aspects of the SPEC benchmarks is the effect of compiler technology on the results. For a benchmark to be run across multiple platforms, it must be compiled for the environment and processor being tested. The compiler has a direct effect on the efficiency of the target code and therefore on the results of the test. A baseline number is always required with minimal compiler optimization switches, and a second number that

SPEC CINT2000	Language	Description
164.gzip	C	Data compression
175.vpr	C	FPGA circuit place and route
176.gcc	C	C compiler
181.mcf	C	Minimum cost-flow network
186.crafty	C	Chess program
197.parser	C	Natural language processing
252.eon	C++	Ray tracing
253.perlbmk	C	Perl
254.gap	C	Computational group theory
255.vortex	C	Object-oriented database
256.bzip2	C	Data compression utility
300.twolf	C	Place and route simulator
SPEC CINT95	**Language**	**Description**
099.go	C	AI, plays the game "go"
124.m88ksim	C	88K chip sim, runs test program
126.gcc	C	C compiler, SPARC code
129.compress	C	Compress and decompress
130.li	C	LISP interpreter
132.ijpeg	C	Graphics compress/decompress
134.perl	C	Perl string manipulation
147.vortex	C	Database program

Table 1. The number of programs in the SPEC integer suite rose from 8 to 12 to increase the variety and breadth of coverage. Since recent processors are much faster, more testing can be performed in the 48 hours allocated on the reference machine. All SPEC CPU2000 integer programs are written in either C or C++.

allows aggressive compiler optimization switches to be turned on is optional. New SPEC performance numbers can be issued when a new compiler is released.

Unfortunately, there is a way around the base number constraints: using a compiler that defaults to aggressive optimizations. For example, the Dell/Intel SPEC CPU2000 performance numbers listed on the SPEC Web site (*www.spec.org*) used an Intel C compiler (4.5), which produced minimal differences between base numbers and those taken with aggressive compiler options. The Intel compiler may or may not be typical of compilers used by the software industry and could have been tuned to perform exceptionally well on these particular sections of code. This may explain why Intel chose to pass up the popular Microsoft C and C++ compilers in favor of its own. It may also explain why AMD has yet to submit CPU2000 numbers for the Athlon processor—it may be waiting for a compiler that can give Athlon more competitive numbers.

The use of specialty compilers in the PC market may not be quite fair, as most applications will use readily available compilers from Microsoft or other vendors or the open-source GNU C compiler. In the scientific/technical space, a highly optimizing compiler is part of the true performance of an processor, because these applications stress a system and often use a maximum-performance compiler.

The SPEC CPU benchmarks were designed for workstation and server platforms, not embedded or general PC applications, and it is also not designed to run on a Macintosh. The workstation bias is obvious in light of SPEC's minimum system requirements of 1G of hard-drive space and 256M of memory, and in the fact that it runs only on Unix (POSIX-compatible versions, including Linux) or WinNT.

Comparing SPEC CPU2000 with SPEC CPU95

SPEC CPU2000 is a significant rework of the SPEC 95 benchmark. SPEC 95 was showing its age, partly because the base reference machine used for comparisons in SPEC 95 was a 40MHz Sun SPARCstation 10/40 with no L2. Modern processors are nearly 100 times faster on certain tests compared with the SPEC 95 reference machine (the swim test ran 96.6 times faster than the reference time on a 733MHz Pentium III system). Clearly, the reference system needed to be updated, and SPEC chose the Sun Ultra 5 with a 300MHz UltraSPARC IIi and a 256KB L2 cache. Because SPEC CPU2000 is based on different programs than CPU95 and on a different reference system, there is no direct conversion factor, and SPEC has requested retesting of older systems with the new benchmark to allow some historical comparisons with present-generation systems. The CPU2000 numbers are scaled up by a factor of 100 to avoid the extensive use of fractional numbers to compare similar machines, and the higher numbers will make CPU2000 appear more like a continuum from CPU95.

The tasks performed by the SPEC CINT95, listed in Table 1, include compressing data files and image files, compiling code, emulating processors, playing a complex game, running a database, and executing Perl scripts. The CINT2000 release requires significantly more work from the processor, with additional tests such as ray tracing (used in high-end graphics), natural language processing, chip design (place and route), and some additional math-intensive programs. While data compression is represented in SPEC CINT2000, the CINT95 JPEG test was dropped, leaving no representative of the discrete-cosine-transform (dct) algorithm used in many multimedia applications.

One of the new programs, "crafty," is based on a 64-bit variable, which could give 64-bit machines and C compilers with "long long" extensions an advantage.

The floating-point programs used in SPEC CFP95, shown in Table 2, are almost exclusively double precision, with the exception of "swim." For SPEC CFP2000, all programs are now double precision. For high accuracy, that is a good choice, but many existing programs, such as the OpenGL graphics API are currently optimized for single-precision data, which on many computers executes considerably faster than double-precision. For example, Pentium III's streaming SIMD extensions (SSE) instructions are helpful on single-precision benchmarks, but for double-precision benchmarks SSE is of no value. The upcoming Willamette processor, however, can process two double-precision values

SPEC CFP2000	Language	Description
168.wupside	Fortran 77	Physics/quantum chromodynamics
171.swim	Fortran 77	Shallow water model
172.mgrid	Fortran 77	Multigrid solver in 3D potential field
173.applu	Fortran 77	Parabolic/elliptic partial differential equations
177.mesa	C	3D graphics library
178.galgel	Fortran 90	Computational fluid dynamics
179.art	C	Image recognition/neural networks
183.equake	C	Seismic wave propagation simulation
187.facerec	Fortran 90	Image processing: face recognition
188.ammp	C	Computational chemistry
189.lucas	Fortran 90	Number theory/primality testing
191.fma3d	Fortran 90	Finite-element crash simulation
200.sixtrack	Fortran 77	High energy nuclear physics design
301.aspi	Fortran 77	Meteorology: pollutant distribution
SPEC CFP95	**Language**	**Description**
101.tomcatv	Fortran	A mesh-generation program
102.swim	Fortran	Shallow water model
103.su2cor	Fortran	Quantum physics, Monte Carlo simulation
104.hydro2d	Fortran	Astrophysics hydrodynamical Navier-Stokes Eq
107.mgrid	Fortran	Multigrid solver in 3D potential field
110.applu	Fortran	Parabolic/elliptic partial differential equations
125.turb3d	Fortran	Sim Isotropic, homogeneous turbulence in a cube
141.apsi	Fortran	Prob of temp, wind, velocity & dist of pollutants
145.fpppp	Fortran	Quantum chemistry
146.wave5	Fortran	Plasma physic, electromagnetic particle simulation

Table 2. The SPEC CFP2000 benchmark increased the number of programs from 8 to 14 and began using C-language programs in addition to Fortran.

with SSE2, and, with the appropriate compiler support, it could produce impressive CFP2000 numbers.

The other major change is that in SPEC 95, all the programs were in Fortran, a language previously popular in engineering and scientific fields but little used in modern commercial software. In SPEC CFP2000, 4 of the 14 programs are C programs—a step in the right direction.

The floating-point programs in SPEC CFP2000 solve a wide array of problems, from quantum physics to 3D graphics. But the tests lean heavily to scientific and challenging engineering problems. The array of 14 programs includes 3 that might apply to an advanced desktop PC environment: 3D graphics (mesa), image recognition/neural nets (art), and image processing: face recognition (facerec). Other programs would apply to a narrow field of interest, such as quantum chromodynamics to scientists at Brookhaven National Labs investigating quarks, gluons and the big bang theory. Missing from CFP2000 are more typical compute-intensive PC applications of floating-point processing: head-transfer functions used in positional 3D sound, artificial intelligence for games, general-purpose physics simulations, and so on.

SPEC CPU95 will have a six-month phase-out. After April 1, 2000, any SPEC CPU95 submissions must include a SPEC CPU2000 submission. By July 1, SPEC plans to stop accepting CPU95 submissions, and SPEC will, at some point, stop selling it. Running SPEC 2000 on older machines is encouraged by SPEC to add reference points to be compared with those from newer processors.

There is no direct conversion between CPU2000 and CPU95 numbers, but it is expected that processors that did well on CPU95 will also do well on CPU2000. Two strong CPU2000 performers—Alpha (667MHz with 4G of memory) and Intel Pentium III (733MHz with 840M of memory) —also had strong CPU95 numbers. A top Alpha processor, the AlphaServer DS20E Model 6/667, has a CINT2000 (base) score of 424 on the SPEC Web site and a CINT95 score of 35.7, as Table 3 shows. The posted details of the Alpha results include the hardware and software configuration information and the compiler switch settings for each test. In this comparison, the revision of the compiler used for each test was different, and the base compiler switches were also slightly different.

Each benchmark is run under two compiler conditions: base (with a few compiler optimization flags on) and aggressive (maximum optimization), and both results are reported. Thus SPEC benchmarks favor companies with the resources to build highly optimizing compilers. In the case of the Alpha system, the CINT95 base number was 35.7, but the aggressive number was 40.1—a substantial 12.3% improvement. Because the benchmark is updated so infrequently (every three to five years), it is also possible to tune microprocessor designs to run the benchmarks well.

While the SPEC benchmarks may favor companies with the resources to tune their compilers and silicon, that situation is nothing more than the harsh realities of the industry and is not necessarily bad. To the extent SPEC benchmarks reflect reality, they present a reasonable target for all processor vendors.

Confusion arises, however, when companies take benchmarking to the extreme by, for example, creating benchmark compilers that artificially inflate results over what users of their processors can expect to see in practice. SPEC benchmark results are definitely more meaningful when taken with a production compiler using optimizations that are similar to those that would be applied when creating production-quality application code. The base SPEC score gives the closest approximation to these conditions, but, unfortunately, the SPEC rules do not outlaw specialty benchmark compilers or profile-directed feedback even for baseline scores. While no holds should be barred for peak scores, a somewhat more level field for baseline results might be an improvement.

In addition to the CINT and CFP benchmarks, SPEC also includes a rate version of both, which is designed to show the performance of multiprocessor systems running similar tasks in parallel. The rate measurement is not generally applied to single-processor performance.

SPEC CPU2000 Program Selection

The software used in the SPEC CPU benchmarks must be available as open source code, which does not allow the measurement of proprietary algorithms and “black-box”

SPEC CINT2000	Ref.Time	Run Time	Base Ratio
164.gzip	1400	436	321
175.vpr	1400	380	368
176.gcc	110	222	496
181.mcf	1800	417	432
186.crafty	1000	200	500
197.parser	1800	616	292
252.eon	1300	252	515
253.perlbmk	1800	421	427
254.gap	1100	345	318
255.vortex	1900	344	552
256.bzip2	1500	329	457
300.twolf	3000	582	515
SPECint_base2000			424
SPEC CINT95	**Ref. Time**	**Run Time**	**Base Ratio**
099.go	4600	135	34
124.m88ksim	1900	43.5	43.6
126.gcc	1700	58.1	29.3
129.compress	1800	58.7	30.7
130.li	1900	54.3	35
132.ijpeg	2400	57.9	41.4
134.perl	1900	56	33.9
147.vortex	2700	66.6	40.5
SPECint_base95			35.7

Table 3. A comparison of a 667MHz Alpha processor on both CINT2000 and CINT95. The base reference time is the number of seconds required to run the test on the base reference machine. Base run time is the time required for the target machine with the base compiler settings. The reference time is divided by the base run time, then multiplied by 100 to produce the base ratio. The most frequently quoted score is the combined number, which is the geometric mean of the individual scores.

Beware of Benchmarks

Benjamin Disraeli's famous quote about statistics can also be paraphrased to address benchmarking—that there are three kinds of lies: lies, damned lies, and benchmarks. Discussions on benchmarking and the measurement of performance are like discussions on religion and on the nature of good and evil—there are strong opinions from many different biases. But if all benchmarks were flawed or wrongheaded, there would be no way to judge the relative merits of various processor architectures and implementations. Therefore we need to understand the relative strengths and weaknesses of each benchmark and how it applies to a particular market.

One of the worst benchmarks is the one most referenced in consumer marketing—the clock frequency of the processor (megahertz marketing). Those familiar with PC systems know that frequency alone is not an accurate measure of performance. Cross-platform benchmarks can be difficult to develop and maintain, but they do allow comparisons of different processors in their native environments. One problem, however, is that the word "performance" is a subjective term. On what set of applications should performance be measured? How many specific metrics are needed to accurately measure performance? What system components should affect the performance measure? These open-ended questions must be answered for each benchmark.

Maintaining a healthy skepticism about benchmarks and benchmarking is wise. But, lacking a better method, we are forced to rely on benchmarks to compare processors on the basis of performance. Because of their focus on CPU performance and because of their independent nature, MDR has in the past relied heavily—although not exclusively—on the SPEC benchmarks to help evaluate processors. And, since the new SPEC CPU2000 benchmarks appear to be a significant improvement over the earlier SPEC 95 benchmarks, we will continue to rely on them in the future. To minimize the distorting effects of highly specialized compiler optimizations, we prefer to judge processors according to their "base" scores rather than their "peak" scores. When practical, we will report the results from individual tests in addition to the composite scores.

code. And the SPEC committee relies on donated code and therefore does not build custom code for benchmarking purposes. SPEC will offer bounties for code and then sift through the submissions to find an agreed-upon mix. If a key algorithm is missing, SPEC does not generate the new code to address this shortcoming.

Reviewing the programs included in SPEC, we found that a common multimedia algorithm was not included—the discrete cosine transform (dct)—which is the basis for MPEG-2 video. The assumption made by the SPEC committee is that this type of performance-sensitive code would be written in assembly language, not in a high-level language. SPEC did include two compression algorithms, which could be interpreted as meaning that compression algorithms are twice as important as other functions, but it missed advanced compression technologies such as fractals and wavelets. Another missing algorithm is speech recognition, even though it is a compute-intensive application for the client. To test speech-recognition performance, a test is available in the BABCO SYSmark2000 benchmark that uses the Dragon speech-recognition engine. Unfortunately, SYSmark2000 is not a cross-platform benchmark, and it runs only in the x86/Windows environment.

Weighting the Importance of Each Component

A geometric mean of the component program scores is used to form the SPEC CINT and CFP composite results, but there is no weighting to frequency of use. While a geometric mean is mathematically superior to an arithmetic mean (average), a straight democratic approach to combining the individual components of the benchmark doesn't account for the relative importance of each. Unfortunately, in the benchmark-by-committee approach of SPEC, it is unrealistic to come to a consensus on such matters—a limitation of the SPEC process.

The approach of applying a weight to each test based on projected frequency of use is exemplified by the Ziff-Davis Winstone benchmark for PCs. The ZD benchmark uses a collection of popular programs run by a script that simulates a user. The performance of the program is then weighted by the relative popularity of the program (in sales), the theory being that the more-popular-selling software is used more often and therefore should have a larger weight in the results. Ziff-Davis's benchmark operations have produced benchmarks that have evolved over the years to a widely adopted standard, and its Winstone benchmark is the basis of the performance rating (PR) used by VIA/Cyrix. These benchmarks, however, are more influenced by system parameters than are the SPEC benchmarks. As a result, they are less revealing than SPEC for evaluating processor performance.

Caveat Emptor

SPEC is a collection of unrelated programs that attempt to represent advanced computational tasks. It is not designed to represent typical applications, and it is biased toward heavy-duty computational tasks. The floating-point applications are all double-precision (64 bit) and very much oriented to scientific applications, where computational performance and accuracy are of primary importance. Server applications are better served by other benchmarks that factor in I/O performance. PC and client system performance models are totally different and are not well represented here. What SPEC CPU benchmarks do is push the bounds of the processor beyond the ordinary, and determine how extraordinary the processor is.

All benchmark results can be distorted in one way or another. How do you protect yourself from misleading or inflated claims? First, insist on full disclosure. All system components should be revealed, including the compilers used and the compiler switch settings. With the knowledge of the components and settings, you can compare the test setups with the products that are actually available for sale. For SPEC benchmarks, verify that the compiler is available outside of the vendor's lab. Encourage vendors to use realistic components in their testing.

Unfortunately, in the hypercompetitive PC space, it is extremely difficult to get AMD and Intel to give apples-to-apples benchmark comparisons—rather, they both seem to strive to avoid a direct comparison. We believe that a reasonable PC processor comparison would be to use the latest Microsoft C, C++, and Fortran compilers (or mutually agreed substitutes), running Windows NT, on a single-processor, commercially available platform with 256M of main memory. If AMD is serious about entering the commercial PC, workstation, and server markets, it will need to address the notable lack of SPEC CPU2000 benchmarks for Athlon.

Standardizing on SPEC Benchmarks

An update to the SPEC CPU benchmark was overdue, since the last update was in 1995, five years ago. The previous update was in 1992, only a three-year period. The process of developing the SPEC benchmarks is similar to the process required for industry standards. It involves getting a number of competitors with differing vested interests to agree on a common set of principles—obviously a slow and torturous process. The president of SPEC and the members of the committee should be congratulated for producing a reasoned and reasonable benchmark.

When comparing entry-level PC/thin-client/appliance-oriented processors, the SPEC CPU2000 benchmarks are not the best measure of end-user experience. For those applications, Ziff-Davis and BAPCO styles of application-based benchmarks are more appropriate. Even those, however, are not without fault. Transmeta, for example, makes a solid argument that these script-based benchmarks don't accurately reflect the real user experience on a code-morphing processor such as Crusoe (see *MPR 2/14/00-01*, "Transmeta Breaks x86 Low-Power Barrier"). More processors and systems in the future could have other unique characteristics that cause these benchmarks to distort their performance picture. SPEC benchmarks are certainly not immune to such distortions, but being lower level and less influenced by system features, are probably somewhat less susceptible.

Like most other benchmarks, SPEC benchmarks do not take battery life into account. Portable systems are a special case because pure processing power is not the only important metric; heat dissipation and battery life also play an essential role in evaluating processors for portable systems.

Although they are far from perfect, in our judgment the SPEC benchmark suite offers the best available method of measuring the performance of nonembedded, nonmobile microprocessors. In the past, *Microprocessor Report* has relied on the SPEC CPU95 base numbers supplied by vendors or posted on the SPEC Web site to evaluate and contrast the overall performance of microprocessors. In the future we will gradually be converting to the use of SPEC CPU2000 as more numbers become available, but we will continue to use the base number, as we believe it provides the most meaningful basis for comparing various processors. For more specific performance evaluation, however, readers should look directly to the individual SPEC tests for results that are the most relevant to their application. ◇

Price & Availability

The SPEC 2000 benchmark is available from SPEC for $500. Discounts are offered for research organizations and universities. SPEC can be reached at: 540.349.7878 or *info@spec.org*.

The SPEC Web site is *www.spec.org*.