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

EEMBC Releases First Benchmarks

Five Application Suites Put a Dozen Embedded CPUs to the Test

By Tom R. Halfhill {5/1/00-02}

Years from now, retired engineers will bet drinks not on Joe DiMaggio's batting average in 1941, but on how fast a 167MHz NEC VR5432 processor could execute a finite impulse response filter in 2000. Well, maybe they won't, but the answer is interesting anyway—especially when compared with the same data for competing processors. Until now, there were no objective, independently verified data for comparing the performance of embedded processors on real-world tasks. That changed on April 11 when the EDN Embedded Microprocessor Benchmark Consortium (EEMBC, pronounced "embassy") released its long-awaited first benchmark results.

MIPS-compatible processors dominated this round of benchmarking, with three MIPS licensees (IDT, NEC, and Toshiba) bravely subjecting five different chips to EEMBC's rigorous tests. The x86 architecture was represented by two processors—AMD's K6-2 and National Semiconductor's Geode GX1. Other early birds were Infineon (TriCore TC10GP), Mitsubishi (M16C/62A), and STMicroelectronics (ST20C2). NEC also benchmarked its V832 (a 32-bit CPU based on a proprietary architecture), and Toshiba tested its TMP95FY64F (a proprietary 16-bit microcontroller). NEC and ST were the only vendors to test their chips with all five of EEMBC's application suites: automotive/industrial, consumer, networking, office automation, and telecommunications.

Conspicuously missing from this first round of benchmarking are embedded processors based on the popular ARM, Hitachi SuperH, and Motorola 68K architectures. It's possible that some vendors ran the benchmarks and decided not to complete the certification process or publicly release the results.

The initial test results are based on the 1.0 version of EEMBC's benchmarks. The five application suites have varying numbers of tests—for instance, the office-automation suite consists of four tests, while the automotive/industrial and

EEMBC Test	
Auto/Industrial Suite	
Angle-to-time conversion	Inv discrete cosine transform
Basic floating point	Inverse FFT filter
Bit manipulation	Matrix arithmetic
Cache buster	Pointer chasing
CAN remote data request	Pulse-width modulation
Fast-Fourier transform (FFT)	Road speed calculation
Finite impulse resp (FIR) filter	Table lookup and interpolation
Infinite impulse resp (IIR) filter	Tooth-to-spark calculation
Consumer Suite	
Compress JPEG	RGB-to-CMYK conversion
Decompress JPEG	RGB-to-YIQ conversion
High-pass grayscale filter	
Networking Suite	
OSPF/Dijkstra routing	Packet flow (1MB)
Lookup/Patricia algorithm	Packet flow (2MB)
Packet flow (512B)	
Office Automation Suite	
Bezier-curve calculation	Image rotation
Dithering	Text processing
Telecommunications Suite	
Autocorrelation (3 tests)	Fixed-pt complex FFT (3 tests)
Convolutional encoder (3 tests)	Viterbi GSM decoder (4 tests)
Fixed-point bit alloc (3 tests)	

Table 1. The EEMBC 1.0 benchmarks currently consist of 46 tests divided into five application suites. Vendors can choose which suites to use when benchmarking their processors.

telecommunications suites each have 16 tests. The suites may evolve over time, and EEMBC may add more suites in the future to keep up with new applications for embedded processors. Technical subcommittees staffed with representatives from member companies are responsible for creating the tests (see *MPR 6/21/99-01*, "Embedded Benchmarks Grow Up").

Unlike simple Dhrystone loops, the EEMBC tests are based on algorithms used in real embedded applications. In some suites, the algorithms span a wide range of tasks. As Table 1 shows, the automotive/industrial suite includes algorithms that run the gamut from motor control to in-car entertainment.

As expected, the EEMBC results are an avalanche of raw data. For each test in each suite, EEMBC reports the number of times per second each processor executed the algorithm, as well as the size of the compiled code and the sample data. (To avoid granularity problems with very small results—some chips achieve less than one iteration per second on some algorithms—the tests actually run for a longer period, and the result is derived mathematically.) For the automotive/industrial suite, EEMBC is reporting scores for six processors, which yields a table of 96 raw performance numbers—and these are merely the initial results. All together, EEMBC has released data for more than 270 test runs on 12 different chips. Unlike the Standard Performance Evaluation Corporation (SPEC), which mainly benchmarks desktop/server processors (see *MPR 4/17/00-02*, "Update: SPEC CPU2000 Released"), EEMBC doesn't attempt to distill the raw data into a composite score that yields an easy-to-digest single figure of merit.

There are some good arguments for not summarizing EEMBC's results in a single score. In a desktop or server processor, maximum performance is usually the most desirable quality, so a high aggregate SPEC number is a valuable benchmark. But in many embedded applications, adequate performance on a given task is good enough—higher performance might not make a difference and might only increase costs and power consumption. A processor's performance on a single algorithm might be all the information an embedded developer needs to conclude which chip is best for the job. For these reasons, we encourage readers to study EEMBC's raw numbers and, if it makes sense, derive their own aggregate results.

Deriving an aggregate score has some value if it reveals relationships between processors that are difficult to detect in the raw numbers. For now, it's up to enterprising analysts, journalists, and marketing mavens to interpret these data in their own ways. We performed such an exercise with EEMBC's version 0.9 benchmark results in the MPR article cited above, and we'll repeat that process with the first official results reported here. (Incidentally, the aggregate scores we derived from the v0.9 benchmarks bear no relationship to the latest scores.) Those unhappy with the way we've cooked the numbers are free to try their own hand. The raw data is available on EEMBC's Web site at *www.eembc.org* and will be useful for engineers who need algorithm-specific information about a processor.

MDR's Unofficial EEMBCmarks

Our goal is to reach some conclusions based on a first order of approximation—not necessarily the final word, but a reasonable assessment. We started by normalizing all the test results to the slowest processor that participated in all the benchmark suites: NEC's V832. Normalization fixes the V832's composite score at 1.0 and distributes all other scores relative to that baseline. For example, a processor that scores 2.0 in a given benchmark suite would be twice as fast as the V832; a processor that scores 0.5 would be half as fast.

We settled on the V832 as the reference chip only because the other two processors that participated in all the benchmark suites (NEC's VR5432 and VR5000) are faster, so using one of them as the baseline would have resulted in more-fractional composite scores, which are more difficult to interpret at a glance. We would prefer to use the slowest overall processor as the

Benchmark Test	Mitsubishi M16C/62A	NEC VR5000	NEC VR5432	NEC V832	STMicro ST20C2	Infineon TriCore
Angle-to-Time Conversion	1,376.0	341,699.0	230,727.0	35,970.0	8,442.9	85,094.0
Basic Floating Point	160.0	326,715.0	142,083.0	6,181.0	564.3	13,795.0
Bit Manipulation	60.0	4,120.0	4,162.0	980.0	288.0	1,508.0
Cache Buster	19,878.0	1,169,159.0	941,010.0	266,274.0	55,555.1	293,478.0
CAN Remote Data Request	26,859.0	1,379,989.0	1,043,100.0	407,280.0	71,547.2	424,540.0
Fast-Fourier Transform	5.3	351.0	301.0	55.0	7.6	116.0
Finite Impulse Response Filter	849.0	63,847.0	61,528.0	26,052.0	2,315.0	17,371.0
Infinite Impulse Response Filter	790.0	34,162.0	19,837.0	12,027.0	2,674.0	16,390.0
Inverse Discrete Cosine Transform	26.0	6,168.0	4,084.0	1,192.0	245.4	1,064.0
Inverse Fast-Fourier Transform	5.2	372.0	323.0	61.0	8.0	128.0
Matrix Arithmetic	0.8	1,469.0	588.0	28.0	2.9	54.0
Pointer Chasing	141.0	8,722.0	6,682.0	2,030.0	407.8	2,112.0
Pulse-Width Modulation	15,625.0	1,033,578.0	958,662.0	255,094.0	26,383.9	237,603.0
Road Speed Calculation	18,055.0	941,313.0	735,269.0	325,342.0	55,861.0	308,303.0
Table Lookup and Interpolation	230.0	110,064.0	93,633.0	7,699.0	1,786.3	25,773.0
Tooth to Spark	257.0	36,475.0	28,981.0	11,942.0	2,697.3	15,954.0

Table 2. These are the raw numbers that EEMBC reports for the automotive/industrial benchmark suite. They represent the number of iterations per second each processor performed an algorithm in each test.

baseline, but the chips that are slower than the V832 either didn't participate in all of the suites or were unable to run all of the tests. Ideally, we think EEMBC should define a reference processor as a common baseline, much as the Dhrystone benchmark is based on the historic VAX 11/780. In our previous article about EEMBC, we nominated Motorola's 68000 as a candidate because it's a popular and seminal embedded-CPU architecture. Unfortunately, no EEMBC results for the 68000 are currently available.

Table 2 shows the raw benchmark data for the automotive/industrial suite. This is the data available on EEMBC's Web site. To normalize these scores, we divided each chip's result in each test by the number of iterations performed by the V832—such as 35,970 for the angle-to-time conversion test or 6,181 for the basic floating-point test. (The normalized numbers don't appear in this table.)

Next, we computed the geometric means of the normalized scores. An arithmetic mean of the raw data would not be statistically valid for this purpose because of the extremely wide variations of the results, and it is not the proper way to combine normalized results. Raw results that yield a very small number of iterations (such as the matrix-arithmetic test shown in Table 2) would have virtually no effect on an arithmetic mean when combined with raw results that yield a very high number of iterations (such as the control-area network remote data request). In effect, an arithmetic mean of raw results would impose an arbitrary weighting system that heavily favors the tests with the most iterations per second.

In contrast, normalization eliminates the bias toward tests with high iteration counts, and the geometric mean is the preferred way to combine normalized results. Some other benchmark suites and organizations that use geometric means for this purpose include SPEC's CPU2000 (desktop/server CPUs), Viewperf (3D graphics), and SPECjvm98 (Java); the National Software Testing Laboratory (NSTL) Windows application-level benchmarks; AT&T Labs' Bench++ (a compiler benchmark); and Byte.com's CPU-specific BYTEmarks. Although it's possible and even desirable to compute a geometric mean in which the tests are weighted in order of importance—Viewperf and others do this—we chose not to do so, because the weighting is highly application dependent. Again, that's a job for EEMBC.

To calculate a geometric mean, multiply all the results of the tests together and take the *n*th root of the product, where *n* equals the number of tests. (Microsoft Excel has a built-in geometric-mean function—GEOMEAN—that makes this easy.) The result is MDR's unofficial "EEMBCmark." Figure 1 shows the normalized geometric-mean scores for the processors in the automotive/industrial suite. Remember, because we normalized the results first, the V832 will always have a geometric mean of 1.0.

Rather than viewing the results of this mathematical exercise as a horse race, consider the differences among these chips before jumping to any quick conclusions. The 16MHz Mitsubishi microcontroller isn't necessarily the loser just because it has the lowest score; it's certainly not in the same class as the 250MHz NEC VR5000 processor that scored highest in this suite. Indeed, it may be the better choice for a particular application if its performance is adequate, because it costs less, consumes less power, dissipates less heat, and occupies less board space than the VR5000.

On the other hand, Figure 1 makes some conclusions about relative performance quite obvious. NEC's VR5432 is only 17% faster than the V832 in terms of clock frequency, but it runs these EEMBC tests a whopping 4.5x faster. Because the difference can't be wholly explained by the disparity in clock speed, the results suggest that the VR5432 has a significantly superior architecture. And in fact, that's the case. The V832 is a 32-bit uniscalar member of NEC's little-known V830 family, and it has only 8K of primary cache (see *MPR 8/24/98-en*, "NEC Releases 144MHz V832"). In contrast, the VR5432 is a 64-bit dual-issue superscalar MIPS processor with 64K of cache (see *MPR 3/9/98-01*, "NEC VR5400 Makes Media Debut"). The VR5432's six execution units include two double-precision FPUs, while the V832 has no hardware support for floating-point math at all. That certainly explains the vast difference between their raw scores in the basic floating-point test, as shown in Table 2: the VR5432 is 23x faster than the V832.

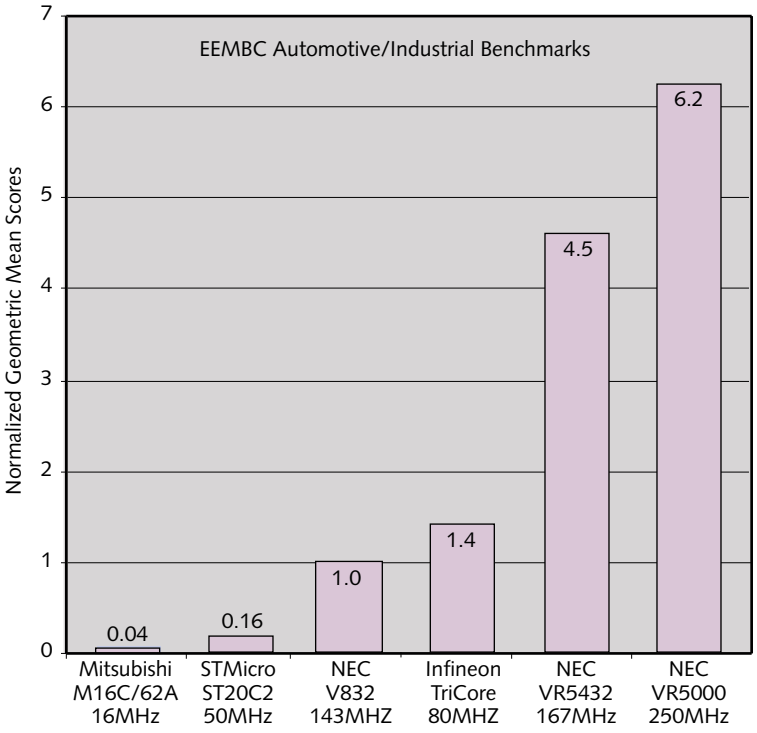

Figure 1. MDR derived these unofficial "EEMBCmark" scores by computing the normalized geometric means of EEMBC's raw benchmark results in the automotive/industrial suite. NEC's V832 is our reference baseline, normalized to 1.0.

Not that the V832 doesn't have some advantages over the VR5432. It costs about 50% less ($16 versus $30) and typically consumes only 15% as much power (380mW versus 2.5W). Its mixed 16/32-bit instruction set is denser than the VR5432's fixed-length 32-bit MIPS instructions, so it conserves memory too. The V832's compiled code for this benchmark suite is only 70% as large as the VR5432's compiled code (59K versus 84K). For many embedded developers, those numbers are as important as raw performance benchmarks. That's why EEMBC results should be approached differently than pure performance-minded benchmarks such as SPEC's CPU2000 scores for desktop/server processors.

Another thing to keep in mind when comparing processors is that MDR's unofficial EEMBCmark scores are specific to each benchmark suite. That is, a chip's EEMBCmark score in the automotive/industrial suite cannot be directly compared with similarly derived EEMBCmark scores in a different suite. The tests in each suite aren't the same. It would be like comparing a desktop processor's SPECint (integer) score to another processor's SPECfp (floating-point) score—the numbers are based on completely different tests, so they aren't comparable with each other. To make this clear, we'll follow SPEC's example and append an abbreviation for each suite to the benchmark designation: NEC's VR5432 has an EEMBCmark.auto/indy score of 4.5, which cannot be compared to the VR5000's EEMBCmark.consumer score of 2.73.

Although it's possible to compute an overall EEMBCmark score by combining the test results from all the suites, this would greatly reduce the number of comparable chips, because most vendors don't submit their chips to all the tests. In this first batch of results, the only chips that participated in all the suites were NEC's V832, VR5432, and VR5000, and ST's ST20C2. The ST20C2 was unable to run all the tests in the networking suite, so any attempt to calculate an overall EEMBCmark score from these results would be limited to NEC's chips.

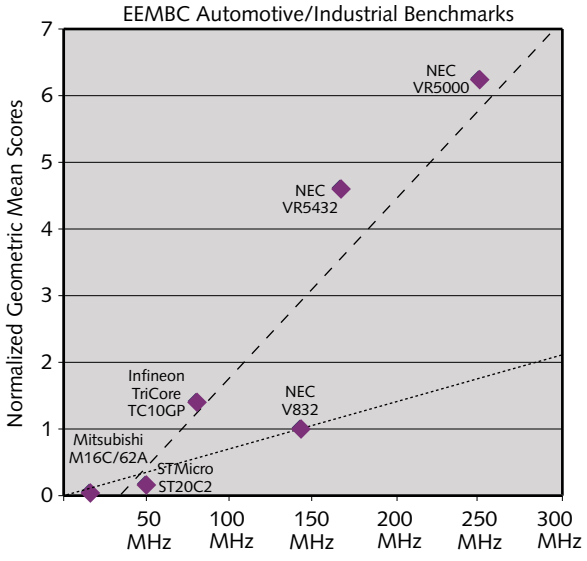

Figure 2. This chart shows how each processor's performance relates to clock frequency. The coarse dotted line with the steeper slope is the average performance trendline, based on all the chips. The finely dotted line at the bottom with the lower slope represents a linear increase in performance with clock speed, based on the reference V832 chip.

Performance Trendlines Show More

To carry our analysis a step further, we wondered how the performance of the processors relates to clock frequency and the distribution of other chips in the same group. Does raw clock speed account for the superior performance of the highest-scoring chips, or are architectural differences a more significant factor? Is a particular chip doing better or worse than the average performance curve? Figure 2 shows one way of illustrating this information. Using the normalized geometric means from Figure 1, we created an X-Y scatter-plot chart that measures clock frequencies along the X-axis and our EEMBCmark.auto/indy scores along the Y-axis.

What's most interesting about Figure 2 isn't the relative performance of the processors—the bar chart in Figure 1 shows that—but rather how the performance of each processor relates to its clock frequency and to the same metrics for other processors in the group. The shallow dotted line near the bottom of the chart represents a linear increase in performance with clock frequency, based on our reference V832 chip. In other words, if the V832's clock frequency were doubled, its score should fall on that line. (Of course, performance doesn't always scale in a linear relationship with clock frequency, but that's the ideal.) The fact that some chips in this group score well above that line indicates they are realizing a great deal of extra performance from their architectural and microarchitectural advantages over the V832. In other words, architecture matters. Indeed, the

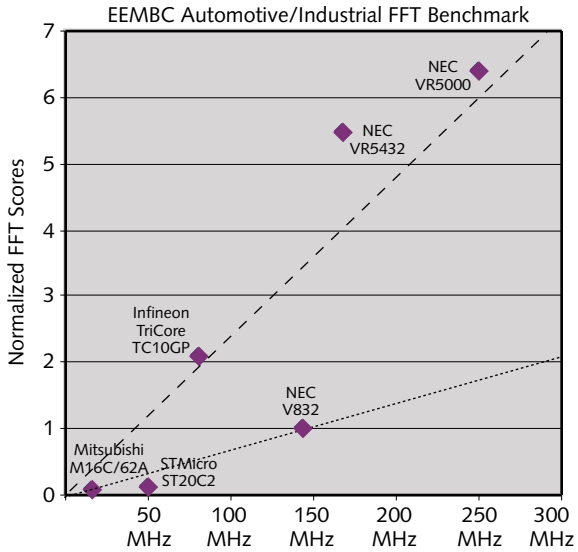

Figure 3. This chart uses only the FFT test results from EEMBC's automotive/industrial suite. Compare it with Figure 2 to see how the relative performance of the processors changes.

VR5432 and VR5000 do so much better than the frequency/performance line predicts that they easily justify the continued employment of CPU architects.

The steeper dotted line in Figure 2 is the average trendline, based on the scores of all the chips in this group. Above the trendline, chips exceed the average performance. Below the line, chips aren't living up to their potential to the same degree as their counterparts.

By that measure, the ST20C2 and V832 appear to be lazy underachievers, while the VR5432 and VR5000 are ambitious overachievers. This isn't just an academic exercise, because real-world performance at a given clock frequency is a good measure of efficiency, which is related to power consumption. (EEMBC currently doesn't benchmark power consumption, but it has established a working group to determine the proper methodology.)

Be careful, though. In this example, the V832 is relatively *more* power-efficient than the VR5432 (380mW vs. 555mW per EEMBCmark.auto/indy), an advantage that persists even after factoring out the VR5432's slightly higher clock frequency. But comparing performance to clock frequency yields quite a different result: the V832 scores only 0.006 EEMBCmarks per megahertz, while the VR5432 scores 0.026 EEMBCmarks per megahertz, nearly a 4x advantage. This apparent contradiction might be explained by the vast architectural differences between the two chips—the VR5432 is so much more complex that its power consumption rises on a steeper curve than both its clock frequency and its performance.

It's possible to generate an almost endless series of charts from EEMBC's raw data. Embedded developers who are focused on a specific application could create variations of the above charts that are based on subsets of the raw scores, or even on the results of individual tests. In fact, "virtual suites" could be created by picking and choosing relevant tests from any of EEMBC's official suites. These exercises could reveal different relationships among the processors. As a random example, we recomputed the X-Y scatter-plot chart using only the results from the fast-Fourier transform (FFT) test in the automotive/industrial suite. Figure 3 shows the results. NEC's MIPS-based processors still do very well, and Infineon's TriCore processor gets a higher score—probably because of its DSP-like architecture, which is well suited to this type of algorithm (see *MPR 4/19/99-02*, "Infineon's TriCore Tackles DSP").

Looking for the Suite Spot

Encouraged by the results of these analyses, we applied the same calculations to all of the EEMBC data released so far. We won't publish all the raw data here—as mentioned before, it's available on EEMBC's Web site—but the following charts illustrate the results of our calculations. Figure 4 shows our EEMBCmark.consumer scores for processors in the consumer-software suite.

There are only 5 tests in the consumer suite, compared with 16 tests in the automotive/industrial suite, so there weren't as many data points to work with. Nevertheless, some interesting results emerged. Once again, NEC's VR5432 demonstrated its architectural superiority over the V832, though not by the wide margin seen in the automotive/industrial suite. National Semiconductor's Geode GX1 fares poorly in this comparison, falling well short of the VR5432's performance, despite its 20% higher clock frequency and 20% higher price ($36 versus $30). The GX1 also doesn't come anywhere close to the performance of NEC's VR5000, whose clock speed is only 25% higher than the GX1's.

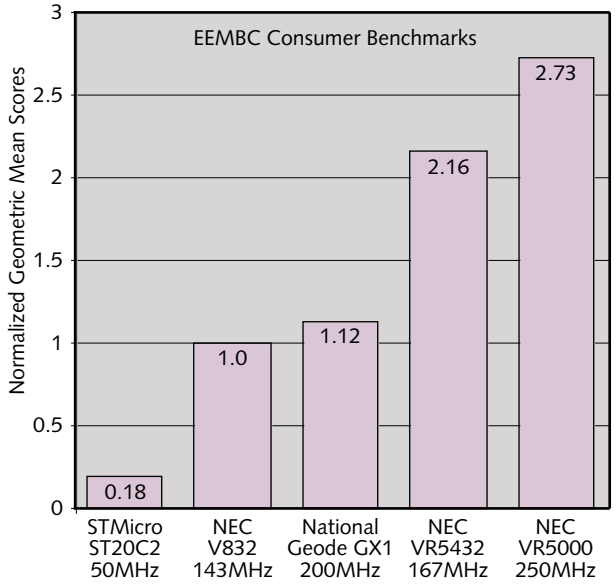

Figure 4. MDR derived these unofficial EEMBCmark.consumer scores from EEMBC's consumer-software benchmark results.

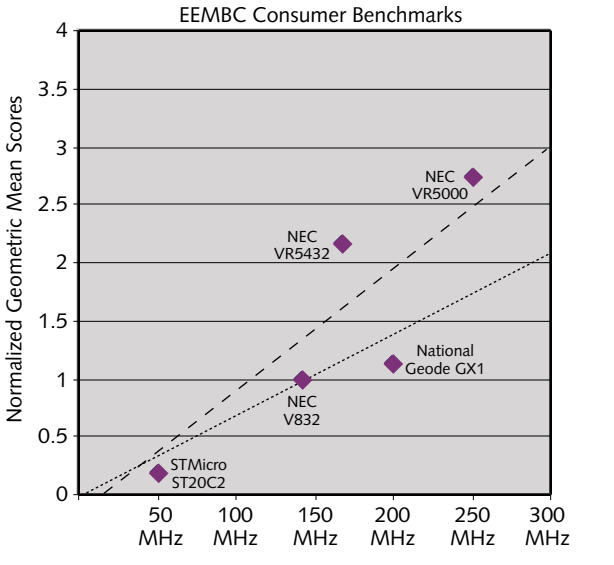

Figure 5. This chart reveals which processors are better or worse than average among other chips in the consumer-software suite.

Again, however, cold numbers don't tell the whole story. The GX1—a low-power version of Cyrix's former MediaGX—is a highly integrated chip that's a better solution for information appliances than either NEC chip, if lower overall system cost and x86 compatibility matter more than raw performance. The GX1 has an integrated PCI controller, memory controller, and 2D-graphics accelerator (see *MPR 3/10/97-01*, "MediaGX Targets Low-Cost PCs").

Figure 5 is a scatter-plot chart of the processors tested in the consumer suite. It confirms again that National's GX1 is an underachiever in terms of raw performance when compared with the MIPS-compatible processors, and that the VR5432 scores well above the average trendline.

EEMBC's networking suite consists of only five tests, and three of those five are almost identical: they measure a chip's ability to process packets in a theoretical router. The only difference is the size of the data stream: 512K, 1M, and 2M. The processor board used to test one chip in this suite, ST's ST20C2, didn't have enough memory to handle the 1M and 2M data streams, so we dropped that chip from this comparison. (Assigning zeros to the empty cells would have flattened the geometric means.) Even after leaving out the ST20C2, there are still seven chips left in this suite, the most in any suite for which EEMBC has released data. Figure 6 shows our derived EEMBCmark network scores.

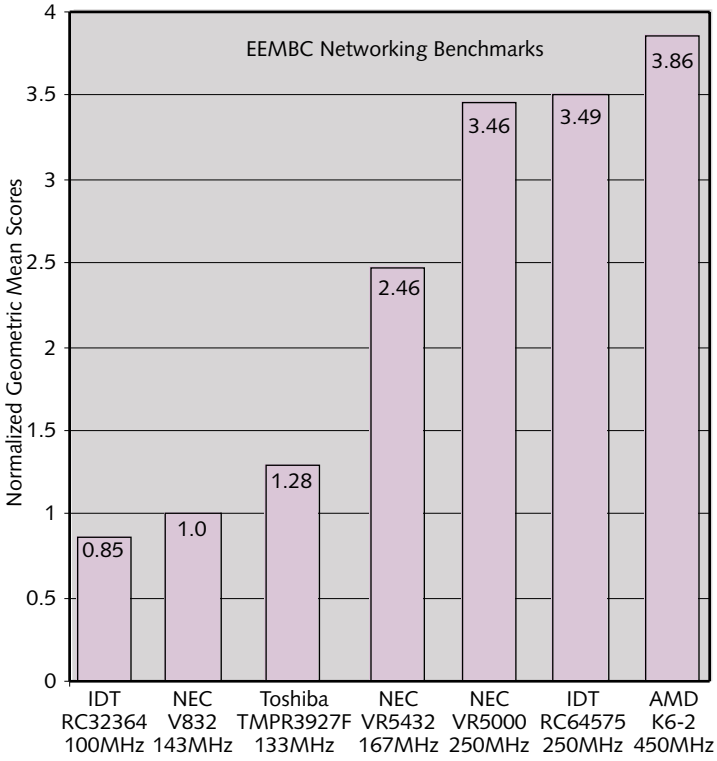

Figure 6. EEMBC released raw benchmark results for eight chips in the networking suite, but one of them—ST's ST20C2—was unable to run all of the tests, so we omitted it from this comparison.

Again, NEC's VR5432 exploits its architectural advantage over the V832. Another expected result is the virtual tie between NEC's VR5000 and IDT's RC64575; both processors are based on the same R5000-class dual-issue MIPS core (see *MPR 8/23/99-04*, "IDT Expands Embedded MIPS Family"). AMD's K6-2 appears to be the winner in this comparison, but note that its clock speed (450MHz) is much higher than the frequency of the RC64575 and VR5000 (250MHz). It spins 80% faster, but it delivers only 10% more performance. Maybe that newfangled RISC technology has a future after all. The K6-2, however, is cheaper: at $45, versus $54 for the VR5000 and $52 for the RC64575, it costs only a dime per megahertz.

The test results would seem to mark the K6-2 as a clock-frequency underachiever in this group, and Figure 7 lends some credence to that conclusion. The scatter-plot chart puts the K6-2 well below the average performance trendline. IDT's RC64575 and NEC's VR5000 merge into a single dot well above the trendline, and the VR5432 does well too. On the other hand, the K6-2 falls well above the linear frequency-performance line (the lower line on this chart), which is evidence of some architectural and microarchitectural advantages over the V832.

EEMBC's office-automation suite has even fewer tests than the networking suite—only four. They measure a processor's ability to perform a Bezier-curve calculation, dithering, image rotation, and text processing. Figure 8 shows our derived EEMBCmark.office scores.

NEC's 250MHz VR5000 fares less well in these tests against the VR5432 than it did in some other suites, and the V832 once again trails NEC's higher-end chips. It's curious that Toshiba entered the 20MHz TMP95FY64F and ST entered the ST20C2 in the office-automation derby, which is

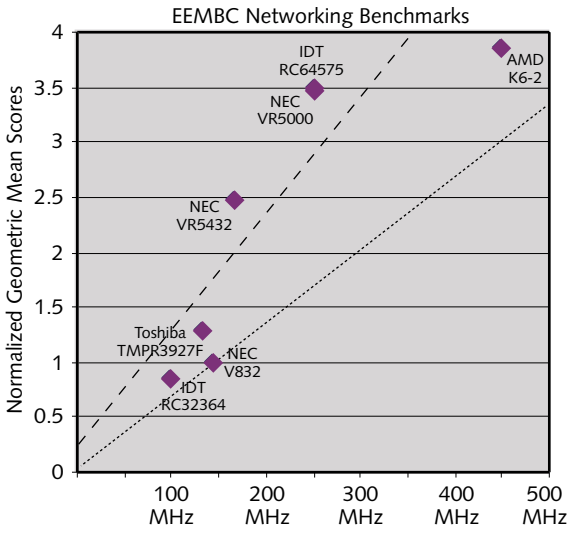

Figure 7. The chart of EEMBC's networking benchmarks shows that the performance of MIPS-compatible processors tends to scale much better than their clock frequencies would imply—especially when compared with the x86-compatible AMD K6-2.

oriented toward the kinds of tasks performed by laser printers and fax machines. Both chips are clearly out of their league when compared with the NEC processors. But ST, for one, positions the ST20C2 as a system monitor that depends heavily on system-level hardware accelerators to handle the heavy lifting, so perhaps these small CPUs are not entirely irrelevant in this category. Figure 9, a scatter-plot chart of the office-automation scores, shows the wide gap between the performance of the slowest and fastest chips in this suite.

The V832 does somewhat better in this scatter-plot comparison than in past examples, coming a little closer to hitting the average performance trendline. The superscalar VR5432 is superlative as usual.

EEMBC's telecommunications suite contains 16 benchmark tests, as many as found in the automotive/industrial suite. Many of the telecommunications tests use common algorithms with different data sets to simulate real-world conditions. Although no vendor subjected a dedicated network processor to these tests, there are a few RISC processors commonly found in routers and other networking equipment. These include IDT's RC32364 and RC64575, and NEC's VR5000 and VR5432. All those chips are based on MIPS-compatible cores. Figure 10 shows the results of the telecommunications tests.

Our EEMBCmark.telecom scores give the nod to the VR5000, which significantly outperforms the RC64575 by 30% at the same 250MHz clock frequency. This is hard to fathom at first, because as noted previously, both chips are based on the same MIPS R5000-class core. Both cores have dual-issue superscalar pipelines and equal amounts of primary cache. Yet the raw scores reported by EEMBC confirm that the VR5000 easily outran the RC64575 in all 16 tests in this suite.

To resolve the mystery, we examined the detailed reports that vendors must include when they submit benchmark results to EEMBC. One salient difference is that IDT ran the memory bus at only 50MHz, and NEC's memory bus ran at 100MHz. That would seem to give the VR5000 a big advantage. But looking further, IDT's memory subsystem used 3-1-1-1 timing versus NEC's much slower 11-1-1-1 timing. That means the RC64575 could fill a cache line in 120ns (6 bus cycles x 20ns at 50MHz), while the VR5000 needed 140ns (14 bus cycles x 10ns at 100MHz), which wiped out the advantage of NEC's 2x faster bus.

Therefore, we think the most likely explanation for the difference in performance between these nearly identical processors is that NEC compiled the benchmarks with the Green Hills Multi2000, while IDT used a GNU compiler. GNU compilers are reliable but generally don't produce the fastest executable code. Also, the compiler flags reported by the two companies indicate that NEC used more aggressive optimizations than IDT did, including a CPU-specific flag that optimizes code for the VR5000's superscalar microarchitecture. IDT used only two compiler flags with GNU: one for O2-level compiler optimizations and another that specifies the MIPS-IV instruction-set architecture (but not the R5000 microarchitecture). That alone could account for the difference in tested performance—the GNU compiler probably didn't know it could schedule certain pairs of instructions for parallel execution in the RC64575's superscalar pipelines.

EEMBC's stringent rules for reporting compilers, flags, and other data make its benchmark results even more respectable, because it's possible to explore mysteries like this. But the data also takes more effort to interpret than a simple Dhrystone score because there's so much more of it.

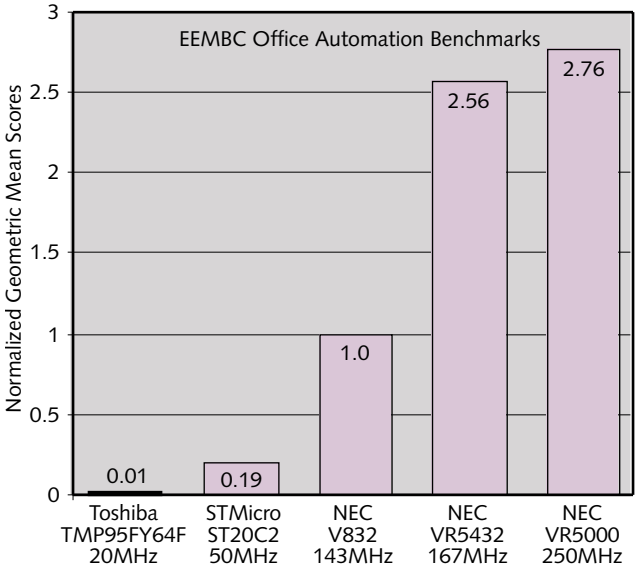

Figure 8. NEC's MIPS-compatible processors excel at running EEMBC's office-automation benchmarks.

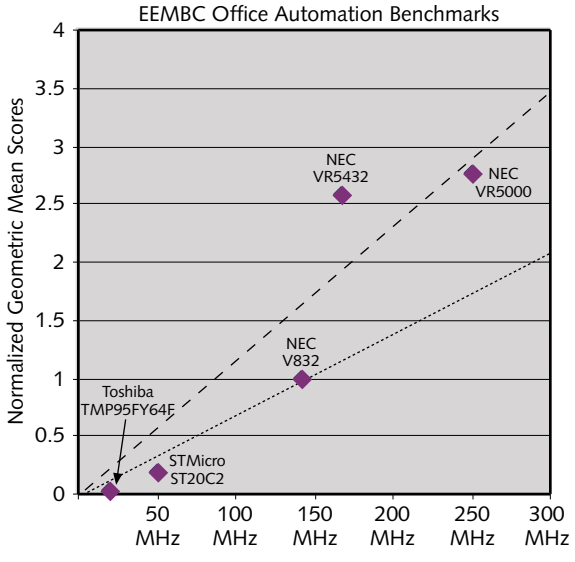

Figure 9. This chart of EEMBC's office-automation benchmarks confirms again that NEC's VR5432 is a clock-frequency overachiever.

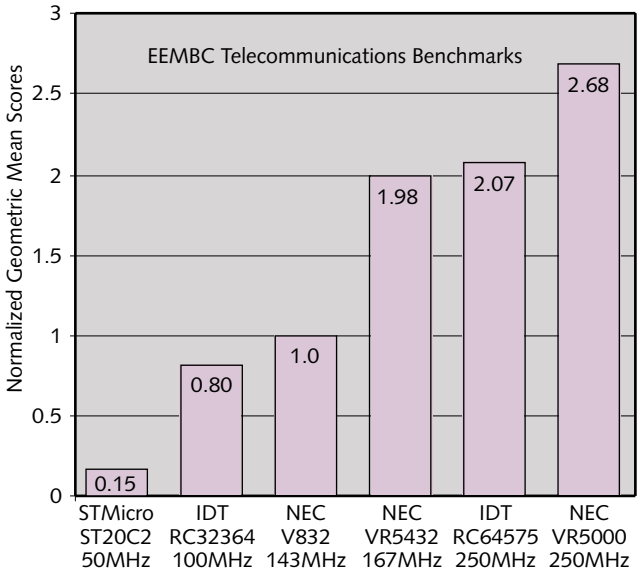

Figure 10. In EEMBC's telecommunications suite, NEC's VR5000 has a clear advantage over IDT's RC64575 at the same clock frequency.

Compiler differences might also be part of the reason that NEC's 167MHz VR5432 performs almost as well as the 250MHz RC64575 in the telecommunications suite, despite its 50% slower clock speed. This time, NEC used Apogee Software's compiler instead of Multi2000. But the VR5432 also has some architectural advantages over the RC64575, including a larger array of function units. Whatever the explanation, it appears that the RC64575 is not performing to its full potential in this suite. Figure 11, the scatter-plot chart, makes this clear.

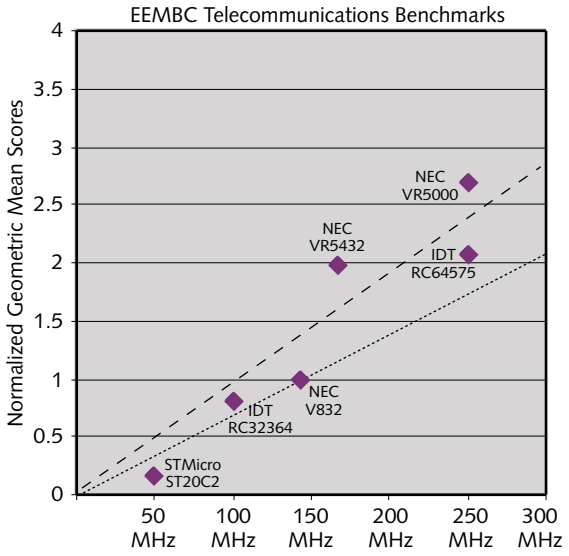

Figure 11. Charting the telecommunications benchmarks on a scatter-plot graph illustrates the significant performance difference between the VR5000 and RC64575, even though they're based on the same MIPS R5000-class core.

For More Information

More information and complete benchmark results for all five test suites are available on EEMBC's Web site at *www.eembc.org*.

Indeed, the scatter plot reveals that the RC64575's performance doesn't scale as well with clock frequency as the performance of the VR5000 and VR5432, though it scales better than the V832. IDT's RC32364 is a bit below the curve too, though not by quite as much. IDT also used a GNU compiler with the RC32364. It would be interesting to compare these results with tests run on IDT's processors using a different compiler. Embedded developers can use the EEMBC scores to compare development tools, not just processors.

More Scores to Come

These initial EEMBC results are so-called "out of the box" scores. Vendors aren't allowed to do any tweaking on the test programs, other than fiddle with compiler flags. In June, at Embedded Processor Forum, EEMBC plans to release the first wave of so-called full fury scores—highly optimized tests in which almost anything is allowed. Vendors can rewrite the C source code of the benchmark programs, optimize critical loops with assembly language, and take advantage of special hardware features in their chips to speed up algorithms.

Allowing vendors to use such heavy-handed optimizations would seem to encourage cheating, but EEMBC has an elaborate system of safeguards to prevent the kinds of abuses that have occasionally caused scandals with other benchmarks. All results are verified by EEMBC Certification Laboratories (ECL), an independent organization that goes as far as checking source code and repeating the tests with private data sets (for more details, see the article referenced earlier, "Embedded Benchmarks Grow Up"). EEMBC's license agreement—which vendors must sign to get the benchmark source code—forbids vendors to make their results public without ECL certification, although they can privately share uncertified data with customers under NDA.

Some critics say the full-fury scores will reduce EEMBC's benchmarks to a programming contest. However, we believe they will reflect the optimizations commonly used by developers in real-world embedded applications. Comparing the full-fury scores to the out-of-the-box scores for the same processors in the same benchmark suites should yield some fascinating conclusions.