White Paper

Advantage Engineering LLC

August 2011

Low-Power Processors:
When Performance Matters

VIA Nano™ X2 compared to Intel® Atom™

J. Scott Gardner
President and Principal Analyst
Advantage Engineering LLC

gardner@advantage-engineer.com

VIA Nano™ X2 compared to Intel® Atom™

Contents

Executive Summary

This white paper takes a fresh look at the performance of low-power x86 CPUs, now that VIA has introduced dual-core and quad-core CPUs. Since many benchmarks are multi-threaded, Intel Atom processors benefit from dual-core and hyper-threading to compete with VIA's performance-oriented Nano architecture. However, a technical comparison of Nano and Atom microarchitectures shows that the VIA Nano is designed for higher performance (50% more instructions/clock on each thread). For multiple threads, the VIA Nano X2 and Quadcore should outperform Atom-based dual-core CPUs. This white paper presents testing results that confirm Nano's raw performance advantage over the fastest Atom CPUs.

The following charts highlight results from performance tests of Atom D525 (1.8GHz) and VIA Nano X2 U4300 (1.2+GHz). Both CPUs have a TDP of 13 watts. Even though the Atom operates at a 50% higher base frequency, the VIA Nano X2 scores almost 40% higher on SPEC® CPU2000. Using only 2 cores and lower clock rates, Nano X2 has similar 4-thread performance to the hyper-threaded, dual-core Atom. In addition, the data also shows that an application with 4 threads would benefit from the extra 2 CPUs in the VIA QuadCore.

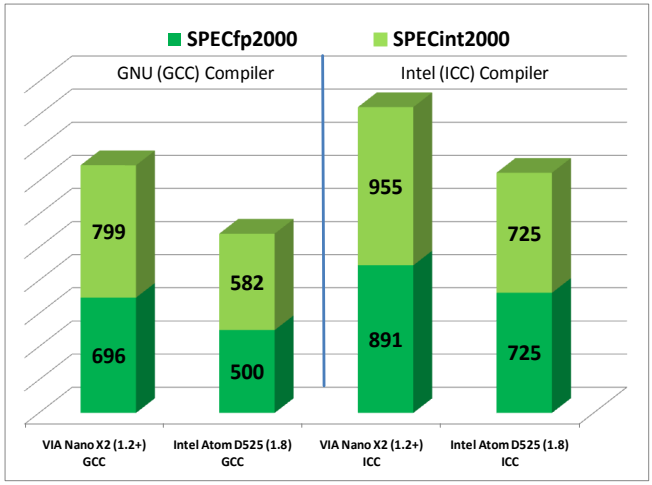

Figure 1 SPEC CPU2000 Single-thread Performance (Peak Speed with GCC and Intel compilers)

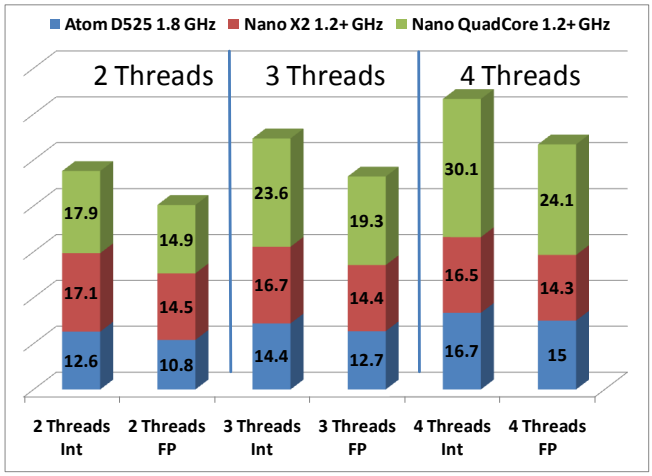

Figure 2 SPEC CPU2000 Multi-thread Performance (Peak Rate with GCC Compiler)

Introduction

Intel's Atom processor changed customer expectations for CPU performance. With power consumption (and to a lesser extent, cost) driving new designs, Intel developed a product family for "good enough" performance, and power-constrained markets skyrocketed, while PC growth rates stagnated. VIA's power-efficient processors (such as the VIA C7) originally dominated the nascent markets for lower-power x86, but Intel deserves credit for validating these markets and helping shift

the entire computing industry to embrace power efficiency.

Mobile computing devices and power-efficient embedded applications have rapidly matured into markets that are poised for another burst of innovation. The renewed focus on performance is driven by competitive pressure from multi-GHz ARM processors. With so many vendors of ARM-based processors, suppliers are seeking differentiation through architectural enhancements, higher clock rates and multiple cores. The marketing focus has returned to maximizing performance, yet every market segment will enforce its own hard limits on power consumption.

Intel Atom performance trade-offs

Intel's Atom was designed with an architecture that sacrifices performance to reduce power and cost. For Intel's target markets, the designers made excellent trade-offs to build a power-efficient CPU that has been very successful. Ironically, the Atom design approach is now facing the same limitations that motivated VIA's Centaur design team to develop the Nano microarchitecture. Like the venerable VIA C7, Atom's in-order pipeline limits the performance available on each core. The Atom processor addressed this performance limitation by including hyper-threading and dual-core, but VIA's Nano family raises the bar again by launching dual-core (Nano X2) and VIA QuadCore processors. As "good enough" performance gives way to marketing demands for benchmark winners, the Nano and Atom will be compared by system designers looking for a performance edge. This whitepaper revisits the microarchitectural advantage maintained by Nano, while also analyzing software workloads that demonstrate the Nano X2 performance lead over a dual-core Atom with hyper-threading.

The author acknowledges that raw performance is only one vector of CPU comparison, and Intel uses leading-edge process technology and design features to target extremely low-power markets with some of its Atom-based products. However, this white paper demonstrates that multicore Nano should deliver higher performance on most applications, assuming that both Atom and Nano CPUs meet the target system design constraints for power, cost, level of integration, etc.

Nano Microarchitecture: Still faster

Comparing microprocessors is often complicated by marketing efforts to carefully select benchmarks that draw attention away from the technical realities of the underlying design. Since Nano and Atom inhabit a diverse realm of applications with a wide range of system architectures and software workloads, standard benchmarks are unlikely to represent the end-user experience for any specific design. Also, many benchmarks do not provide full transparency of the process for selecting software workloads. However, with some knowledge of the CPU microarchitecture, system architects can predict the throughput for either of these processors. The key is to identify design performance bottlenecks for various workloads, based on an objective analysis of the CPU design choices that affect how each processor executes instructions and moves data. Microprocessor analysts[1] agree that the Nano microarchitecture is capable of much higher performance than Atom. The most-obvious advantage is Nano's out-of-order design and three x86 instructions per clock, while Atom is restricted to two instructions per clock and is limited by the in-order pipeline.

[1] Tom Halfhill, Microprocessor Report, The Linley Group, (08/30/2010)

Hitting a wall with in-order pipelines

The definition of an in-order CPU pipeline is the limitation that instructions must be fetched, executed, and retired in consecutive order. If a resource (registers, memory or execution units) is being used by other instructions, that portion of the pipeline stalls. Other instructions pile up in various storage queues that wait for the in-order pipeline to complete the instruction that caused the stall. While the Atom can fetch 2 instructions at a time, the CPU has restrictions on how the instructions are issued to the 2 different execution ports, since only certain types of instructions can be handled by each processing unit. For instance, only one of the execution ports accepts instructions that access memory.

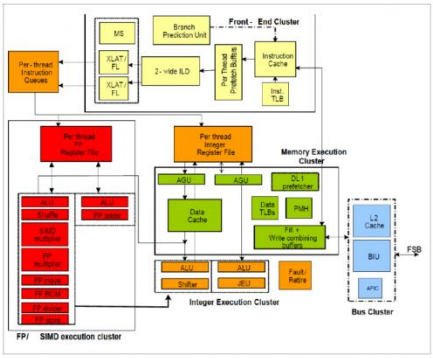

Figure 3 Intel® Atom™ Block Diagram.

The Atom compiler will attempt to statically pair instructions to avoid resource conflicts, but the in-order requirement takes away the ability for the CPU to skip ahead in the instruction flow to find instructions that take advantage of available computing resources. Also, without the register renaming feature of an out-of-order architecture, it is very difficult to scale up Atom performance by adding more execution resources or fetching more instructions each cycle. All instructions (in a thread) share a single integer or floating-point register file. Any later instructions that affect the same registers have to wait in line for the first instruction to finish.

Decades ago, computer scientists created architectural models of an in-order pipeline's diminishing returns for instruction-level parallelism[2]. There are many more Atom "pipeline hazards" than the obvious cases discussed here, and Intel designers have cleverly managed the trade-offs.

Settling for thread-level parallelism

The Nano microarchitecture is designed to yield better instruction-level parallelism (ILP) than Atom, simply because Nano is able to process 50% more x86 instructions simultaneously in its out-of-order pipeline (up to 3 per clock). Intel chose to accept limitations on instruction-level parallelism and focus instead on thread-level parallelism. The Atom architecture is well-suited to hyper-threading. Atom processors that support hyper-threading can execute instructions from a second thread during a pipeline stall, leading to better utilization of Atom's dual-fetch architecture. Since the in-order pipeline is likely to stall while waiting for data, the CPU can shift to a different set of registers for processing a separate instruction stream. Note that single-threaded performance may drop even lower when a core has hyper-threading enabled, since bandwidth and compute resources are shared. The in-order pipeline limitations will be especially severe when multiple threads frequently access main memory, since both execution queues will be blocked and waiting for data.[3]

[2] Hennessy & Patterson, Computer Architecture: A Quantitative Approach.

[3] Atom's lower SPECint2000 score on the place-and-route simulator (300.twolf) is a likely example of this cache-miss penalty.

Nano: ILP makes threads run faster

For multi-tasking and applications with thread-level parallelism, the Atom architecture has been able to overcome the lower ILP of its in-order pipeline. However, in system designs where Nano X2 fits in the power budget, Intel's Atom now has to be compared to another multicore processor. With 50% higher peak instructions/clock, each Nano core should be faster on each instruction thread. Normalized for clock rate, the SPEC CPU2000 testing demonstrated even more than the 50% advantage per-thread predicted by architectural analysis. With higher ILP and multiple cores, a wider range of applications will run faster – not just the multi-threaded ones.

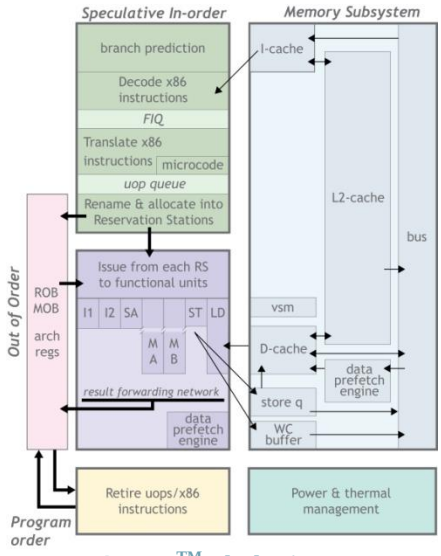

Figure 4 VIA Nano™ Block Diagram.

Nano's other high-performance features

For computer architects, the primary goal is to balance memory bandwidth with "compute bandwidth", and the Nano architecture has a range of memory and bus features that are designed for performance – large, multi-way caches, advanced branch prediction, data pre-fetching, etc. This white paper glosses over many of the key features of the Nano architecture (Virtualization, adaptive power management, encryption, etc.). For readers looking for a deeper dive into details, Glenn Henry (Nano chief architect and president of VIA's Centaur Technologies design center) authored a technical white paper when Nano's architecture (Isaiah) was first launched: (http://www.via.com.tw/en/downloads/whitepapers/processors/WP080124Isaiah-architecture-brief.pdf).

Performance in the Power Budget

– not just performance-per-watt

Intel's in-order Atom pipeline and advanced fabrication technology result in lower absolute power consumption than any modern x86 processor, including VIA's Nano. However, Nano is still a low-power processor family, and the newest multicore CPUs are within the power budget for the majority of x86-based system designs. For instance, consider the large number of designs that allow a CPU with 13-watt thermal design power (TDP). This whitepaper compares VIA Nano™ X2 U4300 (1.2+GHz) to Intel® Atom™ Processor D525 (1.8GHz). Both CPUs operate below 13 watts, but VIA's CPU uses Adaptive Overclocking™ power management to bring the clock rate to over 1.4GHz while maximizing the performance within the 13-watt power envelope.

Any power comparisons are complicated by system issues. For instance, the Atom D525 TDP includes power consumed by the memory controller. However, the entire VIA VX900 chipset consumes less than 4 watts, so this system partitioning issue shouldn't be a factor in the cooling requirements for the CPU or a major factor in total system power. Note that the average power of the D-series Atom may be much higher than Nano, since this Atom family does not include Enhanced Intel

Speedstep® Technology and will never shift into lower P-States to reduce power.

Comparing against the fastest Atoms

At first glance, it may not seem reasonable to compare Nano to a 13-watt Atom, since most other versions of Atom have much better performance per watt. However, system designers likely choose the D525 because the device is one of the highest-performance Atom CPUs on the market with 2 cores and 4 threads, running at 1.8GHz. (Intel currently ships single-core Atoms that operate up to 2.13GHz.) With 800MHz DDR3, the Atom D525 is matched with its fastest memory interface. Instead of artificially limiting the memory speed of the Nano X2, these tests configure the Nano X2 platform with DDR3-1066Mhz memory. If a system designer is concerned about performance, then the system configuration should balance the memory system with the CPU choice. The integrated DRAM controller on the Atom gives the benefit of slightly lower memory latency, but the Nano-based system will have 33% higher memory bandwidth when configured with 1066Mhz memory.[4]

Comparing SPEC CPU2000 performance

While SPEC CPU2006 is the latest version (http://spec.org/cpu2006), Atom results can only be publicly analyzed using retired versions of SPEC. To use SPEC CPU2006 would require a vendor to make an official submission to the spec.org database, something that has never been done for Atom-based systems. The SPEC CPU2000 test suite is a group of system applications that have well-documented characteristics. The SPEC benchmark is more transparent than consumer-oriented benchmarks, since every application is compiled from source code, and spec.org has strict disclosure rules.

While many of the SPEC applications represent high-end workloads that are not practical for most power-constrained products, this white paper focuses on performance. For a 13-watt TDP, which CPU is faster? Running SPEC CPU2000 helps answer that question in a transparent manner.

Avoiding a War of Compilers and Flags

Since SPEC is so important to computer companies, the compilers are modified to produce ever-improving results on the same hardware platform. In addition, the SPEC configuration file is usually tweaked to find the optimum compiler flags that give the highest scores. For this white paper, the SPEC CPU2000 scores used an identical configuration file for both Intel Atom and VIA Nano. Even though the SPEC scores are "Peak" runs, the flags were not optimized, though VIA Nano was allowed to use its SSE 4.1 instruction extensions. These are the GCC flag changes:

- ***Intel Atom:***
  PEAK_FLAGS = -march=atom -mfpmath=sse -msse3 -funroll-loops -fno-strict-aliasing

- ***VIA Nano:***
  PEAK_FLAGS = -march=core2 -mfpmath=sse -msse4.1 -funroll-loops -fno-strict-aliasing

Note also that the VIA Nano was run as an Intel Core2 architecture, so any built-in compiler optimizations should also take advantage of the Nano out-of-order architecture.

[4] Some vendors show the Intel D525MW motherboard with DDR3-800/1066 support. Faster memory would get down-clocked to 800MHz.

Credit Tolly Group for configuration file

The engineers at The Tolly Group (http://tolly.com/) deserve thanks for providing the configuration file from their tests of the Intel Atom Z530 in Tolly Test Report #210100 (March 2010). This configuration file was used to generate the SPEC CPU2000 results in this white paper. The Tolly study is unlikely to include optimizations that unfairly limit Atom performance, so readers of this white paper should be confident that Atom was tested in a consistent and transparent manner. Note that the Tolly configuration file was created for a different version of Linux and only used the Intel compiler. The Tolly engineers could likely optimize the D525 tests for higher scores, just as the VIA engineers have indicated that they could tweak the process for higher Nano scores. However, the goal isn't to generate the highest scores on a retired benchmark. The results are only relevant because the performance testing was consistent for both 13W CPUs and demonstrated performance differences that closely match the architectural comparisons.

SPEC CPU2000 Speed

Figure 1 shows the "Speed" scores for SPECint2000. Appendix A discloses the detailed data tables from these tests, and Appendix B provides the configuration data about the test systems and software testing environment. SPEC Speed tests spawn a single thread for each application. The slight multicore performance improvement is likely a benefit from extra CPUs processing operating system tasks. The Intel compiler generated approximately 20% higher scores than the GCC compiler. The Intel compiler showed a similar advantage for VIA Nano when compiling code for an Intel Core2 architecture. While the Intel compiler produced excellent results (at least on this public benchmark), VIA believes that a majority of its embedded system customers develop with GCC to save cost.

SPEC CPU 2000 Integer Applications

1. *Compression*
2. *FPGA Circuit Placement and Routing*
3. *C Programming Language Compiler*
4. *Combinatorial Optimization*
5. *Game Playing: Chess*
6. *Word Processing*
7. *Computer Visualization*
8. *PERL Programming Language*
9. *Group Theory, Interpreter*
10. *Object-oriented Database*
11. *Compression*
12. *Place and Route Simulator*

Spec CPU 2000 Floating-Point Applications

1. *Physics / Quantum Chromodynamics*
2. *Shallow Water Modeling*
3. *Multi-grid Solver: 3D Potential Field*
4. *Parabolic / Elliptic Partial Differential Equations*
5. *3-D Graphics Library*
6. *Computational Fluid Dynamics*
7. *Image Recognition / Neural Networks*
8. *Seismic Wave Propagation Simulation*
9. *Image Processing: Face Recognition*
10. *Computational Chemistry*
11. *Number Theory / Primality Testing*
12. *Finite-element Crash Simulation*
13. *High Energy Nuclear Physics Accelerator Design*
14. *Meteorology: Pollutant Distribution*

SPEC CPU2000 Rate

The SPEC benchmarks produce a "Rate" score by running multiple copies of each application. Each "user" represents another independent thread of code, offering a representative view of how a CPU would multitask (or run multi-threaded applications that don't operate on shared data). Since SPEC CPU2000 applications focus on CPU

performance, multiple threads are unlikely to have resource conflicts for main memory and I/O. For this reason, the scores scale extremely well with additional CPUs -- making SPEC Rate a popular benchmark for vendors of high-end CPUs.

No dual-thread Hyper-Threading benefit

When running SPEC CPU2000 Rate with just 2 threads, testing showed that the dual-core Atom did not gain any benefit from Hyper-Threading.

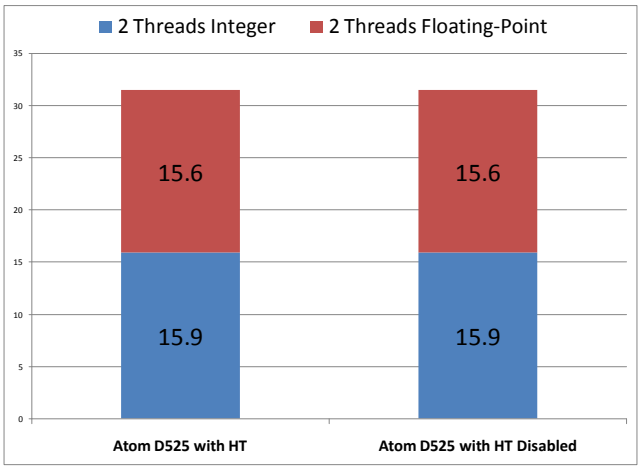

Figure 5 No change in 2-user CPU 2000 Rate score with Atom Hyper-Threading.

However, as shown earlier in Figure 2, Hyper-Threading provides substantial benefit to the Atom architecture when running 3 or more threads. In fact, without adjusting for the higher clock rate, the Atom D525 slightly surpasses the dual-core Nano U4300 in performance for 4 threads. Both CPUs have 2 physical cores, but the Nano is swapping between 2 threads on each core.

For Best Performance, Use More Cores

Unless the clock rate needs to drop in order to save power, applications that need high performance on more than 2 threads should have more physical CPU cores. The Nano-based systems can upgrade to VIA's pin-compatible QuadCore family (also pin-compatible with the single-core Nano). While Hyper-Threading definitely helps the dual-core Atom, the VIA QuadCore is approximately 80% faster than a Hyper-Threaded Atom on 4 threads. The 4 dedicated CPU cores allow each thread to run on a dedicated CPU with dedicated L1 and L2 caches. If 4-thread performance is an important design requirement, then the design is likely to have the power budget for a 4-core CPU (27.5 watts for the QuadCore L4700 @ 1.2+ GHz used in these tests).

Conclusions

Intel clearly made a good decision to build multicore Atom processors and incorporate hyper-threading. Even though the Atom processor architecture was not designed for raw performance, the higher performance on multi-threaded applications has allowed Intel to stake out a performance claim in the low-power processor space. Now that VIA (and AMD) have introduced multicore CPUs for power-constrained x86 designs, the competitive landscape needs to be reevaluated. VIA's higher-performance architecture should offer an advantage in systems with a power budget that allows dedicated CPU cores for each major software thread required by the application.

The SPEC CPU2000 benchmarks provided an interesting software tool for analyzing CPU performance on high-performance workloads, and the results validated the architectural analysis. However, the best measure of performance will always be found by running the actual code that users implement on the microprocessor system.

Appendix A. Data Tables

This section provides more details about the result of the SPEC performance testing. All of the tests used SPEC CPU2000, a retired benchmark from the Standard Performance Evaluation Corporation (SPEC – http://www.spec.org). Since SPEC CPU2000 has been retired, these are non-official scores.

SPEC CPU2000 Speed (Intel 11.1 compiler with profile-guided optimizations)

	Intel Atom D525 1.8Ghz HT ICC11.1		Intel Atom D525 1.8Ghz noHT ICC11.1		VIA Nano X2 U4300 @ 1.2+ ICC 11.1	
	Base Runtime	Base Ratio	Base Runtime	Base Ratio	Base Runtime	Base Ratio
SPEC INT		**725**		**725**		**955**
164.gzip	231	607	231	607	180	779
175.vpr	331	423	332	422	187	749
176.gcc	112	985	112	982	91.7	1200
181.mcf	237	758	238	757	395	455
186.crafty	152	659	152	659	95.5	1047
197.parser	265	679	265	679	227	793
252.eon	148	881	148	880	113	1148
253.perlbmk	208	867	207	868	144	1246
254.gap	131	843	130	843	108	1023
255.vortex	156	1215	156	1215	113	1678
256.bzip2	277	541	277	542	210	713
300.twolf	515	582	512	586	242	1237
SPEC FP		**725**		**724**		**891**
168.wupwise	152	1053	152	1054	116	1385
171.swim	265	1172	265	1171	265	1168
172.mgrid	389	462	390	462	463	389
173.applu	308	682	308	683	251	835
177.mesa	216	648	218	644	129	1085
178.galgel	283	1024	283	1024	174	1667
179.art	142	1825	142	1827	176	1481
183.equake	118	1100	118	1098	194	670
187.facerec	196	970	197	964	199	954
188.ammp	603	365	601	366	343	642
189.lucas	206	973	206	972	195	1025
191.fma3d	295	712	295	713	254	828
200.sixtrack	469	235	469	235	248	443
301.apsi	704	370	702	370	283	920

SPEC CPU2000 Speed (GCC 4.5.1 compiler with profile-guided optimization)

	Intel Atom D525 1.8Ghz HT GCC4.5.1 (pgo)		VIA Nano X2 U4300 @ 1.2+ GCC4.5.1 (pgo)		VIA QuadCore L4700 @ 1.2+ GCC4.5.1 (pgo)	
	Base Runtime	Base Ratio	Base Runtime	Base Ratio	Base Runtime	Base Ratio
SPEC INT		**582**		**799**		**797**
164.gzip	283	495	190	737	190	736
175.vpr	382	367	222	631	222	630
176.gcc	140	783	117	944	118	932
181.mcf	394	457	526	342	526	342
186.crafty	138	722	81.6	1225	81.6	1225
197.parser	370	487	303	594	303	593
252.eon	165	786	107	1212	107	1212
253.perlbmk	243	741	164	1094	165	1093
254.gap	162	680	130	844	131	843
255.vortex	234	812	176	1079	176	1078
256.bzip2	324	463	232	648	231	648
300.twolf	674	445	381	788	381	787
SPEC FP		**500**		**696**		**694**
168.wupwise	300	534	217	739	217	739
171.swim	871	356	481	644	484	640
172.mgrid	548	328	501	359	504	357
173.applu	431	488	298	704	302	696
177.mesa	250	559	141	991	141	991
178.galgel	367	791	293	990	292	994
179.art	324	802	403	645	403	645
183.equake	185	702	266	488	268	485
187.facerec	274	694	208	912	209	907
188.ammp	620	355	314	701	314	700
189.lucas	276	724	185	1084	188	1064
191.fma3d	385	545	276	761	277	758
200.sixtrack	476	231	294	374	294	375
301.apsi	735	354	303	858	303	859

SPEC CPU 2000 Rate (2 threads) with GCC compiler

	Intel Atom D525 1.8Ghz HT GCC4.5.1 (pgo)		VIA Nano X2 U4300 @ 1.2+ GCC4.5.1 (pgo)		VIA QuadCore L4700 @ 1.2+ GCC4.5.1 (pgo)	
	Base Runtime	Base Ratio	Base Runtime	Base Ratio	Base Runtime	Base Ratio
SPEC INT		**12.6**		**17.1**		**17.9**
164.gzip	284	11.4	208	15.6	190	17.1
175.vpr	413	7.86	238	13.6	231	14.1
176.gcc	158	16.2	124	20.7	120	21.3
181.mcf	542	7.7	600	6.96	601	6.95
186.crafty	140	16.6	88.3	26.3	81.8	28.4
197.parser	386	10.8	324	12.9	313	13.3
252.eon	166	18.2	115	26.1	107	28.1
253.perlbmk	245	17	181	23.1	166	25.2
254.gap	169	15.1	143	17.8	140	18.2
255.vortex	241	18.3	189	23.3	179	24.7
256.bzip2	338	10.3	250	13.9	237	14.7
300.twolf	752	9.25	405	17.2	397	17.5
SPEC FP		**10.8**		**14.5**		**14.9**
168.wupwise	305	12.2	242	15.4	233	15.9
171.swim	874	8.23	537	13.4	542	13.3
172.mgrid	558	7.48	554	7.53	550	7.59
173.applu	447	10.9	347	14	340	14.3
177.mesa	252	12.9	155	20.9	142	22.9
178.galgel	386	17.4	315	21.4	307	21.9
179.art	561	10.8	461	13.1	461	13.1
183.equake	194	15.6	300	10	304	9.92
187.facerec	286	15.4	232	19	236	18.7
188.ammp	671	7.6	335	15.2	319	16
189.lucas	295	15.7	220	21.1	221	21
191.fma3d	401	12.1	301	16.2	298	16.3
200.sixtrack	477	5.35	325	7.86	294	8.67
301.apsi	782	7.72	327	18.5	307	19.7

VIA Nano™ X2 compared to Intel® Atom™

SPEC CPU 2000 Rate (3 threads) with GCC compiler

	Intel Atom D525 1.8Ghz HT GCC4.5.1 (pgo)		VIA Nano X2 U4300 @ 1.2+ GCC4.5.1 (pgo)		VIA QuadCore L4700 @ 1.2+ GCC4.5.1 (pgo)	
	Base Runtime	Base Ratio	Base Runtime	Base Ratio	Base Runtime	Base Ratio
SPEC INT		**14.4**		**16.7**		**23.6**
164.gzip	353	13.8	313	15.6	224	21.7
175.vpr	546	8.92	371	13.1	267	18.3
176.gcc	230	16.7	190	20.1	135	28.4
181.mcf	720	8.7	905	6.92	657	9.53
186.crafty	190	18.3	135	25.9	96.5	36
197.parser	505	12.4	499	12.6	362	17.3
252.eon	214	21.1	174	26	119	38.1
253.perlbmk	324	19.4	273	22.9	189	33.2
254.gap	215	17.8	218	17.6	160	23.9
255.vortex	348	19	290	22.8	201	33
256.bzip2	417	12.5	382	13.7	267	19.6
300.twolf	951	11	650	16.1	458	22.8
SPEC FP		**12.7**		**14.4**		**19.3**
168.wupwise	385	14.5	363	15.4	261	21.3
171.swim	1209	8.92	807	13.4	619	17.4
172.mgrid	716	8.74	846	7.41	623	10.1
173.applu	527	13.9	521	14	408	17.9
177.mesa	336	14.5	234	20.8	167	29.1
178.galgel	468	21.5	482	20.9	351	28.8
179.art	906	9.99	697	13	533	17
183.equake	234	19.3	453	9.99	339	13.3
187.facerec	370	17.9	352	18.8	261	25.3
188.ammp	806	9.49	511	15	377	20.3
189.lucas	369	18.8	331	21.1	272	25.6
191.fma3d	485	15.1	452	16.2	345	21.2
200.sixtrack	571	6.7	490	7.82	347	11
301.apsi	952	9.51	494	18.3	358	25.2

VIA Nano™ X2 compared to Intel® Atom™

SPEC CPU 2000 Rate (4 threads) with GCC compiler

	Intel Atom D525 1.8Ghz HT GCC4.5.1 (pgo)		VIA Nano X2 U4300 @ 1.2+ GCC4.5.1 (pgo)		VIA QuadCore L4700 @ 1.2+ GCC4.5.1 (pgo)	
	Base Runtime	Base Ratio	Base Runtime	Base Ratio	Base Runtime	Base Ratio
SPEC INT		**16.7**		**16.5**		**30.1**
164.gzip	379	17.1	417	15.6	229	28.4
175.vpr	608	10.7	516	12.6	272	23.9
176.gcc	290	17.6	254	20.1	141	36.1
181.mcf	927	9.01	1214	6.88	724	11.5
186.crafty	222	20.9	179	25.9	98.3	47.2
197.parser	573	14.6	682	12.2	369	22.6
252.eon	230	26.2	231	26.1	129	46.9
253.perlbmk	367	22.8	365	22.9	197	42.3
254.gap	242	21.1	291	17.6	168	30.4
255.vortex	401	22	396	22.3	212	41.6
256.bzip2	461	15.1	512	13.6	280	24.8
300.twolf	1080	12.9	914	15.2	465	30
SPEC FP		**15**		**14.3**		**24.1**
168.wupwise	415	17.9	485	15.3	268	27.7
171.swim	1479	9.72	1067	13.5	677	21.3
172.mgrid	782	10.7	1148	7.28	648	12.9
173.applu	595	16.4	702	13.9	474	20.5
177.mesa	377	17.2	312	20.8	171	38.1
178.galgel	543	24.8	649	20.7	364	37
179.art	1291	9.35	934	12.9	610	19.8
183.equake	259	23.3	602	10	362	16.7
187.facerec	415	21.2	473	18.7	274	32.2
188.ammp	883	11.6	695	14.7	376	27.2
189.lucas	416	22.3	442	21	332	27.9
191.fma3d	505	19.3	603	16.2	370	26.3
200.sixtrack	594	8.59	650	7.85	353	14.5
301.apsi	1032	11.7	658	18.3	370	32.6

Appendix B. Configurations for Testing

This section discloses details about the methodology for the performance tests. All of the tests were conducted at VIA's Austin design center (Centaur Technology), using hardware provided by Centaur and with the assistance of Centaur engineers. Advantage Engineering LLC was contracted by Centaur to analyze VIA multicore processors and create a 3rd-party white paper. The author was responsible for selecting the benchmarks and system configurations for testing. A subset of the testing results were audited (re-run with confirmation of the software configuration) to validate the process and verify that the results could be reproduced. The author encourages interested parties to conduct their own testing.

Hardware platforms used for testing

The following table discloses details for the Intel and VIA hardware used for these tests. The VIA tests required a non-production board with an engineering development BIOS and early versions of Nano X2 and QuadCore processors. The author expects VIA production systems to deliver higher SPEC CPU2000 scores, since not all VIA CPU performance features were enabled in the test system. As an example, CPU data pre-fetching was not enabled.

	Intel Atom System	VIA System
CPU	Intel Atom D525	VIA Nano X2 U4300 (1.2+GHz) VIA QuadCore L4700 (1.2+GHz)
Chipset	Intel NM10 Express	VIA VX900
Graphics	Integrated	Integrated
Motherboard	Intel D525MW	VIA VT5968
Memory System	4GB DDR3-1066 (800)	4GB DDR3-1066
Memory Components	Super Talent W1066SB2GS (Samsung 128x8)	Super Talent W1066UB2GS (Samsung 128x8)
Disk Drive	Samsung 500G HT502HJ	Samsung 500G HT502HJ

SPEC CPU2000 Configuration File (VIA GCC)

```
company_name    = Centaur Technology
hw_model        = VIA VT5968
hw_cpu          = VIA Nano X2 U4300 @ 1.2+ GHz
hw_cpu_mhz      = 1200
hw_disk         = SATA, 500GB
hw_fpu          = Integrated
hw_memory       = 1 x 4096MB DDR3 1066MHz
hw_avail        =
test_date       = 7/1/11
sw_file         = Linux/ext4
sw_os           = Fedora 14 64-bit
hw_vendor       = Centaur Technology
tester_name     = Jesse Ahrens
license_num     =

hw_ncpu         = 2
hw_ncpuorder    = 1
hw_ocache       = N/A
hw_other        = None
hw_parallel     = No
hw_pcache       = 64KB instruction + 64KB Data (per core)
hw_scache       = 1024KB (per core)
hw_tcache       = N/A
sw_state        = Multi-user text mode

action          = validate
tune            = base
output_format   = asc,html,config,pdf
ext             = gcc4.5.1

check_md5       = 1
reportable      = 1
```

```
teeout=yes
teerunout=yes

default=default=default=default:
CC           = gcc
CXX          = g++
FC           = gfortran
F77          = gfortran

BASE_FLAGS              = -march=core2 -mfpmath=sse -msse4.1 -funroll-loops -fno-strict-aliasing
PEAK_FLAGS              = -march=core2 -mfpmath=sse -msse4.1 -funroll-loops -fno-strict-aliasing

#####################################################################
# Portability Flags
#####################################################################

164.gzip=default=default=default:
CPORTABILITY            = -DSPEC_CPU2000_LP64

186.crafty=default=default=default:
notes0050      = 186.crafty: CPORTABILITY=-DLINUX_i386
CPORTABILITY   = -DLINUX_i386

252.eon=default=default=default:
notes0051      = 252.eon: CXXPORTABILITY=-DHAS_ERRLIST -fpermissive
CXXPORTABILITY  = -DHAS_ERRLIST -fpermissive

253.perlbmk=default=default=default:
notes0052      = 253.perlbmk: CPORTABILITY=-DSPEC_CPU2000_LINUX_I386 -DSPEC_CPU2000_NEED_BOOL
CPORTABILITY   = -DSPEC_CPU2000_NEED_BOOL -DSPEC_CPU2000_LINUX_I386 -DSPEC_CPU2000_LP64

254.gap=default=default=default:
notes0055      = 254.gap: CPORTABILITY=-DSYS_IS_USG -DSYS_HAS_IOCTL_PROTO -DSYS_HAS_TIME_PROTO
notes0056      =                -DSYS_HAS_CALLOC_PROTO -DSYS_HAS_MALLOC_PROTO
CPORTABILITY   = -DSYS_HAS_MALLOC_PROTO -DSYS_HAS_CALLOC_PROTO -DSYS_IS_USG -DSYS_HAS_IOCTL_PROTO -
DSYS_HAS_TIME_PROTO -DSPEC_CPU2000_LP64

255.vortex=default=default=default:
CPORTABILITY            = -DSPEC_CPU2000_LP64

300.twolf=default=default=default:
CPORTABILITY            = -DSPEC_CPU2000_LP64 -DHAVE_SIGNED_CHAR

178.galgel=default=default=default:
notes0050      = 178.galgel: -ffixed-form
FPORTABILITY   = -ffixed-form


#####################################################################
# Baseline Tuning Flags
#####################################################################

int=base=default=default:
COPTIMIZE  = -O3 $(BASE_FLAGS)
CXXOPTIMIZE             = -O3 $(BASE_FLAGS)

fp=base=default=default:
FOPTIMIZE  = -O3 $(BASE_FLAGS)
F77OPTIMIZE             = -O3 $(BASE_FLAGS)
COPTIMIZE  = -O3 $(BASE_FLAGS)

#####################################################################
# Peak Tuning Flags
#####################################################################

int=peak=default=default:
COPTIMIZE    = -O3 $(PEAK_FLAGS)
CXXOPTIMIZE  = -O3 $(PEAK_FLAGS)
PASS1_CFLAGS = -O3 $(PEAK_FLAGS) -fprofile-generate
PASS2_CFLAGS = -O3 $(PEAK_FLAGS) -fprofile-use
PASS1_CXXFLAGS = -O3 $(PEAK_FLAGS) -fprofile-generate
PASS2_CXXFLAGS = -O3 $(PEAK_FLAGS) -fprofile-use
PASS1_LDCFLAGS = -O3 $(PEAK_FLAGS) -fprofile-generate
PASS2_LDCFLAGS = -O3 $(PEAK_FLAGS) -fprofile-use
PASS1_LDCXXFLAGS = -O3 $(PEAK_FLAGS) -fprofile-generate
PASS2_LDCXXFLAGS = -O3 $(PEAK_FLAGS) -fprofile-use

fp=peak=default=default:
FOPTIMIZE   = -O3 $(PEAK_FLAGS)
F77OPTIMIZE = -O3 $(PEAK_FLAGS)
COPTIMIZE   = -O3 $(PEAK_FLAGS)


__MD5__
168.wupwise=base=gcc4.4.3=default:
# Last updated Tue Apr 12 12:55:30 2011
optmd5=c09313c148a35011828afc2e8d09a860
exemd5=24a164d22e9d826bbbaa7eb852f0dc70

171.swim=base=gcc4.4.3=default:
# Last updated Tue Apr 12 12:55:30 2011
optmd5=e3c8cae8e2ef2b033beac5cea2b0546d
exemd5=a9bf8baff292901feb01add6a5c4e5fb

172.mgrid=base=gcc4.4.3=default:
# Last updated Tue Apr 12 12:55:30 2011
optmd5=ba7a413645143639ad93b10e39ab55b4
exemd5=1fe0a644f62a7117a6e127df0538a214

173.applu=base=gcc4.4.3=default:
# Last updated Tue Apr 12 12:55:30 2011
optmd5=0b3df570d86618026ad6f8f95c68677c
```

exemd5=28d3c33a56dd648bcacf01275d3288e6

177.mesa=base=gcc4.4.3=default:
# Last updated Tue Apr 12 12:55:30 2011
optmd5=f6495517da7626c26750598c50555fab
exemd5=d6ee4cb1dddb18e88a202bdf30f42b45

178.galgel=base=gcc4.4.3=default:
# Last updated Tue Apr 12 12:55:30 2011
optmd5=4b3a17732b8dee9375054edd42abf407
exemd5=c09bda1919c3019768a983322fb676f5

179.art=base=gcc4.4.3=default:
# Last updated Tue Apr 12 12:55:30 2011
optmd5=f693e8b8887ac15ad0a4757c585e750e
exemd5=bae8828195d3e60bda6a17372c7afecf

183.equake=base=gcc4.4.3=default:
# Last updated Tue Apr 12 12:55:30 2011
optmd5=8b2a23c22b8e0557ed147ee4a92cbe53
exemd5=2dd124ff59f0f96bfe94082e236ec1d4

187.facerec=base=gcc4.4.3=default:
# Last updated Tue Apr 12 12:55:30 2011
optmd5=4e71dd73e4ec0d8bb4f535406eddd59c
exemd5=0f122903b424d0af82d57a188ffc6a72

188.ammp=base=gcc4.4.3=default:
# Last updated Tue Apr 12 12:55:30 2011
optmd5=ec6395a4dde4076ceb42f50a07e28333
exemd5=7453a7a4134ef01e2f4fb2f576417fe3

189.lucas=base=gcc4.4.3=default:
# Last updated Tue Apr 12 12:55:30 2011
optmd5=c7f4521952295242ef142e2f5e004d88
exemd5=93efd2464fddc84d3d7dc07d032bb466

191.fma3d=base=gcc4.4.3=default:
# Last updated Tue Apr 12 12:55:30 2011
optmd5=9894a3701fce909902a236a17bb9e8a5
exemd5=5bb837eddf197f47b083e833fff46cd5

200.sixtrack=base=gcc4.4.3=default:
# Last updated Tue Apr 12 12:55:30 2011
optmd5=9006ed7f8b00f50a0ee10141f4a367a6
exemd5=fec14cbb545d83ae745da54c027313b5

301.apsi=base=gcc4.4.3=default:
# Last updated Tue Apr 12 12:55:30 2011
optmd5=4cdf0319a476906ec972b66f073ebe3c
exemd5=99ca93d26ff64c2510733585b0051e03

164.gzip=base=gcc4.5.1=default:
# Last updated Wed Apr 20 15:19:13 2011
optmd5=13d4c4370cfd12d1838dbeb3d4fb7f7f
exemd5=2e7e523c7ec2f1d753c8ac0621f6ebff

175.vpr=base=gcc4.5.1=default:
# Last updated Wed Apr 20 15:37:18 2011
optmd5=7d00c4304b01f37be9ec75ffa6129a89
exemd5=b8507e6205444997dd9a428217d33716

176.gcc=base=gcc4.5.1=default:
# Last updated Wed Apr 20 15:37:18 2011
optmd5=d9dd885f40ab62df05d5080073c32b47
exemd5=b4d280ffdaf9c494eed41d73e7f7a847

181.mcf=base=gcc4.5.1=default:
# Last updated Wed Apr 20 15:37:18 2011
optmd5=ef016a3f9381d216abdec9e72c667079
exemd5=dc56ce82b3553c0ea3df0b7e2a4f703d

186.crafty=base=gcc4.5.1=default:
# Last updated Wed Apr 20 15:37:18 2011
optmd5=e4ca99c6354921767f0ffb60f45e8009
exemd5=6feb93e3bebb7a0d8339921b791d0606

197.parser=base=gcc4.5.1=default:
# Last updated Wed Apr 20 15:37:18 2011
optmd5=e2fbe70bd166a16bf4a1cc6a23479657
exemd5=ce15aba85ffa21234ed4b778ec74795f

252.eon=base=gcc4.5.1=default:
# Last updated Wed Apr 20 15:37:18 2011
optmd5=c8cc8e13ce0710e91198b28bd8732ee6
exemd5=2d59a8c4d6679e37324caa6bb704af87

253.perlbmk=base=gcc4.5.1=default:
# Last updated Wed Apr 20 15:37:18 2011
optmd5=53950dc54e5c1bccc46c11486534e5da
exemd5=8b2e673748da7b1f5cf0db4068bc135c

254.gap=base=gcc4.5.1=default:
# Last updated Wed Apr 20 15:37:18 2011
optmd5=ba309de897cad20f0e9fc0b6cef1ef0a
exemd5=a31a23c6d132913bc48f7c01f5e3ff0a

255.vortex=base=gcc4.5.1=default:
# Last updated Wed Apr 20 15:37:18 2011
optmd5=4f43bd25d3fb9417e1596ca94cb29ec7
exemd5=2c5d6fbca2fe43304ccbdacdf5233466

256.bzip2=base=gcc4.5.1=default:
# Last updated Wed Apr 20 15:37:18 2011
optmd5=b718f8a69eedeac674862e84d0106231
exemd5=f5b2134a6d2b3f5d021338ec1e7c1fcc

300.twolf=base=gcc4.5.1=default:
# Last updated Wed Apr 20 15:37:18 2011
optmd5=d0239bd6a4398a0cb948274807eb4404
exemd5=03e233db7455a77435b39d742780fd02

168.wupwise=base=gcc4.5.1=default:
# Last updated Wed Apr 20 17:31:48 2011
optmd5=100958f2cab2024a9229a5c6f356eebf
exemd5=572f22ba457acb30106977cca5439d7e

171.swim=base=gcc4.5.1=default:
# Last updated Wed Apr 20 17:31:48 2011
optmd5=035e7f5779ebf4537671a04f68383555
exemd5=ad391ee8feb3955729b45c1498771757

172.mgrid=base=gcc4.5.1=default:
# Last updated Wed Apr 20 17:31:48 2011
optmd5=e4d40cabd48ff4eb51b499014d641dcd
exemd5=a875566a2ccc3676a6b213c3288d3785

173.applu=base=gcc4.5.1=default:
# Last updated Wed Apr 20 17:31:48 2011
optmd5=1967d8c35157f719bf6aff0af5734070
exemd5=d996e7887246673d7fb9cb7ef4c5002e

177.mesa=base=gcc4.5.1=default:
# Last updated Wed Apr 20 17:31:48 2011
optmd5=483fc1ef91979bcf5bc9216478ae2193
exemd5=d88fb5837718ae68965d439e63925b59

178.galgel=base=gcc4.5.1=default:
# Last updated Wed Apr 20 17:31:48 2011
optmd5=1eb25569214eb8aed082cd71c05e186e
exemd5=93bdcedc5bd1af2be2206bbccc1c5415

179.art=base=gcc4.5.1=default:
# Last updated Wed Apr 20 17:31:48 2011
optmd5=a8839c229d5724cd6000de62a404443d
exemd5=e6a8c51caa24af4a0ea7a88144469c60

183.equake=base=gcc4.5.1=default:
# Last updated Wed Apr 20 17:31:48 2011
optmd5=ea30f5cd3cdeaf180e0fe6f7b4fc622f
exemd5=5c213ada1e32c958d1e74239a51a05a0

187.facerec=base=gcc4.5.1=default:
# Last updated Wed Apr 20 17:31:48 2011
optmd5=32047afec425a88f913b663bc593add8
exemd5=f0e53ee2e6a21aef1489aaccda142e27

188.ammp=base=gcc4.5.1=default:
# Last updated Wed Apr 20 17:31:48 2011
optmd5=3605ee5b625b30dd7a9144a394cf13da
exemd5=2108c7c5c874b44310a57f578d981f8d

189.lucas=base=gcc4.5.1=default:
# Last updated Wed Apr 20 17:31:48 2011
optmd5=8274227aac47863f86de6066f7be1b2b
exemd5=bb055f28b8baaf30d8e00ca8a8832558

191.fma3d=base=gcc4.5.1=default:
# Last updated Wed Apr 20 17:31:48 2011
optmd5=31b2d93901e76499693c6bf307feef39
exemd5=98c81ed0c4dd982edd373eb7bc8aed82

200.sixtrack=base=gcc4.5.1=default:
# Last updated Wed Apr 20 17:31:48 2011
optmd5=5ebd60f970132bd7e529b270f11998d8
exemd5=42fc57a72151ed4acd66d6d0503b1bc5

301.apsi=base=gcc4.5.1=default:
# Last updated Wed Apr 20 17:31:48 2011
optmd5=1c46e15a7273ce74a12730e8b61c56dd
exemd5=4ed040f53fdc403a138f9a85c9006938

164.gzip=peak=gcc4.5.1=default:
# Last updated Thu Apr 28 19:44:20 2011
optmd5=89d2a984e5c878bf4eacc81d23fda58a
exemd5=70473ee908a1405d7f87eaf35458afec

175.vpr=peak=gcc4.5.1=default:
# Last updated Thu Apr 28 19:44:20 2011
optmd5=91d8bb69da7d3f158d0e2bcd497ac8f2
exemd5=d2eab959bd71e95e1e6f0e3127419ba1

176.gcc=peak=gcc4.5.1=default:
# Last updated Thu Apr 28 19:44:20 2011
optmd5=464f3f17a01b40c812a012b0380561e5
exemd5=9cf6a4f270ffc30fb839ad112e3a284e

181.mcf=peak=gcc4.5.1=default:
# Last updated Thu Apr 28 19:44:20 2011
optmd5=9d170670f142168736a4a30948e14147
exemd5=1be63730cec380a526e483a13fe3f4c0

186.crafty=peak=gcc4.5.1=default:
# Last updated Thu Apr 28 19:44:20 2011
optmd5=b0a3fd269e937cbd6f4f850f4457e0dc
exemd5=51dc0ca5bd9d0920262e1ebf2bbc5cf9

197.parser=peak=gcc4.5.1=default:
# Last updated Thu Apr 28 19:44:20 2011
optmd5=49768880079f3d57ca0a2fcd541f2ae7
exemd5=bdea2438b8ccec7deef1edb78ae46001

252.eon=peak=gcc4.5.1=default:
# Last updated Thu Apr 28 19:44:20 2011
optmd5=5771c3b662a4fff2362276bbb8a049ef
exemd5=254d010abc9c8251a875e02419fe9964

253.perlbmk=peak=gcc4.5.1=default:
# Last updated Thu Apr 28 19:44:20 2011
optmd5=7a5fa90d68d8425506fea68737cf6634
exemd5=dc5c75b54d9441315063ab743875119c

254.gap=peak=gcc4.5.1=default:
# Last updated Thu Apr 28 19:44:20 2011
optmd5=ed76b41e3265bb6ff35f0df1be411c5f
exemd5=6b4688fdd64918b565a1f6add39acf59

255.vortex=peak=gcc4.5.1=default:
# Last updated Thu Apr 28 19:44:20 2011
optmd5=99ca1a465967384e860ff3c8d79e0ffe
exemd5=003a7ef4f92939f1c525b44c2e512418

256.bzip2=peak=gcc4.5.1=default:
# Last updated Thu Apr 28 19:44:20 2011
optmd5=612bb2f3125ab812c7ddd1df43cf989d
exemd5=8c8e94559f89c58c036dec067a6edcbe

300.twolf=peak=gcc4.5.1=default:
# Last updated Thu Apr 28 19:44:20 2011
optmd5=d87e66517e271981663affe72922b11a
exemd5=4eeb0ec6d5c1c4c452939f5bdf84d7c7

168.wupwise=peak=gcc4.5.1=default:
# Last updated Fri Apr 29 10:02:18 2011
optmd5=4f75d7d2a25730fb82c00f9b597f7c27
exemd5=572f22ba457acb30106977cca5439d7e

171.swim=peak=gcc4.5.1=default:
# Last updated Fri Apr 29 10:02:18 2011
optmd5=42ad92fc4b2dd719841e522900d8052d
exemd5=ad391ee8feb3955729b45c1498771757

172.mgrid=peak=gcc4.5.1=default:
# Last updated Fri Apr 29 10:02:18 2011
optmd5=8faa02d9b47a7b5e60584c1bb49ae53a
exemd5=a875566a2ccc3676a6b213c3288d3785

173.applu=peak=gcc4.5.1=default:
# Last updated Fri Apr 29 10:02:18 2011
optmd5=5b416c50902ece94f6427cac9fef70f6
exemd5=d996e7887246673d7fb9cb7ef4c5002e

177.mesa=peak=gcc4.5.1=default:
# Last updated Fri Apr 29 10:02:19 2011
optmd5=bc6599561f916e30fd0d2837c0901cf0
exemd5=d88fb5837718ae68965d439e63925b59

178.galgel=peak=gcc4.5.1=default:
# Last updated Fri Apr 29 10:02:19 2011
optmd5=bab3b97ed5178c59efdbea73c86707cd
exemd5=93bdcedc5bd1af2be2206bbccc1c5415

179.art=peak=gcc4.5.1=default:
# Last updated Fri Apr 29 10:02:19 2011
optmd5=b2d1e19193845eec2ccc6cdf20041cc7
exemd5=e6a8c51caa24af4a0ea7a88144469c60

183.equake=peak=gcc4.5.1=default:
# Last updated Fri Apr 29 10:02:19 2011
optmd5=80f621135b599fa4f6c28616ec3c717a
exemd5=5c213ada1e32c958d1e74239a51a05a0

187.facerec=peak=gcc4.5.1=default:
# Last updated Fri Apr 29 10:02:19 2011
optmd5=c3d3a40e6511d847b9a8fdcc61bab300
exemd5=f0e53ee2e6a21aef1489aaccda142e27

188.ammp=peak=gcc4.5.1=default:
# Last updated Fri Apr 29 10:02:19 2011
optmd5=676dacf020b94ce6b4f6db31f632d80c
exemd5=2108c7c5c874b44310a57f578d981f8d

189.lucas=peak=gcc4.5.1=default:
# Last updated Fri Apr 29 10:02:19 2011
optmd5=808ec57186189aaeb8c7c899afba3da2
exemd5=bb055f28b8baaf30d8e00ca8a8832558

191.fma3d=peak=gcc4.5.1=default:
# Last updated Fri Apr 29 10:02:19 2011
optmd5=b14104f7f2306c97ce9657d8cf6987e3
exemd5=98c81ed0c4dd982edd373eb7bc8aed82

200.sixtrack=peak=gcc4.5.1=default:
# Last updated Fri Apr 29 10:02:19 2011
optmd5=91a057db231fdbf92eca0e8d1370845c
exemd5=42fc57a72151ed4acd66d6d0503b1bc5

301.apsi=peak=gcc4.5.1=default:
# Last updated Fri Apr 29 10:02:19 2011
optmd5=29d86eb387b7057128d3fca6439cf36d
exemd5=4ed040f53fdc403a138f9a85c9006938