Cache Performance of the SPEC Benchmark Suite*, **

Jeffrey D. Gee† *Mark D. Hill*‡
Dionisios N. Pnevmatikatos‡ *Alan Jay Smith*†

Computer Science Division†
Dept. of Electrical Engineering
and Computer Science
University of California Berkeley
Berkeley, CA 94720

Computer Sciences Department‡
University of Wisconsin-Madison
Madison, WI 53706

Abstract

The SPEC benchmark suite consists of ten public-domain, non-trivial programs that are widely used to measure the performance of computer systems, particularly those in the Unix workstation market. These benchmarks were expressly chosen to represent real-world applications and were intended to be large enough to stress the computational and memory system resources of current-generation machines. The extent to which the *SPECmark* (the figure of merit obtained from running the SPEC benchmarks under certain specified conditions) accurately represents performance with live real workloads is not well established; in particular, there is some question whether the memory referencing behavior (cache performance) is appropriate.

In this paper, we present measurements of miss ratios for the entire set of SPEC benchmarks for a variety of CPU cache configurations; this study extends earlier work that measured only the performance of the integer (C) SPEC benchmarks. We find that instruction cache miss ratios are generally very low, and that data cache miss ratios for the integer benchmarks are also quite low. Data cache miss ratios for the floating point benchmarks are more in line with published measurements for real workloads. We believe that the discrepancy between the SPEC benchmark miss ratios and those observed elsewhere is partially due to the fact that the SPEC benchmarks are all almost exclusively user state CPU benchmarks run until completion as the single active user process. We therefore believe that SPECmark performance levels may not reflect system performance when there is multiprogramming, time sharing and/or significant operating systems activity.

---

*The material presented here is based on research supported in part by the National Science Foundation under grants MIP-8713274 and MIPS-8957278, by NASA under grant NCC2-550, by the State of California under the MICRO program, and by A.T.& T. Bell Laboratories, Apple Computer Corporation, Cray Research Foundation, Digital Equipment Corporation, Intel Corporation, International Business Machines Corporation, Mitsubishi Corporation, and Philips Laboratories/Signetics.

**This paper has been submitted for publication and released as *University of California at Berkeley Computer Science Division Technical Report UCB/CSD 91/648* and *University of Wisconsin at Madison Computer Sciences Department Technical Report #1049*.

1. Introduction

The SPEC benchmarks [SPEC89,90, Scot90] are a selection of non-trivial programs chosen to standardize benchmarking. SPEC (System Performance Evaluation Consortium) assembled this suite to provide a standard set of realistic benchmarks for inter-system comparisons; see [Pric89, Hinn88] for a discussion of the many problems with the benchmarking situation prior to SPEC. Several factors, including strong industrial support for SPEC, the realistic nature of the benchmarks, and acceptable code portability have led to the wide use of these programs for benchmarking purposes. To improve verification and reproducibility of results, SPEC benchmark results must include a description of any source code modifications, compiler and operating system release numbers, machine characteristics, and most other factors that can affect the reported results. The SPEC benchmarks have become so important as a measure of CPU performance that some system developers are parameterizing their designs to maximize SPEC benchmark performance, even when this might lead to lower performance on other, perhaps more realistic, workloads. Similarly, compiler writers have been concentrating on producing good code for the frequently executed inner loops of some of the SPEC benchmarks, such as *matrix300*. Recent very high SPEC benchmark results for *matrix300* show the success of their efforts.

SPEC Benchmark Suite		
Program	Language	Description
doduc	Fortran	Thermohydraulic simulation of a nuclear reactor
eqntott	C	Builds truth table from a boolean expression
espresso	C	Boolean function minimization
fpppp	Fortran	Two electron integral derivative
gcc	C	GNU C compiler compiling pre-processed source files
matrix300	Fortran	Linpack SAXPY routine on 300x300 matrix
nasa7	Fortran	Seven floating-point synthetic kernels
spice	Fortran	Analog circuit simulator
tomcatv	Fortran	Mesh generation program
xlisp	C	Lisp interpreter solving eight queens problem

Table 1: SPEC Benchmark Applications

Release 1.0 of the SPEC benchmarks consists of four integer-intensive C programs (*eqntott*, *espresso*, *gcc*, and *xlisp*) and six floating-point intensive Fortran programs (*doduc*, *fpppp*, *matrix300*, *nasa7*, *spice*, and *tomcatv*). The SPEC benchmarking procedure is to run each program to completion on the target system, with only one user process active, and then take the ratio of that run time to the run time of the same program on a DEC VAX 11/780. The geometric mean of those ratios, averaged over the SPEC benchmark suite, yields the *"SPECmark"*, which is a single number figure of merit. Table 1 lists and gives a short description of each benchmark.

As noted above, considerable effort is being expended on creating computer systems (hardware and software) to optimize SPEC benchmark results. Two questions therefore arise: (a) In what ways should the system be designed to perform well on the SPEC benchmark suite? (b) Is this a good idea?

One important aspect of CPU performance, and probably the most important of the architectural aspects (as opposed to technology parameters, such as circuit speed) is the performance of the memory hierarchy. We note that SPEC benchmark results are quite sensitive to cache size, as may be seen by comparing the various published measurements of systems with varying caches sizes based on the Motorola 88000. In terms of the SPEC benchmarks, the two questions above become: (a) What miss ratios can be expected when running the SPEC benchmarks on a cache of a given design? (b) Are these miss ratios comparable to those for "typical" user workloads, for some definition of typical?

In this paper we present measurements of the cache miss ratios of the entire SPEC benchmark suite and comment on their potential use in the design of caches and memory hierarchies. We compare the SPEC cache miss ratios to design target miss ratios [Smit87], miss ratios measured using hardware monitors at Amdahl [Smit82] and on DEC VAX-series machines [Clar83,88], miss ratios observed from very long address traces [Borg90], and other miss ratios that include operating system and multiprogramming behavior. We also note that miss ratios for multiprogrammed workloads with significant operating system activity are known to be high [Agar88,Ande91].

2. SPEC Cache Performance

2.1. Methodology

We compiled and ran the SPEC programs on DECstation 3100's (which contain the MIPS R2000 microprocessor), running version 4.1 of the DEC Ultrix operating system. We used version 2.0 of the C compiler and version 2.1 of the Fortran compiler with optimization level -O3 for espresso, doduc, nasa7 xlisp, eqntott, matrix300, fpppp, tomcatv and -O2 for gcc, spice as per the SPEC Makefiles. Note that we have not used any of the recently developed preprocessors which have been very successful in reorganizing the data references for Matrix300. We then used the MIPS *pixie* [DEC91] tool to generate address traces to feed directly to the *tycho* [Hill] cache simulator. Pixie modifies the compiled code to generate a trace record for each load, store and basic block entry; trace records for all instruction fetches are then constructed from the basic block records. Tycho uses algorithms that, for a given block size, simulate all cache sizes and associativities in a single pass through an address trace [Hill87].

We varied cache size from 1 Kbyte to 1 Mbyte, set size from one (direct-mapped) to eight, and block size from 16 to 256 bytes. All caches used the LRU replacement algorithm and the lowest order available address bits to select the set. We simulated instruction, data, and unified caches, without any periodic cache flushing, as the SPEC benchmarks are typically run in a uniprogrammed environment. Miss ratios represent the complete execution of a benchmark and include start-up as well as steady-state effects. The

use of *pixie* to generate address traces allows simulation of only user, and not system references, and our data is for user code only. Table 2 lists the number of instruction, data, and total user memory references made by each program. Note that the trace reflects a 4-byte memory interface; the trace would be different for a different memory interface width. Note also that the trace includes only actual program loads, stores and instruction fetches; it does not include the extra memory activity such as instruction prefetch that would occur on most machines [Clar83].

Program	Instruction	Data	Total
eqntott	1,241,913,236	215,772,134	1,457,685,370
espresso	3,143,686,831	696,870,530	3,840,557,361
gcc	1,262,492,069	398,952,157	1,661,444,226
xlisp	1,234,252,567	457,209,682	1,691,462,249
doduc	1,619,374,300	583,667,566	2,203,041,866
fpppp	2,396,679,406	1,514,694,293	3,911,373,699
matrix300	2,766,534,109	1,311,922,365	4,078,456,474
nasa7	9,195,719,149	4,720,515,938	13,916,235,087
spice	28,696,843,509	8,288,246,353	36,985,089,862
tomcatv	1,872,460,468	913,221,318	2,785,681,786

Table 2: Program Reference Counts

To increase our confidence in our results we compared them with two other studies that ran the SPEC benchmarks on a MIPS R2000 microprocessor. Pnevmatikatos and Hill [Pnev90] presented cache miss ratios for the four integer (C language) SPEC benchmarks. They used a different compiler (gcc) and a tracing methodology that excludes library references. Nevertheless, most miss ratio differences are less than 0.01. In few cases, however, a seemingly small miss ratio difference translates into a substantial relative change. We are inclined to place the most confidence in the results presented here, since this analysis has used much more mature and sophisticated compilers, but the comparison demonstrates that cache miss ratios, instruction counts, and related measures are, as might be expected, sensitive to the compiler used. We must thus caution readers that *your actual mileage may vary*. Cmelik et al. [Cmel91] give instruction counts for SPEC benchmarks. With two exceptions, Xlisp and Spice, their counts are close to ours. For Xlisp, the instruction count difference is due to different input files; we solved the eight queens problem, while they solved the nine queens problem. For Spice, however, we cannot explain the difference, although simulation runs at both Berkeley and Madison yielded consistent results.

Simulating these caches required 400 to 800 microseconds of CPU time per memory reference in each trace. Assuming an average 600 microseconds per memory reference, simulating all ten SPEC benchmarks requires some 500 days or *nearly 17 months* of CPU time. Including false starts, simulation errors, and operating system

bugs, we used *two to three years of machine time* to compute our results; this type of measurement would not have been possible if it had been necessary to pay for CPU time on a timeshared machine. With five machines available for running simulations at Berkeley and Madison, we were able to generate these results in less than six months of calendar time.

2.2. Results

The appendix of this paper displays the miss ratios for each SPEC program, for each block (line) size from 16 to 256 bytes, each cache size from 4Kbytes to 1Mbyte, for set-associativity of 1 (direct mapping), 2, 4 and 8, and for instruction, data and unified caches.† In this section, we comment on some of that data.

We first examine instruction cache miss ratios for the different programs. For *eqntott*, *matrix300*, *nasa7*, and *tomcatv*, instruction cache miss ratios are very low, generally less than 0.0001 for caches as small as a few kilobytes. These programs spend much of their execution time in a few small routines; *matrix300*, for example, spends about 99% of its execution time in one small basic block in the code [Saav90,Saav91]. Miss ratios for *espresso*, *xlisp*, and *spice* are only slightly larger, as miss ratios again fall below 0.0001 for cache sizes as small as 16 or 32 Kbytes. Instruction cache miss ratios are largest for *doduc*, *gcc*, and *fpppp*, yet are well below half a percent for caches as small as 64 or 128 Kbytes. None of the SPEC benchmarks makes significant use of more than 128 Kbytes of instruction cache.

Miss ratios for data caches are larger, especially for several of the floating-point Fortran benchmarks, but for the most part are quite low as cache size approaches one megabyte. Miss ratios for *xlisp* and *doduc* are the lowest among the SPEC suite, dropping below one percent for caches as small as 16 or 32 Kbytes, and falling below 0.0001 for a 64 Kbyte cache. Results for *fpppp* and *espresso* are nearly as low as results for *xlisp* and *doduc* when set size is greater than one, and somewhat larger for direct-mapped caches. Among the integer programs, *eqntott* and *gcc* exercise fairly large data caches; miss ratios remain above one percent until cache size reaches 512 Kbytes.

The floating-point programs *matrix300*, *nasa7*, *spice*, and *tomcatv* exhibit the largest data cache miss ratios. Miss ratios for *matrix300*, *nasa7*, and *spice* are several percent until the cache size reaches one megabyte, causing miss rates to fall below one percent. *Tomcatv* requires extremely large caches when the cache block size is small. Data cache miss ratios are *over 6 percent* for a 1 Mbyte cache at a 16-byte block size. Each successive doubling of block size in *tomcatv* at 1Mbyte reduces data cache miss ratios by almost half, and miss ratios do become less than one percent for a 128 byte block size.

Unified (data and instruction) cache miss ratios usually fall between instruction and data cache miss ratios, as the strong locality in instruction references offsets the weaker locality in data references. We do observe several instances where unified cache miss rates are *higher* than corresponding data cache miss rates (*espresso, xlisp, doduc, fpppp*).

† See the appendix for a description of how to obtain an electronic copy of these results.

This behavior occurs mainly at larger cache sizes coupled with low associativities, and where separate instruction and data cache miss ratios have fallen to nearly zero. The low associativity causes instruction and data references to conflict for cache sets, while such conflicts do not occur in separate instruction and data caches. Note that a split direct-mapped instruction/data cache pair is more like a 2-way set-associative unified cache than a direct-mapped unified cache.

It is worth noting that there are a few anomalies in the data with respect to the effect of associativity on miss ratio. Generally, miss ratios decrease with increased degrees of set associativity, since the probability of mapping conflicts decreases [Hill89]. It is possible, however, that miss ratios can increase with increasing associativity if certain reference patterns are present in the memory reference string; we note just that effect at one or more data points for the fpppp, matrix300, spice, tomcatv, and doduc miss ratios. These anomalies are disturbing in that they suggest, incorrectly, that on the the average miss ratios will not decrease with increasing associativity.

3. Evaluation

In this section we compare the SPEC miss ratios with miss ratios from previous studies and discuss whether the SPEC applications make suitable cache benchmarks. We first describe the other studies.

(a) Smith [Smit82] includes several measurements taken with a hardware monitor at Amdahl Corporation on various models of the Amdahl 470V machines. Results showed that supervisor state miss ratios were much higher than problem state miss ratios, and that the miss ratio for each of user and supervisor state could be approximated by equations of the form $m = a * k^b$, where a and b are constants and k is the cache size in kilobytes.

(b) Two studies [Clar83,88] provide cache miss ratios taken via hardware measurement from VAX 11/780 and VAX 8800 computers. The 11/780 has an 8 Kbyte, write-through unified cache with an 8-byte block size and a set size of two. The 8800 has a 64 Kbyte, write-through, direct-mapped unified cache with a 64-byte block size.

(c) Smith [Smit85] introduced the *design target miss ratios (DTMRs)* to represent typical levels of performance, averaged over a wide class of workloads, ranging from workstations to timeshared mainframes. He synthesized them from real (hardware monitor) measurements that existed in the literature and a large number of trace-driven simulation results. The initial DTMRs for 16-byte line size, fully-associative caches [Smit85] were later extended to other line sizes [Smit87] and to set-associative caches [Hill87,89].

(d) Agarwal, et al. [Agar88] presented miss ratios that include the effects of operating system references and multiprogramming by using microcode to capture address traces from multi-tasked machines. These effects can more than double miss rates from those measured in a uniprogrammed, user-only environment.

(e) Borg, et al. [Borg90] generated miss ratios for very long address traces using tools similar to our own; those traces were over twelve billion memory references long. The traces were used to evaluate the performance of a variety of caches.

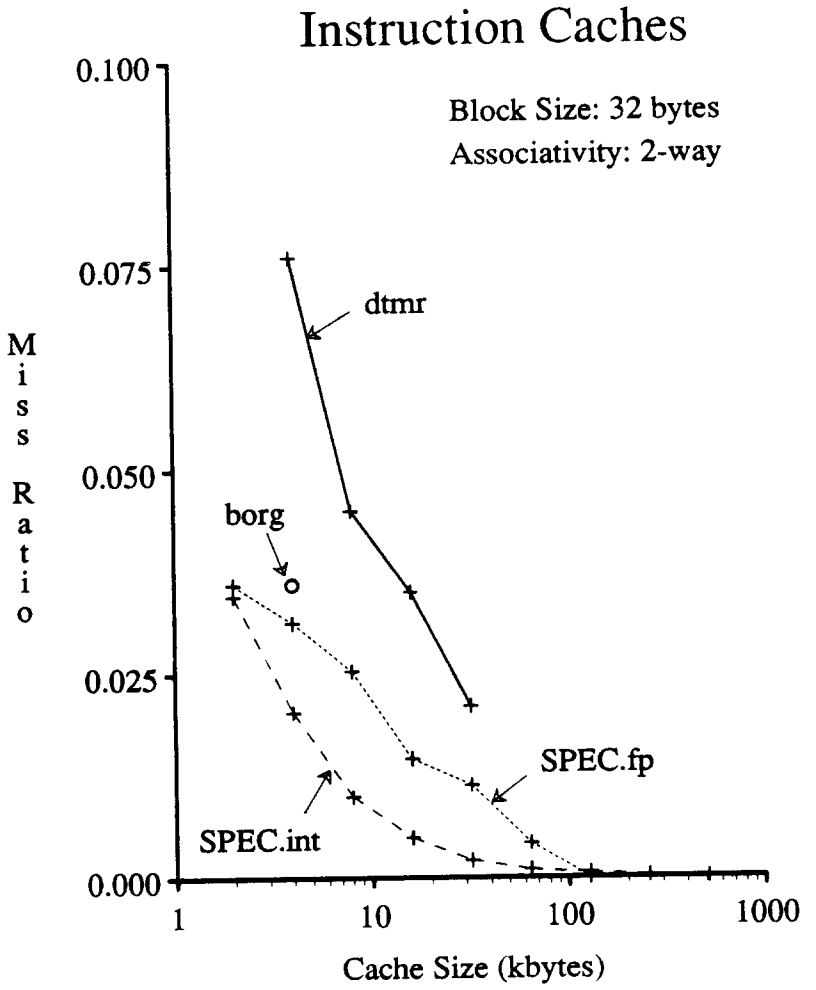

Figure 1: Instruction cache miss ratios

Figures 1 through 3 show average SPEC miss ratios for instruction, data, and unified caches, with 32-byte lines and 2-way set-associativity, computed separately for the integer and floating-point benchmarks. We also list in Tables 3 through 5 average miss ratios for the integer, floating-point, and complete SPEC suite across the entire range of simulation parameters. These averages represent the unweighted arithmetic mean of individual program miss ratios, similar to how the SPECmark represents the geometric mean of individual program SPECratios. In Figures 1 and 2, averages miss rates are plotted against the design target miss ratios (labeled *dtmr*) and primary cache miss ratios from [Borg90] for a multiprogrammed workload (labeled *borg*). Unfortunately, miss ratios from the other studies are not available for separate instruction and data caches, but are plotted against SPEC unified cache results in Figure 3. Previous results based on

different block sizes (VAX 11/780, VAX 8800, Agarwal, et al.) or different associativities (VAX 8800, Borg et al.) are adjusted for these parameters using ratios of miss ratios from prior studies [Hill89, Smit87].

A look at Figure 1 suggests that instruction cache miss ratios for the SPEC benchmarks are highly optimistic, as they are as low as one-third of the design target miss ratios and one-half of Borg's miss ratios.

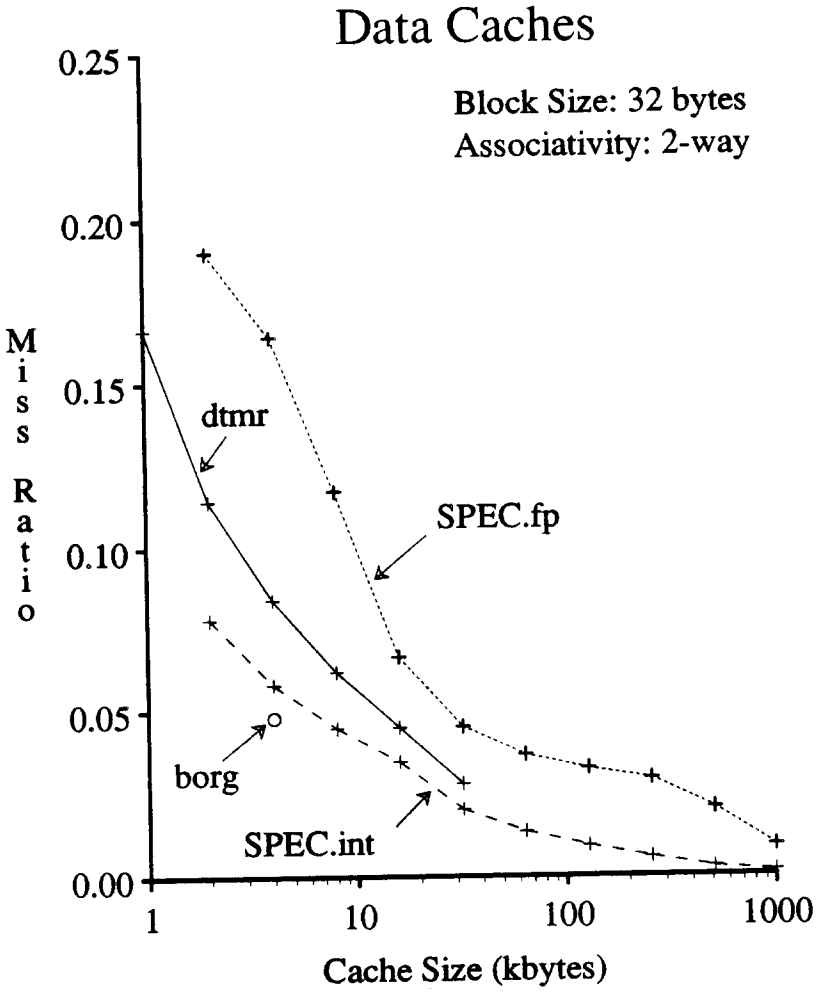

Figure 2: Data cache miss ratios

In Figure 2, we see that data cache miss ratios for the SPEC integer and floating point benchmarks bracket the DTMRs; both are above the [Borg90] measurements. Both sets of SPEC benchmarks approach zero miss ratio for large caches, a phenomenon that might not occur for a time sharing workload where programs task switch many times before completing.

Figure 3 contains unified cache measurements from the various other studies in addition to SPEC and design target miss ratios. These include: Amdahl 470 supervisor and user state miss ratios (plots labeled *470.sup* and *470.user*), VAX 11/780 and VAX 8800 miss ratios (plots labeled *VAX.780* and *VAX.8800*), and miss ratios from [Agar88] for a multiprogramming level of 3 (plots labeled *agarwal.mul3*). (We plot the Amdahl data from the fitted curve in [Smit82]; the original data points are not available.) We note that the VAX8800 data was collected from a very heavily used timeshared system. The Amdahl 470 supervisor data was collected from the execution of a standard internal Amdahl commercial workload. For both the VAX8800 and Amdahl data, the level of supervisor activity was quite high. Following in decreasing order of miss ratio are the DTMRs, the SPEC floating point miss ratios, and Agarwal's multiprogrammed miss ratios. VAX 11/780 and Amdahl 470 user state miss ratios follow, and the SPEC integer miss rates are smallest by a wide margin.

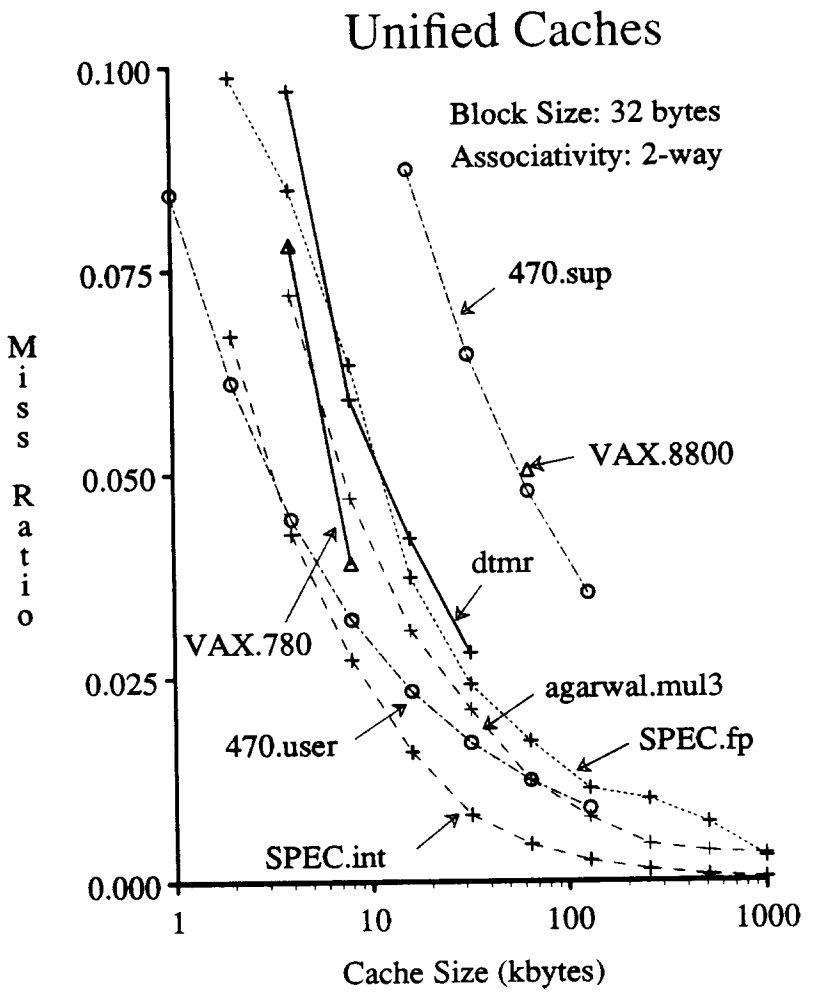

Figure 3: Unified cache miss ratios

Table 3: Average miss ratios for the integer SPEC benchmarks.

Integer Averages : Associativity 1

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.0882	0.0581	0.0394	0.0288	0.0229
2K	0.0599	0.0394	0.0269	0.0192	0.0146
4K	0.0429	0.0289	0.0197	0.0145	0.0110
8K	0.0229	0.0155	0.0111	0.0082	0.0062
16K	0.0119	0.0079	0.0058	0.0044	0.0034
32K	0.0071	0.0047	0.0033	0.0025	0.0019
64K	0.0031	0.0020	0.0014	0.0010	0.0008
128K	0.0015	0.0009	0.0006	0.0005	0.0004
256K	0.0005	0.0003	0.0002	0.0002	0.0001
512K	0.0002	0.0001	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.1663	0.1554	0.1826	0.2552	0.2774
2K	0.1247	0.1107	0.1239	0.1650	0.1829
4K	0.0948	0.0797	0.0840	0.1046	0.1185
8K	0.0725	0.0583	0.0582	0.0658	0.0727
16K	0.0564	0.0430	0.0416	0.0428	0.0464
32K	0.0372	0.0255	0.0204	0.0182	0.0205
64K	0.0266	0.0173	0.0134	0.0108	0.0120
128K	0.0174	0.0111	0.0086	0.0064	0.0072
256K	0.0109	0.0068	0.0053	0.0040	0.0044
512K	0.0063	0.0039	0.0030	0.0024	0.0025
1M	0.0032	0.0019	0.0013	0.0010	0.0009

Unified

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.1508	0.1263	0.1248	0.1616	0.1833
2K	0.1069	0.0879	0.0840	0.1019	0.1159
4K	0.0792	0.0638	0.0593	0.0673	0.0735
8K	0.0492	0.0389	0.0370	0.0397	0.0423
16K	0.0305	0.0232	0.0220	0.0230	0.0247
32K	0.0199	0.0144	0.0125	0.0125	0.0135
64K	0.0118	0.0085	0.0075	0.0073	0.0081
128K	0.0065	0.0045	0.0037	0.0035	0.0037
256K	0.0043	0.0030	0.0025	0.0024	0.0026
512K	0.0030	0.0021	0.0018	0.0017	0.0019
1M	0.0020	0.0014	0.0012	0.0013	0.0015

Integer Averages : Associativity 2

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.0517	0.0346	0.0232	0.0176	0.0127
4K	0.0294	0.0203	0.0145	0.0109	0.0084
8K	0.0144	0.0100	0.0074	0.0057	0.0045
16K	0.0069	0.0049	0.0036	0.0029	0.0024
32K	0.0031	0.0021	0.0015	0.0012	0.0010
64K	0.0016	0.0010	0.0007	0.0005	0.0004
128K	0.0009	0.0006	0.0004	0.0003	0.0002
256K	0.0003	0.0002	0.0001	0.0001	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.0956	0.0782	0.0778	0.0985	0.1149
4K	0.0756	0.0580	0.0521	0.0569	0.0683
8K	0.0603	0.0448	0.0379	0.0351	0.0402
16K	0.0473	0.0346	0.0284	0.0240	0.0265
32K	0.0310	0.0203	0.0148	0.0102	0.0113
64K	0.0220	0.0136	0.0100	0.0065	0.0074
128K	0.0149	0.0093	0.0071	0.0046	0.0054
256K	0.0091	0.0057	0.0046	0.0032	0.0036
512K	0.0044	0.0026	0.0020	0.0016	0.0016
1M	0.0023	0.0012	0.0007	0.0006	0.0004

Unified

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.0897	0.0669	0.0573	0.0601	0.0610
4K	0.0567	0.0426	0.0363	0.0360	0.0357
8K	0.0360	0.0272	0.0231	0.0217	0.0218
16K	0.0209	0.0158	0.0133	0.0120	0.0120
32K	0.0115	0.0081	0.0063	0.0052	0.0052
64K	0.0068	0.0045	0.0034	0.0026	0.0026
128K	0.0040	0.0025	0.0019	0.0014	0.0013
256K	0.0024	0.0014	0.0011	0.0008	0.0008
512K	0.0012	0.0007	0.0005	0.0004	0.0003
1M	0.0007	0.0004	0.0002	0.0002	0.0001

Integer Averages : Associativity 4

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0263	0.0177	0.0126	0.0099	0.0077
8K	0.0121	0.0087	0.0067	0.0052	0.0042
16K	0.0044	0.0032	0.0026	0.0022	0.0020
32K	0.0021	0.0014	0.0011	0.0008	0.0007
64K	0.0013	0.0008	0.0006	0.0004	0.0003
128K	0.0008	0.0005	0.0003	0.0002	0.0001
256K	0.0003	0.0002	0.0001	0.0001	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0628	0.0431	0.0341	0.0350	0.0415
8K	0.0500	0.0328	0.0241	0.0187	0.0221
16K	0.0382	0.0245	0.0174	0.0118	0.0132
32K	0.0300	0.0188	0.0132	0.0084	0.0092
64K	0.0203	0.0127	0.0094	0.0059	0.0068
128K	0.0145	0.0091	0.0069	0.0043	0.0052
256K	0.0078	0.0050	0.0042	0.0030	0.0035
512K	0.0035	0.0020	0.0016	0.0014	0.0013
1M	0.0022	0.0011	0.0006	0.0004	0.0003

Unified

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0492	0.0364	0.0298	0.0283	0.0274
8K	0.0286	0.0209	0.0167	0.0148	0.0150
16K	0.0154	0.0111	0.0089	0.0076	0.0079
32K	0.0092	0.0061	0.0046	0.0037	0.0039
64K	0.0054	0.0035	0.0026	0.0019	0.0019
128K	0.0036	0.0022	0.0016	0.0011	0.0011
256K	0.0020	0.0013	0.0010	0.0007	0.0007
512K	0.0010	0.0006	0.0004	0.0003	0.0003
1M	0.0006	0.0003	0.0002	0.0001	0

Integer Averages : Associativity 8

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0110	0.0082	0.0066	0.0050	0.0041
16K	0.0037	0.0026	0.0021	0.0021	0.0020
32K	0.0019	0.0013	0.0009	0.0007	0.0006
64K	0.0012	0.0007	0.0005	0.0003	0.0003
128K	0.0008	0.0005	0.0003	0.0002	0.0001
256K	0.0001	0.0001	0.0001	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0492	0.0320	0.0232	0.0173	0.0201
16K	0.0375	0.0241	0.0169	0.0111	0.0122
32K	0.0295	0.0184	0.0129	0.0080	0.0087
64K	0.0202	0.0126	0.0092	0.0059	0.0068
128K	0.0145	0.0090	0.0068	0.0043	0.0051
256K	0.0070	0.0046	0.0041	0.0030	0.0035
512K	0.0030	0.0017	0.0011	0.0013	0.0009
1M	0.0022	0.0011	0.0006	0.0003	0.0002

Unified

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0272	0.0197	0.0154	0.0134	0.0136
16K	0.0141	0.0101	0.0081	0.0070	0.0071
32K	0.0087	0.0057	0.0041	0.0031	0.0034
64K	0.0051	0.0032	0.0023	0.0016	0.0017
128K	0.0035	0.0022	0.0015	0.0010	0.0010
256K	0.0019	0.0012	0.0009	0.0007	0.0007
512K	0.0009	0.0005	0.0003	0.0003	0.0002
1M	0.0006	0.0003	0.0002	0.0001	0

Table 4: Average miss ratios for the floating-point SPEC benchmarks.

Floating Point Averages : Associativity 1

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.0782	0.0416	0.0229	0.0134	0.0087
2K	0.0709	0.0372	0.0203	0.0114	0.0069
4K	0.0604	0.0312	0.0166	0.0092	0.0054
8K	0.0488	0.0251	0.0132	0.0073	0.0042
16K	0.0291	0.0149	0.0078	0.0043	0.0025
32K	0.0182	0.0093	0.0048	0.0026	0.0014
64K	0.0064	0.0033	0.0017	0.0009	0.0005
128K	0.0001	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.2639	0.2540	0.2507	0.2664	0.3028
2K	0.2315	0.2220	0.2157	0.2222	0.2426
4K	0.1898	0.1878	0.1828	0.1862	0.1980
8K	0.1368	0.1280	0.1263	0.1322	0.1436
16K	0.1079	0.0941	0.0907	0.0966	0.1031
32K	0.0875	0.0720	0.0668	0.0711	0.0792
64K	0.0783	0.0609	0.0546	0.0535	0.0622
128K	0.0709	0.0533	0.0447	0.0422	0.0469
256K	0.0634	0.0464	0.0372	0.0336	0.0349
512K	0.0386	0.0286	0.0234	0.0212	0.0208
1M	0.0165	0.0092	0.0056	0.0040	0.0037

Unified

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.1736	0.1452	0.1430	0.1737	0.2502
2K	0.1503	0.1210	0.1112	0.1204	0.1563
4K	0.1225	0.0982	0.0867	0.0880	0.1033
8K	0.0921	0.0700	0.0600	0.0594	0.0673
16K	0.0655	0.0488	0.0418	0.0416	0.0454
32K	0.0491	0.0355	0.0297	0.0295	0.0333
64K	0.0340	0.0250	0.0213	0.0205	0.0236
128K	0.0268	0.0198	0.0163	0.0153	0.0171
256K	0.0240	0.0173	0.0138	0.0124	0.0130
512K	0.0156	0.0111	0.0088	0.0078	0.0077
1M	0.0086	0.0049	0.0031	0.0023	0.0021

Floating Point Averages : Associativity 2

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.0692	0.0360	0.0195	0.0110	0.0066
4K	0.0605	0.0313	0.0167	0.0091	0.0054
8K	0.0492	0.0253	0.0133	0.0073	0.0041
16K	0.0285	0.0146	0.0076	0.0041	0.0024
32K	0.0226	0.0114	0.0058	0.0030	0.0015
64K	0.0081	0.0042	0.0021	0.0011	0.0006
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.2075	0.1899	0.1783	0.1838	0.1949
4K	0.1753	0.1638	0.1526	0.1528	0.1596
8K	0.1190	0.1170	0.1131	0.1167	0.1223
16K	0.0809	0.0665	0.0717	0.0763	0.0811
32K	0.0645	0.0452	0.0439	0.0544	0.0588
64K	0.0565	0.0367	0.0294	0.0318	0.0450
128K	0.0526	0.0324	0.0232	0.0204	0.0247
256K	0.0490	0.0294	0.0194	0.0148	0.0140
512K	0.0360	0.0205	0.0126	0.0089	0.0075
1M	0.0165	0.0090	0.0053	0.0035	0.0029

Unified

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.1308	0.0987	0.0815	0.0777	0.0823
4K	0.1127	0.0848	0.0684	0.0614	0.0610
8K	0.0837	0.0633	0.0512	0.0468	0.0456
16K	0.0539	0.0373	0.0322	0.0304	0.0301
32K	0.0386	0.0241	0.0193	0.0205	0.0208
64K	0.0279	0.0171	0.0123	0.0119	0.0155
128K	0.0186	0.0113	0.0080	0.0069	0.0083
256K	0.0167	0.0100	0.0065	0.0050	0.0047
512K	0.0125	0.0071	0.0043	0.0030	0.0026
1M	0.0052	0.0029	0.0016	0.0011	0.0010

Floating Point Averages : Associativity 4

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0600	0.0310	0.0164	0.0089	0.0052
8K	0.0484	0.0249	0.0131	0.0072	0.0040
16K	0.0277	0.0141	0.0073	0.0039	0.0021
32K	0.0227	0.0114	0.0058	0.0029	0.0015
64K	0.0028	0.0015	0.0009	0.0005	0.0003
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.1559	0.1451	0.1344	0.1358	0.1437
8K	0.1179	0.1197	0.1136	0.1157	0.1203
16K	0.0755	0.0635	0.0704	0.0707	0.0734
32K	0.0621	0.0407	0.0419	0.0547	0.0561
64K	0.0552	0.0340	0.0229	0.0308	0.0439
128K	0.0523	0.0312	0.0200	0.0145	0.0250
256K	0.0491	0.0289	0.0182	0.0126	0.0099
512K	0.0389	0.0219	0.0130	0.0085	0.0062
1M	0.0171	0.0095	0.0056	0.0037	0.0027

Unified

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.1054	0.0766	0.0602	0.0549	0.0545
8K	0.0828	0.0636	0.0506	0.0454	0.0442
16K	0.0519	0.0366	0.0318	0.0286	0.0277
32K	0.0364	0.0214	0.0180	0.0200	0.0193
64K	0.0225	0.0134	0.0088	0.0109	0.0147
128K	0.0167	0.0099	0.0064	0.0047	0.0082
256K	0.0154	0.0090	0.0057	0.0040	0.0032
512K	0.0123	0.0069	0.0042	0.0027	0.0020
1M	0.0054	0.0030	0.0018	0.0012	0.0008

Floating Point Averages : Associativity 8

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0480	0.0248	0.0132	0.0071	0.0040
16K	0.0287	0.0147	0.0077	0.0041	0.0022
32K	0.0227	0.0114	0.0058	0.0029	0.0015
64K	0.0014	0.0008	0.0005	0.0003	0.0002
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.1088	0.1005	0.0879	0.0848	0.0856
16K	0.0772	0.0664	0.0712	0.0699	0.0711
32K	0.0612	0.0397	0.0434	0.0553	0.0550
64K	0.0546	0.0333	0.0220	0.0327	0.0464
128K	0.0524	0.0307	0.0194	0.0134	0.0263
256K	0.0488	0.0284	0.0177	0.0118	0.0090
512K	0.0390	0.0219	0.0130	0.0085	0.0062
1M	0.0174	0.0098	0.0059	0.0039	0.0029

Unified

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0786	0.0564	0.0415	0.0344	0.0328
16K	0.0513	0.0362	0.0315	0.0277	0.0260
32K	0.0357	0.0206	0.0180	0.0198	0.0187
64K	0.0233	0.0137	0.0087	0.0116	0.0155
128K	0.0164	0.0096	0.0061	0.0043	0.0086
256K	0.0153	0.0089	0.0055	0.0037	0.0029
512K	0.0123	0.0069	0.0042	0.0027	0.0020
1M	0.0055	0.0031	0.0019	0.0013	0.0009

Table 5: Average miss ratios for the complete SPEC benchmark suite.

Overall Averages : Associativity 1

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.0822	0.0482	0.0295	0.0195	0.0144
2K	0.0665	0.0381	0.0229	0.0145	0.0100
4K	0.0534	0.0303	0.0179	0.0113	0.0077
8K	0.0384	0.0212	0.0123	0.0076	0.0050
16K	0.0222	0.0121	0.0070	0.0044	0.0029
32K	0.0138	0.0074	0.0042	0.0025	0.0016
64K	0.0051	0.0028	0.0016	0.0009	0.0006
128K	0.0007	0.0004	0.0003	0.0002	0.0002
256K	0.0002	0.0001	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.2249	0.2145	0.2235	0.2619	0.2926
2K	0.1888	0.1775	0.1790	0.1993	0.2187
4K	0.1518	0.1446	0.1433	0.1535	0.1662
8K	0.1111	0.1001	0.0991	0.1056	0.1152
16K	0.0873	0.0737	0.0711	0.0751	0.0804
32K	0.0674	0.0534	0.0483	0.0499	0.0557
64K	0.0576	0.0435	0.0381	0.0365	0.0421
128K	0.0495	0.0365	0.0302	0.0279	0.0310
256K	0.0424	0.0306	0.0245	0.0218	0.0227
512K	0.0257	0.0187	0.0152	0.0136	0.0135
1M	0.0112	0.0063	0.0039	0.0028	0.0026

Unified

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.1645	0.1376	0.1357	0.1688	0.2234
2K	0.1330	0.1078	0.1003	0.1130	0.1401
4K	0.1052	0.0844	0.0758	0.0798	0.0914
8K	0.0749	0.0575	0.0508	0.0515	0.0573
16K	0.0515	0.0386	0.0338	0.0341	0.0371
32K	0.0374	0.0270	0.0228	0.0227	0.0254
64K	0.0251	0.0184	0.0158	0.0152	0.0174
128K	0.0187	0.0136	0.0112	0.0106	0.0118
256K	0.0161	0.0115	0.0092	0.0084	0.0088
512K	0.0105	0.0075	0.0060	0.0054	0.0053
1M	0.0059	0.0035	0.0024	0.0019	0.0018

Overall Averages : Associativity 2

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.0622	0.0355	0.0210	0.0137	0.0090
4K	0.0481	0.0269	0.0158	0.0098	0.0066
8K	0.0352	0.0192	0.0109	0.0067	0.0043
16K	0.0199	0.0107	0.0060	0.0036	0.0024
32K	0.0148	0.0077	0.0041	0.0023	0.0013
64K	0.0055	0.0029	0.0015	0.0009	0.0005
128K	0.0004	0.0003	0.0002	0.0001	0.0001
256K	0.0001	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.1628	0.1452	0.1381	0.1496	0.1629
4K	0.1354	0.1215	0.1124	0.1145	0.1231
8K	0.0955	0.0881	0.0830	0.0841	0.0895
16K	0.0674	0.0537	0.0544	0.0554	0.0592
32K	0.0511	0.0353	0.0322	0.0367	0.0398
64K	0.0427	0.0275	0.0216	0.0217	0.0300
128K	0.0376	0.0231	0.0168	0.0141	0.0170
256K	0.0330	0.0199	0.0134	0.0102	0.0098
512K	0.0234	0.0134	0.0084	0.0060	0.0052
1M	0.0109	0.0059	0.0034	0.0023	0.0019

Unified

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.1143	0.0860	0.0718	0.0706	0.0737
4K	0.0903	0.0679	0.0556	0.0512	0.0509
8K	0.0646	0.0488	0.0400	0.0368	0.0361
16K	0.0407	0.0287	0.0246	0.0230	0.0229
32K	0.0278	0.0177	0.0141	0.0144	0.0145
64K	0.0194	0.0121	0.0088	0.0081	0.0103
128K	0.0127	0.0078	0.0056	0.0047	0.0055
256K	0.0110	0.0066	0.0044	0.0033	0.0031
512K	0.0080	0.0046	0.0028	0.0020	0.0017
1M	0.0034	0.0019	0.0011	0.0007	0.0006

Overall Averages : Associativity 4

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0465	0.0257	0.0149	0.0093	0.0062
8K	0.0338	0.0184	0.0106	0.0064	0.0041
16K	0.0184	0.0097	0.0054	0.0032	0.0021
32K	0.0145	0.0074	0.0039	0.0021	0.0012
64K	0.0022	0.0012	0.0007	0.0005	0.0003
128K	0.0003	0.0002	0.0001	0	0
256K	0.0001	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.1186	0.1043	0.0943	0.0955	0.1029
8K	0.0907	0.0850	0.0778	0.0769	0.0810
16K	0.0605	0.0479	0.0492	0.0471	0.0493
32K	0.0493	0.0319	0.0304	0.0361	0.0374
64K	0.0412	0.0255	0.0175	0.0209	0.0291
128K	0.0372	0.0223	0.0148	0.0104	0.0171
256K	0.0326	0.0193	0.0126	0.0088	0.0073
512K	0.0248	0.0139	0.0085	0.0057	0.0042
1M	0.0111	0.0062	0.0036	0.0024	0.0017

Unified

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0829	0.0605	0.0480	0.0443	0.0437
8K	0.0611	0.0465	0.0370	0.0332	0.0325
16K	0.0373	0.0264	0.0226	0.0202	0.0198
32K	0.0255	0.0153	0.0126	0.0135	0.0131
64K	0.0157	0.0095	0.0063	0.0073	0.0096
128K	0.0115	0.0068	0.0045	0.0032	0.0054
256K	0.0101	0.0059	0.0038	0.0027	0.0022
512K	0.0078	0.0044	0.0027	0.0018	0.0013
1M	0.0035	0.0019	0.0011	0.0008	0.0005

Overall Averages : Associativity 8

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0332	0.0181	0.0105	0.0063	0.0041
16K	0.0187	0.0098	0.0055	0.0033	0.0021
32K	0.0144	0.0074	0.0038	0.0021	0.0011
64K	0.0013	0.0008	0.0005	0.0003	0.0002
128K	0.0003	0.0002	0.0001	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0850	0.0731	0.0620	0.0578	0.0594
16K	0.0614	0.0495	0.0495	0.0464	0.0476
32K	0.0485	0.0312	0.0312	0.0364	0.0365
64K	0.0408	0.0250	0.0169	0.0220	0.0306
128K	0.0372	0.0220	0.0144	0.0098	0.0178
256K	0.0321	0.0189	0.0123	0.0083	0.0068
512K	0.0246	0.0138	0.0083	0.0056	0.0041
1M	0.0113	0.0063	0.0037	0.0025	0.0018

Unified

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0581	0.0417	0.0311	0.0260	0.0251
16K	0.0364	0.0258	0.0221	0.0195	0.0184
32K	0.0249	0.0146	0.0125	0.0131	0.0125
64K	0.0160	0.0095	0.0062	0.0076	0.0100
128K	0.0113	0.0066	0.0043	0.0030	0.0056
256K	0.0099	0.0058	0.0037	0.0025	0.0020
512K	0.0077	0.0043	0.0026	0.0017	0.0013
1M	0.0035	0.0020	0.0012	0.0008	0.0006

All of the data in the literature (see e.g. [Smit82], [Ande91], [Agar88]) suggests that operating systems activity significantly increases miss ratios. First, operating systems code tends to loop less than user code, and so instruction miss ratios are high. Second, operating systems routines are usually called into the cache by an exception, interrupt or trap, then run for a short time, and finally are replaced from the cache before they run again; they effectively always face a "cold start" situation. Sanguinetti observes [Sang84] that for the Amdahl 580, routines must execute over 600 times per second to stay cache resident. Third, operating system activity is associated with timesharing and high levels of multiprogramming; frequent task switching means that programs are constantly experiencing cold start. As illustrated by Figure 3, miss ratios for the SPEC benchmarks are considerably below those for any workloads with significant OS activity. Similar differences in cache performance between compute bound and multiprogrammed environments are reported in [Mogu91]. The SPEC floating-point benchmark miss ratios are quite close to the DTMRs, the data from [Agar88], and the VAX 11/780 measurements, and for large cache sizes are also very close to the Amdahl 470 user program miss ratios. The SPEC integer benchmark miss ratios are lowest.

4. Conclusions

The purpose of this study is two-fold: to show measurements of the cache performance of the SPEC benchmarks and to comment on the usefulness of those benchmarks for cache and memory system design. While the cache performance of the SPEC benchmarks varies from program to program, we have found that the floating-point benchmarks generally require much larger cache sizes relative to the integer benchmarks. The integer benchmarks use no more than 128 Kbytes of instruction and 128 Kbytes of data cache, while the floating-point programs can take advantage of data caches of a megabyte or more.

Comparisons with other studies suggest that the SPEC integer benchmarks are too small to represent real workloads. Miss ratios for the SPEC floating-point benchmarks seem consistent with previous measurements of user program miss ratios but are quite low relative to supervisor code miss ratios.

From these measurements and comparisons, we conclude that miss ratios for the SPEC benchmarks are potentially typical of only a certain environment - Unix workstations running user state CPU bound jobs as the single active user process. The integer benchmarks have very low miss ratios, and provide very little stress on the memory system. The floating point benchmarks provide reasonable measurements of memory system performance for user code, but are still much better behaved than commercial and timeshared workloads. The SPEC benchmarks are conspicuously lacking a significant operating system component, which affects their validity in two ways: miss ratios are too low, and the performance impacts of operating systems functions themselves are not tested.

Bibliography

[Agar88] A. Agarwal, J. Hennessy, and M. Horowitz, "Cache Performance of Operating System and Multiprogramming Workloads," *ACM Trans. Comp. Sys.*, vol. 6, 4, November 1988, pp. 393-433.

[Ande91] T.E. Anderson, H.M. Levy, B.N. Bershad, and E.D. Lazowska, "The Interaction of Architecture and Operating System Design," Proc. ASPLOS-IV, April, 1991, Santa Clara, CA, pp. 108-120.

[Borg90] A. Borg, R.E. Kessler, and D.W. Wall, "Generation and Analysis of Very Long Address Traces," *Proc. 17th Int'l Symp. Comp. Arch.*, May, 1990, Seattle, WA, pp. 270-279.

[Cmel91] R. M. Cmelik, S. I. Kong, D. R. Ditzel, and E. J. Kelly, "An Analysis of SPARC and MIPS Instruction Set Utilization on the SPEC benchmarks," Proc. ASPLOS-IV, April, 1991, Santa Clara, CA, pp. 290-302.

[DEC91] "Pixie," DEC Ultrix manual page.

[Clar83] D.W. Clark, "Cache Performance in the VAX-11/780," *ACM Trans. Comp. Sys.*, vol. 1, 1, February 1983, pp. 24-37.

[Clar88] D.W. Clark, P.J. Bannon, J.B. Keller, "Measuring VAX 8800 Performance with a Histogram Hardware Monitor," *Proc. 15th Int'l Symp. Comp. Arch.*, May, 1988, Honolulu, HI, pp. 176-185.

[Hill87] M.D. Hill, "Aspects of Cache Memory and Instruction Buffer Performance," Ph.D. Thesis, Univ. of California at Berkeley, Technical Report UCB/CSD 87/381, November 1987.

[Hill89] Mark Hill and Alan Jay Smith, "Evaluating Associativity in CPU Caches," IEEETC, 38, 12, December, 1989, pp. 1612-1630.

[Hill] M.D. Hill, "Tycho," Unpublished UNIX-style manual page. The Tycho simulator is available from Prof. Mark Hill, Computer Sciences Dept., University of Wisconsin.

[Hinn88] David Hinnant, "Accurate Unix Benchmarking: Art, Science or Black Magic?," IEEE MICRO, October, 1988, pp. 64-75.

[Mogu91] J. C. Mogul, and Anita Borg, "The Effects of Context Switches on Cache Performance," Proc. ASPLOS-IV, April, 1991, Santa Clara, CA, pp. 75-84.

[Pnev90] D.N. Pnevmatikatos, M.D. Hill, "Cache Performance of the Integer SPEC Benchmarks on a RISC," *Computer Architecture News*, vol. 18, 2, June 1990, pp. 53-68.

[Pric89] Walter Price, "A Benchmark Tutorial," IEEE MICRO, October, 1989, pp. 28-43.

[Saav90] Rafael H. Saavedra-Barrera, and Alan Jay Smith, "Performance Prediction by Benchmark and Machine Analysis," UC Berkeley Computer Science Division Technical Report UCB/CSD 90/607, December, 1990.

[Saav91] Rafael Saavedra-Barrera and Alan Jay Smith, "Analysis of Standard Benchmark Programs," in preparation, available fall, 1991.

[Sang84] John Sanguinetti, "Program Optimization for a Pipelined Machine: A Case Study," Proc. 1984 ACM Sigmetrics Conf. on Measurement and Modeling of Computer Systems, August, 1984, Cambridge, Mass., pp. 88-95.

[Scot90] Vicki Scott, "Is Standardization of Benchmarks Feasible?," Proceedings of the Buscon Conference, Feb 14-16, 1990. Long Beach, CA, 139-147.

[Smit82] Alan Jay Smith, "Cache Memories," *Computing Surveys*, vol. 14, 3, September 1982.

[Smit85] Alan Jay Smith, "Cache Evaluation and the Impact of Workload Choice," Proc. 12'th International Symposium on Computer Architecture, June 17-19, 1985, Boston, Mass, pp. 64-75.

[Smit87] Alan Jay Smith, "Line (Block) Size Choice for CPU Cache Memories," *IEEE Trans. on Computers*, vol. C-36, 9, September 1987, pp. 1063-1075.

[Spec89] SPEC newsletter, vol. 1, 1989.

[Spec90] SPEC newsletter, vol. 2, 1990.

Appendix[†]

Disclaimer: Data is this appendix is correct to the best of our knowledge. However, we provide it *as is* without an expressed or implied warranty, and we accept no responsibility for the consequences of the use or misuse of this data.

[†] This paper (in postscript) and the average tables and appendix (in ascii) are available via anonymous ftp:

```
ftp reggiano.cs.wisc.edu  (or: ftp 128.105.8.27 )
reply to login: anonymous
reply to passwd: type any non-null string here
cd SPEC
get README
get body.postscript
get averages.postscript
get appendix.postscript
get averages.ascii
get appendix.ascii
bye
```

Doduc : Associativity 1

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.1791	0.0979	0.0570	0.0345	0.0244
2K	0.1498	0.0815	0.0471	0.0284	0.0191
4K	0.1040	0.0560	0.0313	0.0190	0.0132
8K	0.0605	0.0327	0.0182	0.0117	0.0081
16K	0.0280	0.0154	0.0087	0.0061	0.0046
32K	0.0118	0.0065	0.0037	0.0024	0.0018
64K	0.0064	0.0035	0.0019	0.0012	0.0008
128K	0.0007	0.0004	0.0003	0.0003	0.0002
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.2034	0.2011	0.2087	0.2064	0.2218
2K	0.1431	0.1442	0.1521	0.1595	0.1734
4K	0.0987	0.0940	0.1026	0.1120	0.1306
8K	0.0665	0.0489	0.0545	0.0694	0.0835
16K	0.0404	0.0317	0.0319	0.0348	0.0416
32K	0.0162	0.0111	0.0122	0.0110	0.0142
64K	0.0140	0.0092	0.0107	0.0093	0.0116
128K	0.0071	0.0040	0.0060	0.0047	0.0051
256K	0.0001	0.0001	0.0001	0	0
512K	0.0001	0.0001	0.0001	0	0
1M	0.0001	0.0001	0.0001	0	0

Unified

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.2327	0.1707	0.1482	0.1458	0.2225
2K	0.1971	0.1403	0.1137	0.1061	0.1534
4K	0.1484	0.1040	0.0835	0.0748	0.0905
8K	0.1019	0.0687	0.0534	0.0484	0.0525
16K	0.0591	0.0408	0.0328	0.0293	0.0300
32K	0.0315	0.0205	0.0161	0.0139	0.0148
64K	0.0224	0.0145	0.0116	0.0092	0.0091
128K	0.0147	0.0093	0.0075	0.0057	0.0051
256K	0.0099	0.0065	0.0049	0.0035	0.0031
512K	0.0099	0.0065	0.0049	0.0035	0.0031
1M	0.0099	0.0065	0.0049	0.0035	0.0031

Doduc : Associativity 2

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.1436	0.0779	0.0451	0.0276	0.0184
4K	0.1028	0.0557	0.0319	0.0189	0.0131
8K	0.0586	0.0317	0.0180	0.0112	0.0075
16K	0.0220	0.0120	0.0068	0.0041	0.0031
32K	0.0116	0.0063	0.0035	0.0020	0.0012
64K	0.0042	0.0025	0.0015	0.0010	0.0007
128K	0.0002	0.0001	0.0001	0.0001	0.0001
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.1208	0.1214	0.1335	0.1442	0.1499
4K	0.0681	0.0684	0.0780	0.0915	0.1043
8K	0.0426	0.0348	0.0383	0.0485	0.0610
16K	0.0242	0.0169	0.0162	0.0176	0.0238
32K	0.0105	0.0074	0.0069	0.0052	0.0065
64K	0.0008	0.0009	0.0010	0.0008	0.0008
128K	0.0004	0.0004	0.0004	0.0002	0.0001
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Unified

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.1866	0.1255	0.0933	0.0744	0.0686
4K	0.1408	0.0958	0.0710	0.0572	0.0522
8K	0.0887	0.0576	0.0430	0.0368	0.0334
16K	0.0425	0.0278	0.0221	0.0205	0.0187
32K	0.0194	0.0114	0.0083	0.0071	0.0065
64K	0.0107	0.0062	0.0039	0.0026	0.0019
128K	0.0052	0.0032	0.0022	0.0014	0.0011
256K	0.0009	0.0005	0.0004	0.0003	0.0002
512K	0	0	0	0	0
1M	0	0	0	0	0

Doduc : Associativity 4

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.1018	0.0549	0.0308	0.0181	0.0125
8K	0.0578	0.0310	0.0174	0.0100	0.0067
16K	0.0139	0.0074	0.0041	0.0023	0.0017
32K	0.0119	0.0063	0.0034	0.0019	0.0011
64K	0.0046	0.0027	0.0017	0.0011	0.0007
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0568	0.0597	0.0726	0.0855	0.0974
8K	0.0337	0.0242	0.0241	0.0381	0.0516
16K	0.0222	0.0147	0.0116	0.0099	0.0145
32K	0.0067	0.0054	0.0051	0.0036	0.0023
64K	0.0001	0.0002	0.0003	0.0003	0.0003
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Unified

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.1317	0.0856	0.0648	0.0546	0.0483
8K	0.0854	0.0553	0.0402	0.0335	0.0320
16K	0.0324	0.0224	0.0180	0.0162	0.0156
32K	0.0170	0.0098	0.0064	0.0046	0.0041
64K	0.0110	0.0064	0.0040	0.0023	0.0014
128K	0.0011	0.0009	0.0008	0.0007	0.0005
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Doduc : Associativity 8

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0578	0.0312	0.0174	0.0098	0.0067
16K	0.0138	0.0072	0.0039	0.0021	0.0012
32K	0.0124	0.0065	0.0035	0.0019	0.0011
64K	0.0040	0.0024	0.0017	0.0011	0.0008
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0323	0.0207	0.0203	0.0359	0.0475
16K	0.0222	0.0140	0.0102	0.0082	0.0111
32K	0.0045	0.0049	0.0051	0.0036	0.0024
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Unified

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0820	0.0528	0.0386	0.0319	0.0305
16K	0.0238	0.0173	0.0169	0.0158	0.0140
32K	0.0167	0.0094	0.0058	0.0037	0.0032
64K	0.0113	0.0065	0.0041	0.0024	0.0014
128K	0.0007	0.0005	0.0007	0.0007	0.0006
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Eqntott : Associativity 1

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.0141	0.0116	0.0091	0.0086	0.0097
2K	0.0048	0.0034	0.0022	0.0018	0.0027
4K	0.0039	0.0030	0.0018	0.0012	0.0022
8K	0.0008	0.0004	0.0002	0.0002	0.0002
16K	0.0008	0.0004	0.0002	0.0002	0.0002
32K	0.0008	0.0004	0.0002	0.0002	0.0002
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.1237	0.1080	0.1371	0.3293	0.2216
2K	0.0993	0.0764	0.0877	0.1901	0.1208
4K	0.0859	0.0618	0.0618	0.1118	0.0756
8K	0.0775	0.0530	0.0481	0.0667	0.0520
16K	0.0719	0.0475	0.0402	0.0429	0.0389
32K	0.0660	0.0426	0.0347	0.0297	0.0311
64K	0.0575	0.0369	0.0297	0.0226	0.0261
128K	0.0450	0.0289	0.0235	0.0170	0.0205
256K	0.0279	0.0184	0.0156	0.0119	0.0143
512K	0.0144	0.0098	0.0086	0.0074	0.0085
1M	0.0048	0.0035	0.0030	0.0026	0.0029

Unified

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.0528	0.0544	0.0647	0.1637	0.1018
2K	0.0294	0.0284	0.0335	0.0891	0.0531
4K	0.0216	0.0185	0.0193	0.0471	0.0288
8K	0.0148	0.0115	0.0116	0.0249	0.0153
16K	0.0125	0.0088	0.0080	0.0139	0.0095
32K	0.0111	0.0074	0.0063	0.0083	0.0065
64K	0.0088	0.0058	0.0048	0.0051	0.0048
128K	0.0069	0.0045	0.0038	0.0036	0.0037
256K	0.0042	0.0028	0.0025	0.0023	0.0025
512K	0.0022	0.0015	0.0013	0.0012	0.0013
1M	0.0007	0.0005	0.0005	0.0004	0.0005

Eqntott : Associativity 2

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.0006	0.0003	0.0003	0.0009	0.0009
4K	0.0006	0.0003	0.0003	0.0002	0.0003
8K	0	0	0	0	0
16K	0	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.0761	0.0518	0.0466	0.0966	0.0698
4K	0.0735	0.0482	0.0391	0.0465	0.0370
8K	0.0714	0.0462	0.0362	0.0290	0.0307
16K	0.0687	0.0440	0.0343	0.0227	0.0281
32K	0.0645	0.0409	0.0319	0.0199	0.0261
64K	0.0569	0.0358	0.0280	0.0176	0.0231
128K	0.0425	0.0273	0.0222	0.0144	0.0189
256K	0.0233	0.0156	0.0141	0.0104	0.0130
512K	0.0077	0.0052	0.0052	0.0049	0.0055
1M	0.0018	0.0011	0.0009	0.0011	0.0010

Unified

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.0171	0.0128	0.0139	0.0421	0.0236
4K	0.0125	0.0086	0.0076	0.0161	0.0096
8K	0.0109	0.0071	0.0057	0.0063	0.0054
16K	0.0103	0.0066	0.0052	0.0038	0.0043
32K	0.0097	0.0061	0.0048	0.0031	0.0039
64K	0.0085	0.0054	0.0042	0.0027	0.0035
128K	0.0063	0.0041	0.0033	0.0022	0.0028
256K	0.0035	0.0023	0.0021	0.0015	0.0019
512K	0.0011	0.0008	0.0008	0.0007	0.0008
1M	0.0003	0.0002	0.0001	0.0002	0.0001

Eqntott : Associativity 4

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
4K	0	0	0	0	0
8K	0	0	0	0	0
16K	0	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0727	0.0471	0.0367	0.0371	0.0321
8K	0.0711	0.0458	0.0356	0.0225	0.0287
16K	0.0687	0.0439	0.0341	0.0208	0.0274
32K	0.0644	0.0408	0.0318	0.0195	0.0258
64K	0.0569	0.0356	0.0279	0.0175	0.0230
128K	0.0422	0.0271	0.0220	0.0140	0.0186
256K	0.0187	0.0131	0.0129	0.0100	0.0126
512K	0.0042	0.0030	0.0035	0.0042	0.0043
1M	0.0014	0.0007	0.0004	0.0005	0.0004

Unified

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0114	0.0076	0.0061	0.0095	0.0065
8K	0.0107	0.0069	0.0054	0.0036	0.0044
16K	0.0103	0.0066	0.0051	0.0031	0.0041
32K	0.0096	0.0061	0.0047	0.0029	0.0039
64K	0.0085	0.0053	0.0042	0.0026	0.0034
128K	0.0063	0.0040	0.0033	0.0021	0.0028
256K	0.0028	0.0020	0.0019	0.0015	0.0019
512K	0.0006	0.0004	0.0005	0.0006	0.0006
1M	0.0002	0.0001	0	0	0

Eqntott : Associativity 8

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
8K	0	0	0	0	0
16K	0	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0711	0.0458	0.0355	0.0218	0.0286
16K	0.0686	0.0439	0.0341	0.0207	0.0274
32K	0.0644	0.0407	0.0317	0.0194	0.0258
64K	0.0568	0.0355	0.0277	0.0175	0.0229
128K	0.0423	0.0271	0.0220	0.0139	0.0184
256K	0.0158	0.0118	0.0126	0.0100	0.0125
512K	0.0023	0.0015	0.0016	0.0036	0.0029
1M	0.0012	0.0006	0.0003	0.0001	0.0002

Unified

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0107	0.0069	0.0053	0.0033	0.0043
16K	0.0103	0.0066	0.0051	0.0031	0.0041
32K	0.0096	0.0061	0.0047	0.0029	0.0038
64K	0.0085	0.0053	0.0041	0.0026	0.0034
128K	0.0063	0.0040	0.0033	0.0021	0.0027
256K	0.0024	0.0018	0.0019	0.0015	0.0019
512K	0.0003	0.0002	0.0002	0.0005	0.0004
1M	0.0002	0	0	0	0

Espresso : Associativity 1

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.0455	0.0312	0.0214	0.0149	0.0135
2K	0.0180	0.0112	0.0076	0.0055	0.0044
4K	0.0112	0.0070	0.0045	0.0030	0.0025
8K	0.0076	0.0047	0.0029	0.0020	0.0016
16K	0.0039	0.0024	0.0015	0.0009	0.0006
32K	0.0035	0.0021	0.0013	0.0008	0.0005
64K	0.0019	0.0012	0.0007	0.0004	0.0002
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.1718	0.1500	0.1761	0.2237	0.3277
2K	0.1357	0.1120	0.1255	0.1515	0.2200
4K	0.1069	0.0844	0.0901	0.1040	0.1457
8K	0.0732	0.0543	0.0514	0.0538	0.0700
16K	0.0482	0.0344	0.0323	0.0320	0.0375
32K	0.0243	0.0178	0.0127	0.0116	0.0139
64K	0.0091	0.0066	0.0052	0.0050	0.0064
128K	0.0043	0.0032	0.0022	0.0020	0.0023
256K	0.0006	0.0004	0.0003	0.0003	0.0004
512K	0.0002	0.0001	0	0	0
1M	0.0002	0	0	0	0

Unified

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.1203	0.1075	0.1116	0.1273	0.1745
2K	0.0751	0.0656	0.0703	0.0774	0.1106
4K	0.0539	0.0470	0.0505	0.0541	0.0755
8K	0.0317	0.0245	0.0270	0.0277	0.0382
16K	0.0196	0.0142	0.0153	0.0157	0.0209
32K	0.0127	0.0088	0.0070	0.0070	0.0087
64K	0.0070	0.0050	0.0042	0.0044	0.0055
128K	0.0041	0.0030	0.0028	0.0032	0.0041
256K	0.0029	0.0022	0.0021	0.0027	0.0033
512K	0.0027	0.0020	0.0020	0.0025	0.0032
1M	0.0027	0.0020	0.0020	0.0025	0.0032

Espresso : Associativity 2

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.0166	0.0102	0.0069	0.0051	0.0039
4K	0.0086	0.0054	0.0038	0.0027	0.0022
8K	0.0022	0.0015	0.0010	0.0007	0.0007
16K	0.0005	0.0003	0.0002	0.0002	0.0002
32K	0.0002	0.0001	0	0	0
64K	0.0002	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.1042	0.0722	0.0599	0.0638	0.1028
4K	0.0823	0.0543	0.0404	0.0371	0.0554
8K	0.0592	0.0391	0.0279	0.0234	0.0281
16K	0.0332	0.0227	0.0145	0.0108	0.0113
32K	0.0176	0.0131	0.0077	0.0051	0.0042
64K	0.0032	0.0023	0.0016	0.0012	0.0011
128K	0.0009	0.0005	0.0004	0.0003	0.0002
256K	0.0004	0.0002	0.0001	0	0
512K	0.0002	0.0001	0	0	0
1M	0.0002	0	0	0	0

Unified

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.0519	0.0387	0.0370	0.0383	0.0540
4K	0.0321	0.0219	0.0197	0.0192	0.0253
8K	0.0205	0.0140	0.0121	0.0108	0.0126
16K	0.0111	0.0076	0.0053	0.0041	0.0044
32K	0.0052	0.0037	0.0024	0.0017	0.0016
64K	0.0016	0.0011	0.0007	0.0005	0.0005
128K	0.0004	0.0003	0.0002	0.0002	0.0001
256K	0.0001	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Espresso : Associativity 4

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0084	0.0053	0.0035	0.0026	0.0021
8K	0.0020	0.0014	0.0009	0.0006	0.0007
16K	0.0002	0.0001	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0766	0.0488	0.0321	0.0257	0.0271
8K	0.0569	0.0357	0.0214	0.0147	0.0134
16K	0.0300	0.0207	0.0125	0.0082	0.0069
32K	0.0173	0.0115	0.0065	0.0040	0.0029
64K	0.0026	0.0021	0.0013	0.0009	0.0007
128K	0.0005	0.0003	0.0002	0.0001	0.0001
256K	0.0003	0.0001	0	0	0
512K	0.0002	0.0001	0	0	0
1M	0.0002	0	0	0	0

Unified

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0296	0.0194	0.0140	0.0121	0.0135
8K	0.0181	0.0118	0.0080	0.0062	0.0060
16K	0.0088	0.0060	0.0039	0.0029	0.0027
32K	0.0041	0.0027	0.0017	0.0012	0.0010
64K	0.0008	0.0006	0.0004	0.0002	0.0002
128K	0.0002	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Espresso : Associativity 8

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0013	0.0009	0.0006	0.0004	0.0006
16K	0.0001	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0567	0.0350	0.0203	0.0133	0.0114
16K	0.0290	0.0208	0.0121	0.0075	0.0054
32K	0.0174	0.0114	0.0064	0.0039	0.0027
64K	0.0026	0.0023	0.0013	0.0008	0.0007
128K	0.0005	0.0003	0.0002	0.0001	0
256K	0.0003	0.0001	0	0	0
512K	0.0002	0.0001	0	0	0
1M	0.0002	0	0	0	0

Unified

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0172	0.0114	0.0076	0.0059	0.0054
16K	0.0081	0.0055	0.0034	0.0024	0.0021
32K	0.0039	0.0026	0.0015	0.0009	0.0007
64K	0.0007	0.0005	0.0003	0.0002	0.0002
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Fpppp : Associativity 1

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.2456	0.1242	0.0635	0.0330	0.0178
2K	0.2411	0.1220	0.0624	0.0325	0.0173
4K	0.2352	0.1188	0.0607	0.0315	0.0166
8K	0.2173	0.1098	0.0563	0.0291	0.0152
16K	0.1424	0.0719	0.0367	0.0191	0.0101
32K	0.0957	0.0482	0.0246	0.0126	0.0065
64K	0.0322	0.0162	0.0084	0.0043	0.0023
128K	0.0002	0.0001	0.0001	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.1792	0.1656	0.1616	0.2003	0.2665
2K	0.1158	0.1035	0.1000	0.1136	0.1542
4K	0.0601	0.0539	0.0489	0.0510	0.0604
8K	0.0556	0.0494	0.0447	0.0458	0.0552
16K	0.0468	0.0413	0.0361	0.0358	0.0450
32K	0.0434	0.0377	0.0312	0.0305	0.0386
64K	0.0407	0.0345	0.0277	0.0266	0.0348
128K	0.0405	0.0342	0.0274	0.0263	0.0344
256K	0.0404	0.0342	0.0274	0.0262	0.0343
512K	0	0	0	0	0
1M	0	0	0	0	0

Unified

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.2628	0.1797	0.1576	0.1905	0.2863
2K	0.2353	0.1487	0.1163	0.1193	0.1712
4K	0.2097	0.1225	0.0830	0.0710	0.0868
8K	0.1886	0.1083	0.0687	0.0537	0.0595
16K	0.1380	0.0811	0.0513	0.0384	0.0405
32K	0.1056	0.0630	0.0399	0.0300	0.0330
64K	0.0469	0.0301	0.0205	0.0168	0.0186
128K	0.0261	0.0192	0.0146	0.0133	0.0165
256K	0.0257	0.0189	0.0143	0.0131	0.0163
512K	0.0102	0.0054	0.0031	0.0022	0.0018
1M	0.0102	0.0054	0.0031	0.0022	0.0018

Fpppp : Associativity 2

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.2416	0.1220	0.0623	0.0322	0.0169
4K	0.2351	0.1188	0.0608	0.0313	0.0164
8K	0.2250	0.1134	0.0577	0.0297	0.0155
16K	0.1459	0.0736	0.0377	0.0198	0.0106
32K	0.1240	0.0622	0.0314	0.0159	0.0081
64K	0.0446	0.0224	0.0113	0.0058	0.0030
128K	0.0003	0.0002	0.0001	0.0001	0.0001
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.1027	0.0853	0.0725	0.0793	0.0993
4K	0.0501	0.0446	0.0348	0.0316	0.0340
8K	0.0105	0.0114	0.0115	0.0123	0.0147
16K	0.0080	0.0085	0.0086	0.0092	0.0114
32K	0.0036	0.0040	0.0041	0.0040	0.0056
64K	0.0021	0.0027	0.0027	0.0025	0.0031
128K	0.0001	0.0001	0.0001	0.0001	0.0001
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Unified

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.2202	0.1302	0.0836	0.0676	0.0707
4K	0.1978	0.1111	0.0653	0.0419	0.0355
8K	0.1756	0.0957	0.0526	0.0306	0.0206
16K	0.1282	0.0714	0.0395	0.0235	0.0160
32K	0.0945	0.0514	0.0277	0.0155	0.0104
64K	0.0516	0.0295	0.0163	0.0092	0.0061
128K	0.0078	0.0044	0.0025	0.0015	0.0010
256K	0.0072	0.0040	0.0022	0.0012	0.0007
512K	0.0072	0.0040	0.0021	0.0012	0.0007
1M	0	0	0	0	0

Fpppp : Associativity 4

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.2345	0.1184	0.0605	0.0312	0.0163
8K	0.2245	0.1132	0.0576	0.0297	0.0155
16K	0.1521	0.0770	0.0399	0.0209	0.0112
32K	0.1242	0.0622	0.0313	0.0158	0.0080
64K	0.0124	0.0063	0.0035	0.0019	0.0012
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0448	0.0404	0.0310	0.0280	0.0271
8K	0.0067	0.0074	0.0085	0.0093	0.0098
16K	0.0014	0.0014	0.0017	0.0022	0.0031
32K	0.0007	0.0007	0.0007	0.0008	0.0011
64K	0.0001	0.0001	0.0001	0.0001	0.0002
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Unified

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.1953	0.1085	0.0627	0.0387	0.0280
8K	0.1735	0.0938	0.0507	0.0286	0.0178
16K	0.1384	0.0768	0.0411	0.0230	0.0132
32K	0.0877	0.0456	0.0244	0.0140	0.0085
64K	0.0219	0.0122	0.0074	0.0047	0.0031
128K	0.0012	0.0009	0.0006	0.0005	0.0005
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Fpppp : Associativity 8

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.2231	0.1125	0.0574	0.0295	0.0155
16K	0.1583	0.0808	0.0424	0.0223	0.0118
32K	0.1240	0.0622	0.0313	0.0158	0.0080
64K	0.0041	0.0023	0.0014	0.0008	0.0005
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0042	0.0053	0.0067	0.0084	0.0080
16K	0.0013	0.0012	0.0013	0.0015	0.0022
32K	0.0006	0.0006	0.0006	0.0005	0.0006
64K	0.0001	0.0001	0.0001	0.0001	0.0001
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Unified

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.1721	0.0931	0.0505	0.0286	0.0171
16K	0.1448	0.0780	0.0410	0.0224	0.0129
32K	0.0848	0.0432	0.0226	0.0125	0.0077
64K	0.0274	0.0152	0.0087	0.0052	0.0035
128K	0.0001	0.0001	0.0001	0.0001	0.0001
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Gcc : Associativity 1

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.1535	0.0979	0.0679	0.0495	0.0371
2K	0.1288	0.0824	0.0573	0.0408	0.0304
4K	0.0960	0.0623	0.0445	0.0323	0.0241
8K	0.0630	0.0416	0.0308	0.0231	0.0175
16K	0.0374	0.0251	0.0184	0.0141	0.0109
32K	0.0192	0.0128	0.0089	0.0067	0.0052
64K	0.0104	0.0067	0.0047	0.0035	0.0028
128K	0.0059	0.0038	0.0026	0.0020	0.0015
256K	0.0021	0.0013	0.0009	0.0007	0.0005
512K	0.0008	0.0005	0.0003	0.0002	0.0002
1M	0.0002	0.0001	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.2159	0.2138	0.2468	0.2855	0.3346
2K	0.1667	0.1622	0.1799	0.2054	0.2411
4K	0.1259	0.1184	0.1300	0.1437	0.1682
8K	0.1005	0.0921	0.1005	0.1072	0.1231
16K	0.0796	0.0710	0.0770	0.0781	0.0876
32K	0.0420	0.0307	0.0256	0.0241	0.0288
64K	0.0300	0.0206	0.0158	0.0138	0.0140
128K	0.0202	0.0125	0.0085	0.0067	0.0060
256K	0.0149	0.0086	0.0055	0.0038	0.0031
512K	0.0106	0.0059	0.0034	0.0021	0.0015
1M	0.0078	0.0042	0.0023	0.0013	0.0008

Unified

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.2174	0.1738	0.1696	0.1841	0.2292
2K	0.1781	0.1393	0.1275	0.1309	0.1531
4K	0.1380	0.1060	0.0951	0.0939	0.1046
8K	0.1000	0.0774	0.0695	0.0666	0.0716
16K	0.0706	0.0549	0.0497	0.0473	0.0505
32K	0.0405	0.0301	0.0250	0.0235	0.0256
64K	0.0230	0.0169	0.0139	0.0128	0.0134
128K	0.0146	0.0100	0.0077	0.0067	0.0069
256K	0.0094	0.0064	0.0049	0.0042	0.0041
512K	0.0065	0.0044	0.0034	0.0028	0.0028
1M	0.0041	0.0028	0.0021	0.0018	0.0019

Gcc : Associativity 2

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.1172	0.0744	0.0510	0.0369	0.0268
4K	0.0869	0.0567	0.0400	0.0291	0.0215
8K	0.0519	0.0356	0.0263	0.0203	0.0153
16K	0.0254	0.0177	0.0133	0.0107	0.0086
32K	0.0121	0.0083	0.0061	0.0049	0.0040
64K	0.0062	0.0039	0.0027	0.0020	0.0015
128K	0.0035	0.0022	0.0015	0.0010	0.0008
256K	0.0013	0.0009	0.0006	0.0005	0.0003
512K	0.0003	0.0002	0.0001	0	0
1M	0.0002	0.0001	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.1383	0.1322	0.1385	0.1577	0.1885
4K	0.1072	0.0988	0.0986	0.1097	0.1281
8K	0.0851	0.0761	0.0731	0.0762	0.0856
16K	0.0693	0.0609	0.0570	0.0564	0.0602
32K	0.0328	0.0219	0.0162	0.0132	0.0127
64K	0.0230	0.0139	0.0091	0.0064	0.0050
128K	0.0164	0.0095	0.0058	0.0037	0.0026
256K	0.0126	0.0069	0.0040	0.0024	0.0015
512K	0.0096	0.0051	0.0028	0.0016	0.0009
1M	0.0073	0.0038	0.0020	0.0011	0.0006

Unified

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.1560	0.1164	0.0983	0.0920	0.0956
4K	0.1211	0.0903	0.0750	0.0684	0.0671
8K	0.0830	0.0638	0.0538	0.0491	0.0473
16K	0.0512	0.0406	0.0353	0.0331	0.0321
32K	0.0264	0.0192	0.0155	0.0137	0.0131
64K	0.0144	0.0097	0.0072	0.0060	0.0054
128K	0.0090	0.0056	0.0038	0.0028	0.0023
256K	0.0059	0.0035	0.0023	0.0016	0.0012
512K	0.0036	0.0021	0.0013	0.0008	0.0005
1M	0.0024	0.0013	0.0007	0.0005	0.0003

Gcc : Associativity 4

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0822	0.0541	0.0381	0.0280	0.0207
8K	0.0443	0.0318	0.0245	0.0190	0.0144
16K	0.0175	0.0126	0.0103	0.0090	0.0079
32K	0.0085	0.0057	0.0042	0.0034	0.0028
64K	0.0053	0.0033	0.0022	0.0016	0.0012
128K	0.0032	0.0020	0.0013	0.0009	0.0006
256K	0.0010	0.0007	0.0005	0.0004	0.0003
512K	0.0003	0.0001	0	0	0
1M	0.0002	0.0001	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0669	0.0521	0.0480	0.0560	0.0718
8K	0.0501	0.0351	0.0284	0.0290	0.0369
16K	0.0377	0.0243	0.0175	0.0146	0.0158
32K	0.0295	0.0183	0.0123	0.0088	0.0074
64K	0.0216	0.0129	0.0082	0.0053	0.0037
128K	0.0154	0.0088	0.0053	0.0033	0.0022
256K	0.0121	0.0066	0.0038	0.0022	0.0014
512K	0.0096	0.0051	0.0028	0.0015	0.0009
1M	0.0073	0.0038	0.0020	0.0011	0.0006

Unified

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.1064	0.0758	0.0598	0.0530	0.0520
8K	0.0678	0.0506	0.0409	0.0355	0.0336
16K	0.0351	0.0266	0.0223	0.0209	0.0201
32K	0.0195	0.0137	0.0108	0.0095	0.0095
64K	0.0122	0.0079	0.0056	0.0045	0.0039
128K	0.0081	0.0049	0.0031	0.0022	0.0016
256K	0.0053	0.0031	0.0020	0.0013	0.0009
512K	0.0033	0.0019	0.0011	0.0007	0.0005
1M	0.0023	0.0012	0.0007	0.0004	0.0002

Gcc : Associativity 8

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0421	0.0309	0.0246	0.0186	0.0144
16K	0.0147	0.0104	0.0085	0.0082	0.0080
32K	0.0075	0.0050	0.0036	0.0028	0.0023
64K	0.0049	0.0029	0.0019	0.0014	0.0011
128K	0.0031	0.0020	0.0013	0.0009	0.0006
256K	0.0006	0.0005	0.0004	0.0003	0.0003
512K	0.0002	0.0001	0	0	0
1M	0.0002	0.0001	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0485	0.0336	0.0266	0.0265	0.0340
16K	0.0364	0.0233	0.0166	0.0132	0.0137
32K	0.0270	0.0166	0.0108	0.0074	0.0057
64K	0.0212	0.0126	0.0079	0.0052	0.0035
128K	0.0151	0.0085	0.0051	0.0032	0.0021
256K	0.0120	0.0065	0.0037	0.0022	0.0013
512K	0.0096	0.0051	0.0028	0.0016	0.0009
1M	0.0074	0.0038	0.0020	0.0011	0.0006

Unified

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0656	0.0489	0.0390	0.0334	0.0308
16K	0.0314	0.0236	0.0204	0.0197	0.0190
32K	0.0175	0.0120	0.0092	0.0080	0.0083
64K	0.0110	0.0070	0.0049	0.0037	0.0032
128K	0.0078	0.0047	0.0029	0.0020	0.0014
256K	0.0050	0.0030	0.0019	0.0012	0.0008
512K	0.0032	0.0018	0.0010	0.0006	0.0004
1M	0.0022	0.0012	0.0007	0.0004	0.0002

Matrix300 : Associativity 1

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.0005	0.0005	0.0005	0.0010	0.0010
2K	0	0	0	0	0
4K	0	0	0	0	0
8K	0	0	0	0	0
16K	0	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.2549	0.2176	0.2045	0.2098	0.2368
2K	0.2499	0.2139	0.1968	0.1941	0.2051
4K	0.2162	0.2079	0.1909	0.1851	0.1885
8K	0.1635	0.1671	0.1870	0.1798	0.1793
16K	0.1220	0.1177	0.1426	0.1747	0.1753
32K	0.1027	0.0914	0.0970	0.1333	0.1697
64K	0.0930	0.0780	0.0763	0.0866	0.1273
128K	0.0881	0.0713	0.0658	0.0687	0.0851
256K	0.0856	0.0679	0.0606	0.0598	0.0652
512K	0.0475	0.0480	0.0490	0.0509	0.0547
1M	0.0018	0.0017	0.0021	0.0033	0.0058

Unified

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.1155	0.1097	0.1246	0.1586	0.2336
2K	0.0990	0.0893	0.0928	0.1087	0.1469
4K	0.0820	0.0784	0.0769	0.0833	0.1019
8K	0.0608	0.0614	0.0684	0.0701	0.0787
16K	0.0441	0.0426	0.0510	0.0626	0.0671
32K	0.0356	0.0320	0.0342	0.0467	0.0602
64K	0.0312	0.0265	0.0261	0.0300	0.0442
128K	0.0290	0.0236	0.0219	0.0232	0.0291
256K	0.0279	0.0222	0.0199	0.0198	0.0218
512K	0.0155	0.0156	0.0160	0.0167	0.0181
1M	0.0007	0.0006	0.0008	0.0012	0.0021

Matrix300 : Associativity 2

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
2K	0	0	0	0	0
4K	0	0	0	0	0
8K	0	0	0	0	0
16K	0	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.2494	0.2103	0.1892	0.1783	0.1733
4K	0.2268	0.2049	0.1867	0.1773	0.1729
8K	0.1574	0.1834	0.1837	0.1755	0.1717
16K	0.0934	0.0848	0.1545	0.1751	0.1713
32K	0.0836	0.0539	0.0706	0.1454	0.1711
64K	0.0833	0.0476	0.0377	0.0619	0.1414
128K	0.0832	0.0446	0.0292	0.0294	0.0563
256K	0.0832	0.0432	0.0251	0.0201	0.0261
512K	0.0686	0.0352	0.0195	0.0135	0.0143
1M	0.0015	0.0013	0.0017	0.0029	0.0055

Unified

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.0851	0.0708	0.0645	0.0636	0.0676
4K	0.0786	0.0688	0.0619	0.0591	0.0592
8K	0.0553	0.0620	0.0603	0.0575	0.0565
16K	0.0324	0.0303	0.0514	0.0570	0.0558
32K	0.0275	0.0184	0.0241	0.0478	0.0555
64K	0.0270	0.0157	0.0127	0.0208	0.0462
128K	0.0268	0.0145	0.0096	0.0098	0.0187
256K	0.0268	0.0139	0.0082	0.0066	0.0086
512K	0.0222	0.0114	0.0063	0.0044	0.0047
1M	0.0005	0.0005	0.0006	0.0010	0.0019

Matrix300 : Associativity 4

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
4K	0	0	0	0	0
8K	0	0	0	0	0
16K	0	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.2347	0.2073	0.1880	0.1779	0.1731
8K	0.1601	0.1990	0.1832	0.1751	0.1713
16K	0.0873	0.0916	0.1724	0.1750	0.1711
32K	0.0832	0.0419	0.0733	0.1598	0.1711
64K	0.0832	0.0417	0.0212	0.0670	0.1494
128K	0.0832	0.0417	0.0210	0.0109	0.0634
256K	0.0832	0.0417	0.0210	0.0106	0.0057
512K	0.0831	0.0417	0.0210	0.0106	0.0055
1M	0.0008	0.0005	0.0003	0.0003	0.0003

Unified

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0810	0.0694	0.0617	0.0580	0.0564
8K	0.0555	0.0658	0.0599	0.0570	0.0557
16K	0.0295	0.0326	0.0569	0.0568	0.0554
32K	0.0268	0.0136	0.0250	0.0525	0.0554
64K	0.0268	0.0134	0.0070	0.0223	0.0490
128K	0.0268	0.0134	0.0068	0.0036	0.0208
256K	0.0268	0.0134	0.0068	0.0034	0.0019
512K	0.0268	0.0134	0.0068	0.0034	0.0018
1M	0.0003	0.0002	0.0001	0	0

Matrix300 : Associativity 8

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
8K	0	0	0	0	0
16K	0	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.1580	0.1990	0.1831	0.1750	0.1712
16K	0.0839	0.1083	0.1831	0.1750	0.1711
32K	0.0832	0.0417	0.0880	0.1750	0.1711
64K	0.0832	0.0417	0.0210	0.0832	0.1710
128K	0.0832	0.0417	0.0210	0.0106	0.0746
256K	0.0832	0.0417	0.0210	0.0106	0.0055
512K	0.0831	0.0417	0.0210	0.0106	0.0055
1M	0.0007	0.0004	0.0002	0.0001	0.0001

Unified

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0554	0.0657	0.0598	0.0570	0.0558
16K	0.0281	0.0363	0.0597	0.0568	0.0554
32K	0.0268	0.0134	0.0292	0.0568	0.0554
64K	0.0268	0.0134	0.0068	0.0272	0.0554
128K	0.0268	0.0134	0.0068	0.0034	0.0244
256K	0.0268	0.0134	0.0068	0.0034	0.0018
512K	0.0268	0.0134	0.0068	0.0034	0.0018
1M	0.0002	0.0001	0	0	0

Nasa Kernels : Associativity 1

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.0043	0.0026	0.0016	0.0010	0.0013
2K	0.0014	0.0009	0.0007	0.0005	0.0005
4K	0	0	0.0002	0.0001	0.0001
8K	0	0	0	0	0
16K	0	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.2914	0.2812	0.2636	0.2663	0.2809
2K	0.2838	0.2715	0.2498	0.2452	0.2584
4K	0.2687	0.2565	0.2339	0.2221	0.2252
8K	0.2088	0.1965	0.1751	0.1635	0.1646
16K	0.1708	0.1578	0.1444	0.1369	0.1379
32K	0.1387	0.1181	0.1138	0.1133	0.1129
64K	0.1235	0.0996	0.0977	0.0970	0.1003
128K	0.1100	0.0870	0.0762	0.0751	0.0837
256K	0.0933	0.0703	0.0589	0.0545	0.0561
512K	0.0421	0.0322	0.0266	0.0245	0.0249
1M	0.0181	0.0120	0.0090	0.0078	0.0077

Unified

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.1331	0.1336	0.1449	0.1764	0.2474
2K	0.1151	0.1132	0.1144	0.1277	0.1633
4K	0.1021	0.0989	0.0936	0.0985	0.1164
8K	0.0772	0.0732	0.0674	0.0676	0.0768
16K	0.0624	0.0579	0.0540	0.0534	0.0584
32K	0.0503	0.0432	0.0421	0.0428	0.0454
64K	0.0443	0.0361	0.0356	0.0358	0.0385
128K	0.0379	0.0300	0.0264	0.0262	0.0296
256K	0.0320	0.0241	0.0202	0.0189	0.0196
512K	0.0145	0.0111	0.0092	0.0085	0.0088
1M	0.0062	0.0042	0.0032	0.0028	0.0028

Nasa Kernels : Associativity 2

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.0007	0.0005	0.0006	0.0006	0.0005
4K	0	0	0	0	0
8K	0	0	0	0	0
16K	0	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.2521	0.2223	0.1885	0.1838	0.1833
4K	0.2435	0.2184	0.1851	0.1672	0.1714
8K	0.1897	0.1705	0.1490	0.1349	0.1261
16K	0.1421	0.1197	0.1058	0.0961	0.0933
32K	0.1167	0.0841	0.0852	0.0844	0.0829
64K	0.1032	0.0725	0.0663	0.0708	0.0750
128K	0.0969	0.0688	0.0589	0.0580	0.0652
256K	0.0892	0.0646	0.0514	0.0447	0.0420
512K	0.0447	0.0328	0.0262	0.0228	0.0211
1M	0.0203	0.0129	0.0093	0.0074	0.0065

Unified

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.0934	0.0839	0.0744	0.0729	0.0776
4K	0.0867	0.0779	0.0672	0.0623	0.0661
8K	0.0677	0.0608	0.0529	0.0487	0.0470
16K	0.0504	0.0424	0.0371	0.0337	0.0330
32K	0.0408	0.0296	0.0296	0.0292	0.0288
64K	0.0356	0.0250	0.0229	0.0243	0.0257
128K	0.0331	0.0235	0.0202	0.0198	0.0223
256K	0.0303	0.0220	0.0175	0.0152	0.0143
512K	0.0152	0.0112	0.0089	0.0078	0.0072
1M	0.0069	0.0044	0.0031	0.0025	0.0022

Nasa Kernels : Associativity 4

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
4K	0	0	0	0	0
8K	0	0	0	0	0
16K	0	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.2207	0.2003	0.1675	0.1513	0.1548
8K	0.2001	0.1840	0.1639	0.1471	0.1373
16K	0.1435	0.1208	0.1018	0.0936	0.0914
32K	0.1186	0.0843	0.0856	0.0829	0.0806
64K	0.1019	0.0698	0.0529	0.0703	0.0717
128K	0.0973	0.0674	0.0515	0.0447	0.0643
256K	0.0906	0.0639	0.0497	0.0426	0.0401
512K	0.0497	0.0361	0.0285	0.0247	0.0228
1M	0.0217	0.0157	0.0127	0.0111	0.0104

Unified

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0789	0.0709	0.0590	0.0547	0.0570
8K	0.0709	0.0647	0.0569	0.0508	0.0481
16K	0.0503	0.0423	0.0354	0.0325	0.0313
32K	0.0409	0.0293	0.0295	0.0284	0.0276
64K	0.0348	0.0239	0.0182	0.0240	0.0245
128K	0.0331	0.0229	0.0175	0.0152	0.0219
256K	0.0308	0.0217	0.0169	0.0145	0.0136
512K	0.0170	0.0123	0.0097	0.0084	0.0078
1M	0.0074	0.0053	0.0043	0.0038	0.0035

Nasa Kernels : Associativity 8

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
8K	0	0	0	0	0
16K	0	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.1998	0.1609	0.1285	0.1046	0.0909
16K	0.1460	0.1224	0.1008	0.0914	0.0879
32K	0.1169	0.0819	0.0845	0.0773	0.0769
64K	0.0999	0.0668	0.0491	0.0673	0.0679
128K	0.0980	0.0657	0.0485	0.0394	0.0619
256K	0.0885	0.0608	0.0460	0.0379	0.0356
512K	0.0508	0.0367	0.0288	0.0249	0.0230
1M	0.0226	0.0168	0.0139	0.0125	0.0117

Unified

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0702	0.0566	0.0445	0.0361	0.0319
16K	0.0511	0.0428	0.0351	0.0318	0.0301
32K	0.0403	0.0284	0.0291	0.0264	0.0263
64K	0.0340	0.0227	0.0169	0.0230	0.0231
128K	0.0333	0.0223	0.0165	0.0134	0.0210
256K	0.0301	0.0207	0.0156	0.0129	0.0121
512K	0.0173	0.0125	0.0098	0.0085	0.0078
1M	0.0077	0.0057	0.0047	0.0042	0.0040

Spice : Associativity 1

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.0389	0.0239	0.0146	0.0107	0.0074
2K	0.0330	0.0190	0.0113	0.0068	0.0046
4K	0.0232	0.0126	0.0074	0.0045	0.0028
8K	0.0148	0.0081	0.0045	0.0027	0.0018
16K	0.0041	0.0023	0.0015	0.0008	0.0006
32K	0.0020	0.0011	0.0007	0.0004	0.0003
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.3448	0.3401	0.3415	0.3730	0.4331
2K	0.2929	0.2869	0.2792	0.2921	0.3246
4K	0.2395	0.2363	0.2287	0.2348	0.2590
8K	0.1904	0.1889	0.1837	0.1875	0.2038
16K	0.1436	0.1382	0.1336	0.1360	0.1424
32K	0.1125	0.1023	0.0958	0.0957	0.1002
64K	0.0879	0.0739	0.0648	0.0612	0.0623
128K	0.0698	0.0537	0.0434	0.0386	0.0376
256K	0.0522	0.0371	0.0275	0.0222	0.0197
512K	0.0347	0.0232	0.0164	0.0129	0.0113
1M	0.0181	0.0111	0.0070	0.0052	0.0046

Unified

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.1475	0.1300	0.1290	0.1760	0.2544
2K	0.1235	0.1057	0.0998	0.1072	0.1240
4K	0.0990	0.0852	0.0772	0.0786	0.0871
8K	0.0708	0.0620	0.0565	0.0570	0.0613
16K	0.0440	0.0405	0.0383	0.0387	0.0408
32K	0.0321	0.0285	0.0263	0.0262	0.0278
64K	0.0217	0.0185	0.0164	0.0159	0.0165
128K	0.0166	0.0130	0.0107	0.0098	0.0097
256K	0.0123	0.0089	0.0068	0.0056	0.0052
512K	0.0080	0.0054	0.0039	0.0031	0.0027
1M	0.0042	0.0026	0.0017	0.0013	0.0011

Spice : Associativity 2

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.0291	0.0156	0.0088	0.0058	0.0038
4K	0.0249	0.0134	0.0076	0.0044	0.0027
8K	0.0113	0.0065	0.0042	0.0028	0.0018
16K	0.0033	0.0019	0.0012	0.0006	0.0005
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.2773	0.2682	0.2550	0.2595	0.2862
4K	0.2231	0.2203	0.2083	0.2058	0.2156
8K	0.1678	0.1700	0.1659	0.1634	0.1660
16K	0.1238	0.1209	0.1192	0.1236	0.1266
32K	0.0937	0.0819	0.0758	0.0764	0.0794
64K	0.0749	0.0590	0.0495	0.0448	0.0441
128K	0.0608	0.0428	0.0317	0.0251	0.0214
256K	0.0479	0.0317	0.0212	0.0148	0.0108
512K	0.0315	0.0193	0.0121	0.0078	0.0051
1M	0.0164	0.0094	0.0053	0.0031	0.0018

Unified

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.1062	0.0908	0.0817	0.0883	0.0964
4K	0.0861	0.0734	0.0640	0.0606	0.0603
8K	0.0606	0.0543	0.0491	0.0465	0.0455
16K	0.0367	0.0342	0.0325	0.0327	0.0332
32K	0.0225	0.0200	0.0188	0.0190	0.0199
64K	0.0174	0.0138	0.0118	0.0108	0.0109
128K	0.0138	0.0098	0.0073	0.0058	0.0050
256K	0.0109	0.0072	0.0049	0.0034	0.0026
512K	0.0071	0.0044	0.0027	0.0018	0.0012
1M	0.0037	0.0021	0.0012	0.0007	0.0004

Spice : Associativity 4

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0234	0.0124	0.0071	0.0041	0.0026
8K	0.0079	0.0051	0.0038	0.0032	0.0021
16K	0	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.2183	0.2160	0.2024	0.1966	0.2025
8K	0.1550	0.1627	0.1608	0.1574	0.1572
16K	0.1116	0.1080	0.1102	0.1172	0.1157
32K	0.0881	0.0743	0.0678	0.0713	0.0770
64K	0.0711	0.0545	0.0440	0.0380	0.0370
128K	0.0594	0.0406	0.0292	0.0222	0.0178
256K	0.0474	0.0310	0.0201	0.0133	0.0089
512K	0.0292	0.0175	0.0108	0.0068	0.0041
1M	0.0155	0.0086	0.0047	0.0026	0.0013

Unified

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0825	0.0692	0.0596	0.0566	0.0579
8K	0.0554	0.0512	0.0465	0.0432	0.0423
16K	0.0295	0.0290	0.0294	0.0306	0.0299
32K	0.0205	0.0175	0.0163	0.0175	0.0187
64K	0.0161	0.0124	0.0101	0.0089	0.0088
128K	0.0134	0.0092	0.0066	0.0051	0.0041
256K	0.0108	0.0070	0.0046	0.0030	0.0020
512K	0.0066	0.0040	0.0025	0.0016	0.0009
1M	0.0035	0.0019	0.0011	0.0006	0.0003

Spice : Associativity 8

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0073	0.0048	0.0042	0.0035	0.0021
16K	0	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.1472	0.1609	0.1606	0.1562	0.1521
16K	0.1063	0.1006	0.1059	0.1158	0.1134
32K	0.0863	0.0712	0.0629	0.0658	0.0744
64K	0.0698	0.0534	0.0428	0.0362	0.0349
128K	0.0587	0.0396	0.0282	0.0213	0.0166
256K	0.0478	0.0312	0.0208	0.0130	0.0085
512K	0.0292	0.0174	0.0106	0.0068	0.0040
1M	0.0156	0.0086	0.0047	0.0026	0.0013

Unified

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0545	0.0514	0.0464	0.0426	0.0404
16K	0.0254	0.0254	0.0274	0.0301	0.0292
32K	0.0197	0.0164	0.0149	0.0160	0.0179
64K	0.0158	0.0121	0.0098	0.0084	0.0082
128K	0.0133	0.0090	0.0064	0.0049	0.0038
256K	0.0108	0.0071	0.0045	0.0029	0.0019
512K	0.0066	0.0039	0.0024	0.0015	0.0009
1M	0.0036	0.0020	0.0011	0.0006	0.0003

TomcatV : Associativity 1

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.0009	0.0005	0.0003	0.0002	0.0001
2K	0.0001	0.0001	0	0	0
4K	0.0001	0	0	0	0
8K	0	0	0	0	0
16K	0	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.3097	0.3183	0.3246	0.3424	0.3776
2K	0.3037	0.3120	0.3162	0.3288	0.3401
4K	0.2556	0.2781	0.2920	0.3120	0.3242
8K	0.1362	0.1175	0.1128	0.1472	0.1752
16K	0.1235	0.0780	0.0558	0.0617	0.0765
32K	0.1117	0.0711	0.0511	0.0425	0.0398
64K	0.1109	0.0703	0.0501	0.0406	0.0367
128K	0.1101	0.0697	0.0494	0.0397	0.0352
256K	0.1091	0.0691	0.0490	0.0391	0.0343
512K	0.1072	0.0681	0.0484	0.0386	0.0339
1M	0.0609	0.0306	0.0154	0.0078	0.0041

Unified

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.1500	0.1472	0.1535	0.1946	0.2568
2K	0.1317	0.1291	0.1302	0.1532	0.1792
4K	0.0941	0.1001	0.1062	0.1221	0.1374
8K	0.0532	0.0462	0.0457	0.0598	0.0751
16K	0.0453	0.0299	0.0232	0.0271	0.0355
32K	0.0393	0.0256	0.0193	0.0174	0.0184
64K	0.0377	0.0242	0.0177	0.0151	0.0147
128K	0.0368	0.0234	0.0168	0.0139	0.0129
256K	0.0361	0.0229	0.0164	0.0133	0.0119
512K	0.0353	0.0225	0.0160	0.0129	0.0114
1M	0.0201	0.0101	0.0051	0.0027	0.0015

TomcatV : Associativity 2

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.0001	0.0001	0	0	0
4K	0.0001	0	0	0	0
8K	0	0	0	0	0
16K	0	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.2429	0.2319	0.2313	0.2576	0.2774
4K	0.2401	0.2265	0.2225	0.2436	0.2596
8K	0.1461	0.1317	0.1302	0.1658	0.1945
16K	0.0939	0.0482	0.0259	0.0360	0.0600
32K	0.0791	0.0400	0.0205	0.0112	0.0074
64K	0.0749	0.0377	0.0191	0.0100	0.0058
128K	0.0743	0.0374	0.0188	0.0097	0.0052
256K	0.0734	0.0369	0.0185	0.0094	0.0049
512K	0.0715	0.0359	0.0180	0.0091	0.0046
1M	0.0610	0.0306	0.0153	0.0077	0.0039

Unified

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.0934	0.0910	0.0918	0.0991	0.1127
4K	0.0861	0.0819	0.0811	0.0873	0.0926
8K	0.0540	0.0493	0.0493	0.0609	0.0704
16K	0.0329	0.0177	0.0104	0.0149	0.0239
32K	0.0270	0.0138	0.0073	0.0044	0.0034
64K	0.0249	0.0126	0.0065	0.0035	0.0022
128K	0.0246	0.0124	0.0063	0.0033	0.0019
256K	0.0242	0.0122	0.0061	0.0031	0.0017
512K	0.0235	0.0118	0.0059	0.0030	0.0016
1M	0.0200	0.0101	0.0050	0.0025	0.0013

TomcatV : Associativity 4

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0001	0	0	0	0
8K	0	0	0	0	0
16K	0	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.1599	0.1467	0.1452	0.1757	0.2075
8K	0.1518	0.1411	0.1409	0.1670	0.1948
16K	0.0867	0.0445	0.0246	0.0264	0.0448
32K	0.0753	0.0378	0.0189	0.0095	0.0048
64K	0.0747	0.0375	0.0188	0.0094	0.0047
128K	0.0742	0.0372	0.0186	0.0093	0.0047
256K	0.0733	0.0368	0.0184	0.0092	0.0046
512K	0.0715	0.0359	0.0180	0.0090	0.0045
1M	0.0645	0.0324	0.0162	0.0081	0.0041

Unified

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0632	0.0557	0.0535	0.0668	0.0795
8K	0.0562	0.0506	0.0494	0.0595	0.0694
16K	0.0310	0.0164	0.0100	0.0126	0.0207
32K	0.0253	0.0127	0.0064	0.0032	0.0016
64K	0.0245	0.0123	0.0062	0.0031	0.0015
128K	0.0244	0.0122	0.0061	0.0031	0.0015
256K	0.0241	0.0121	0.0060	0.0030	0.0015
512K	0.0235	0.0118	0.0059	0.0029	0.0015
1M	0.0212	0.0106	0.0053	0.0027	0.0013

TomcatV : Associativity 8

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
8K	0	0	0	0	0
16K	0	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.1112	0.0560	0.0282	0.0289	0.0439
16K	0.1037	0.0521	0.0262	0.0278	0.0412
32K	0.0755	0.0380	0.0191	0.0096	0.0048
64K	0.0747	0.0375	0.0188	0.0094	0.0047
128K	0.0743	0.0373	0.0186	0.0093	0.0047
256K	0.0733	0.0368	0.0184	0.0092	0.0046
512K	0.0711	0.0357	0.0178	0.0089	0.0045
1M	0.0655	0.0329	0.0164	0.0082	0.0041

Unified

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0376	0.0189	0.0095	0.0103	0.0209
16K	0.0347	0.0175	0.0088	0.0095	0.0145
32K	0.0257	0.0129	0.0065	0.0033	0.0016
64K	0.0245	0.0123	0.0062	0.0031	0.0015
128K	0.0244	0.0122	0.0061	0.0031	0.0015
256K	0.0241	0.0121	0.0060	0.0030	0.0015
512K	0.0233	0.0117	0.0059	0.0029	0.0015
1M	0.0215	0.0108	0.0054	0.0027	0.0014

Xlisp : Associativity 1

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.1398	0.0917	0.0592	0.0421	0.0313
2K	0.0881	0.0604	0.0407	0.0289	0.0211
4K	0.0606	0.0431	0.0281	0.0213	0.0152
8K	0.0202	0.0152	0.0106	0.0074	0.0054
16K	0.0054	0.0038	0.0030	0.0024	0.0020
32K	0.0048	0.0033	0.0027	0.0021	0.0017
64K	0.0001	0.0001	0.0001	0.0001	0.0001
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.1539	0.1497	0.1702	0.1824	0.2255
2K	0.0970	0.0921	0.1027	0.1129	0.1495
4K	0.0606	0.0542	0.0541	0.0587	0.0846
8K	0.0386	0.0336	0.0328	0.0354	0.0455
16K	0.0260	0.0190	0.0170	0.0180	0.0217
32K	0.0163	0.0107	0.0087	0.0073	0.0082
64K	0.0098	0.0052	0.0029	0.0019	0.0016
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Unified

Size	Block size (bytes)				
	16	32	64	128	256
1K	0.2128	0.1696	0.1535	0.1712	0.2277
2K	0.1452	0.1181	0.1046	0.1101	0.1466
4K	0.1031	0.0835	0.0724	0.0743	0.0852
8K	0.0504	0.0423	0.0399	0.0396	0.0440
16K	0.0195	0.0148	0.0149	0.0150	0.0178
32K	0.0154	0.0113	0.0117	0.0112	0.0134
64K	0.0085	0.0063	0.0072	0.0070	0.0087
128K	0.0006	0.0004	0.0003	0.0003	0.0003
256K	0.0006	0.0004	0.0003	0.0003	0.0003
512K	0.0006	0.0004	0.0003	0.0003	0.0003
1M	0.0006	0.0004	0.0003	0.0003	0.0003

Xlisp : Associativity 2

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.0724	0.0535	0.0347	0.0276	0.0192
4K	0.0216	0.0186	0.0138	0.0116	0.0097
8K	0.0034	0.0028	0.0023	0.0019	0.0021
16K	0.0018	0.0015	0.0010	0.0009	0.0009
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.0637	0.0566	0.0662	0.0757	0.0986
4K	0.0393	0.0305	0.0304	0.0344	0.0528
8K	0.0256	0.0179	0.0144	0.0119	0.0163
16K	0.0179	0.0108	0.0079	0.0061	0.0063
32K	0.0091	0.0053	0.0034	0.0025	0.0021
64K	0.0048	0.0025	0.0014	0.0008	0.0005
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Unified

Size	Block size (bytes)				
	16	32	64	128	256
2K	0.1336	0.0998	0.0799	0.0680	0.0706
4K	0.0610	0.0496	0.0428	0.0401	0.0408
8K	0.0297	0.0239	0.0210	0.0208	0.0220
16K	0.0109	0.0082	0.0074	0.0069	0.0074
32K	0.0048	0.0033	0.0025	0.0022	0.0022
64K	0.0026	0.0016	0.0013	0.0011	0.0010
128K	0.0003	0.0002	0.0002	0.0002	0.0002
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Xlisp : Associativity 4

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0146	0.0116	0.0090	0.0091	0.0079
8K	0.0020	0.0015	0.0013	0.0012	0.0016
16K	0	0	0	0	0.0002
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0349	0.0243	0.0196	0.0212	0.0352
8K	0.0219	0.0145	0.0110	0.0084	0.0095
16K	0.0163	0.0089	0.0054	0.0035	0.0025
32K	0.0088	0.0045	0.0024	0.0012	0.0007
64K	0.0002	0.0001	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Unified

Size	Block size (bytes)				
	16	32	64	128	256
4K	0.0494	0.0427	0.0392	0.0386	0.0377
8K	0.0179	0.0142	0.0125	0.0137	0.0161
16K	0.0074	0.0052	0.0042	0.0035	0.0046
32K	0.0035	0.0020	0.0013	0.0010	0.0010
64K	0.0003	0.0002	0.0001	0.0001	0.0001
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Xlisp : Associativity 8

Instruction

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0007	0.0008	0.0010	0.0012	0.0014
16K	0	0	0	0	0
32K	0	0	0	0	0
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Data

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0206	0.0137	0.0105	0.0077	0.0066
16K	0.0161	0.0085	0.0049	0.0030	0.0024
32K	0.0090	0.0047	0.0026	0.0013	0.0007
64K	0	0	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0

Unified

Size	Block size (bytes)				
	16	32	64	128	256
8K	0.0153	0.0114	0.0098	0.0111	0.0138
16K	0.0066	0.0046	0.0036	0.0029	0.0030
32K	0.0037	0.0020	0.0012	0.0007	0.0006
64K	0.0002	0.0001	0	0	0
128K	0	0	0	0	0
256K	0	0	0	0	0
512K	0	0	0	0	0
1M	0	0	0	0	0