Optimal Design of Megabyte Second-Level Caches for Minimizing Bus Traffic in Shared-Memory Shared-Bus Multiprocessors *

Yen-Jen Oyang and Le-Chun Wu

Department of Computer Science
and Information Engineering
National Taiwan University
Taipei, Taiwan

Abstract

As the design of shared-memory shared-bus multiprocessors is heading toward employing megabyte second-level caches, how to optimize the design of the second-level caches in order to minimize the traffic on the shared bus and thus improve the system scalability is of great interest. This paper presents a comprehensive study on this issue through extensive trace-driven simulations and concludes with a few general and effective rules.

1 Introduction

In recent years, the shared-memory shared-bus architecture has become a favorite scheme for building multiprocessors. One of the major issues in the design of such systems is how to minimize the traffic a CPU would induce on the shared bus so that more CPUs can be incorporated. In this regard, a number of papers discussing how to achieve the goal with better cache design and/or cache coherence protocols have been published [1,2,3,4]. Nevertheless, a comprehensive investigation on optimal design of megabyte second-level caches for minimizing bus traffic in shared-memory shared-bus multiprocessors is yet to be carried out as employing megabyte second-level caches will soon become a common practice in computer design due to two recent developments:

1. The introducing of commodity megabit memory chips makes the implementation of megabyte caches no longer unaffordable for most systems.

2. As computer architects turn to two-level cache design to overcome the widening gap between the CPU and main memory speeds, incorporating megabyte caches at the second-level of the cache hierarchy is favorite for maximizing system performance [5,6].

Motivated by these observations, we carried out the study presented in this paper. In this study, we conducted extensive trace-driven simulations to determine the optimal configuration of the megabyte second-level cache under various system configurations. One major distinction of this study is that the width of the shared bus is taken as one major system configuration parameter and simulation runs are conducted to investigate the effect of bus width on optimal cache design. The reason behind adopting this practice is that employing a wide bus, 64 bits or more, is getting common in shared-memory shared-bus multiprocessor design. For example, the MBus adopted in a number of Sparc chip sets [7] is 64-bit

*This research was sponsored by National Science Council of R.O.C. under grant NSC 79-0408-E-002-10

0073-1129-1/92 $3.00

wide and the width of the Futurebus+ [8] can range from 32 bits to 256 bits.

When discussing multiprocessor design optimization, we must take into account how the multiprocessor will be put into use since multiprocessors installed for different purposes may run applications that have very different characteristics and behavior. In this paper, we concentrate on multiprocessors that are positioned to improve the system throughput in a multiple-user/multiple-task environment. The reason is that multiprocessors serve this kind of purposes may account for the the largest share of multiprocessor installations as of today.

The rest part of the paper is organized as follows. Section 2 discusses the methodology used in this study. Section 3 presents the simulation results and elaborates interesting observations. Section 4 concludes the study of this paper.

2 Methodology

The study is carried out through trace-driven simulations. In the simulation, a collection of 15 traces, detailed in Appendix A, generated by SPARCSim [9] are used. The effect of multitasking is simulated by having a fixed context switch interval of 16,000 memory references for all the traces. That is, each CPU is assumed to execute a section of code equivalent to 16,000 memory references from a task/process during each context switch interval. The system is assumed to maintain one global ready queue for all the CPUs and tasks/processes are scheduled based on a round-robin strategy.

Figure 1 shows the machine model used in the simulation. As one may note, the machine model has only one level of cache memory. This seems to contradict a previous argument that says the move toward employing two-level caches is a major reason behind the popularization of megabyte caches. However, as pointed out by Przybylski [10], the existence of the lower-level caches can be ignored in the study of the behavior of higher-level caches as long as the higher-level caches are several times larger than the lower-level caches. To our concern, this condition is generally true since the typical size of the first-level cache ranges from 4K Bytes to 16K Bytes while the size of the second-level caches of interest in this paper is 1 megabytes or larger. Therefore, the machine model in Figure 1 is able to accurately reflect the behavior of the local second-level caches as observed from the shared bus.

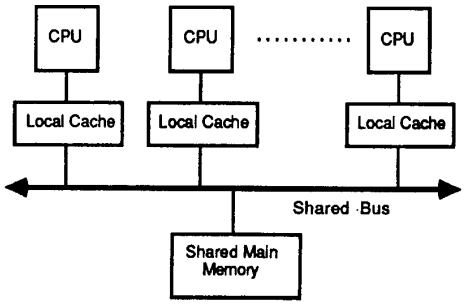

Figure 1: Machine Model Used in the Simulation.

In this study, the amount of bus traffic is measured by numbers of clock cycles. The bus is assumed to execute the Berkeley ownership protocol [11] with the following timing extracted from the MBus specification [7]:

- 2 clock cycles latency for starting a new bus transaction.
- A total of 6 clock cycles, including the 2 clock cycles for starting a new bus transaction, for executing an invalidation operation.
- 1 clock cycle for transferring a piece of data over the shared bus in the burst mode.

Furthermore, accesses to the main memory and to the cache are assumed to take 20 and 4 clock cycles, respectively.

3 Optimization of Cache Design

This section elaborates optimal design of megabyte second-level caches for minimizing bus traffic in shared-memory shared-bus multiprocessors. Tables 1 through 8 list the amount of bus traffic in numbers of clock cycles throughout the trace simulation under various system and cache configurations. The

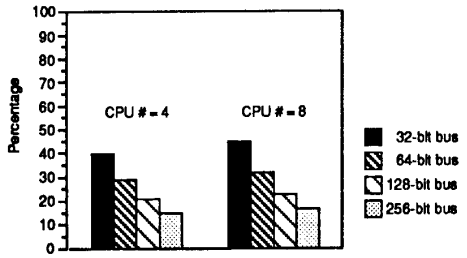

Figure 2: Bus traffic ratio between the minimum value in each of Tables 1-8 and the value in the entry of the same table corresponding to the 1-Megabyte direct-mapping cache with 16-byte blocks.

Set Associativity	Cache Size in Bytes			
	1M	2M	4M	8M
1-way				
2-way				
4-way				
8-way				

Figure 3: The region where the reducing of bus traffic due to larger cache size and higher degree of set associativity gets saturated.

first observation on the data in these tables is that a good cache design is essential for minimizing bus traffic in shared-memory shared-bus multiprocessors with megabyte second-level caches. For illustration, the ratio between the minimum bus traffic value in each table and the value in the entry of the same table corresponding to the 1-Megabyte direct-mapping cache with 16-byte blocks is plotted in Figure 2. The plot shows that a reduction of bus traffic ranging from 55% to 85% would results due to a good cache design. Moreover, the reduction of bus traffic is more significant for systems with a wider bus.

With the acknowledgement of the importance of good cache design, the next issue is to figure out, from Tables 1-8, the optimal choice of cache metrics, i.e. cache size, degree of set associativity, and block size, for various system configurations. Based on general perception, one may anticipate that the amount of bus traffic would decrease as cache size and degree of set associativity increase. This perception is confirmed by the data in Tables 1-8. However, the data in Tables 1-8 also show that the improvement gets saturated once the 2-tuple of (cache size , degree of set associativity) falls in the shaded region in Figure 3. It is interesting to observe that the same saturation region occurs no matter what the combination of block size, number of CPUs, and bus width is. This result is significant since it implies that caches with the following combinations of cache size and degree of set associativity are in general the most cost-effective choices with respect to minimizing bus traffic.

- 1 Megabyte with 8-way set associativity.

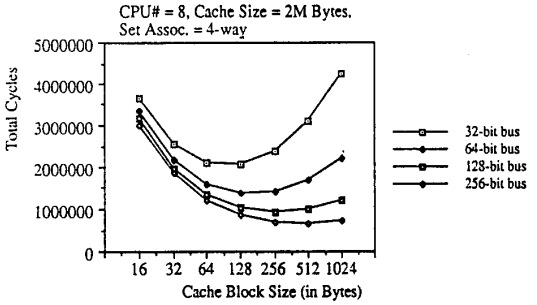

Figure 4: Bus traffic vs. block size in systems with 8 CPUs and 2-megabyte 4-way set associative caches.

- 2 Megabyte with 4-way set associativity.

- 4 Megabyte with 2-way set associativity.

The remaining cache metric yet to be discussed is the block size. Figure 4 plots the amount of bus traffic vs. block size in systems with 8 CPUs and 2-Megabyte 4-way set associative caches. The figure shows that the optimal block size is a function of bus width. For systems with a wider bus, the optimal choice of block size tends to be larger. An important observation from Tables 1-8 is that the optimal choice of block size does not depend on the CPU number, cache size, and degree of set associativity as long as the 2-tuple of (cache size , degree of set associativity) is in the saturation region as shown in Figure 3. Table 9 summarizes the optimal choice of block size for systems with an optimal choice of cache size and degree of set associativity.

Cache Block Size	Set Assoc.	Cache Size in Bytes			
		1M	2M	4M	8M
16 bytes	1-way	3965268	3285828	3087520	2949678
	2-way	3039110	2889148	2832108	2820722
	4-way	2848336	2820484	2816942	2816786
	8-way	2819184	2816744	2816744	2816744
32 bytes	1-way	3011400	2346576	2185590	2063242
	2-way	2148470	2017560	1964728	1955178
	4-way	1983266	1955714	1952096	1951946
	8-way	1955526	1951900	1951900	1951900
64 bytes	1-way	2590242	1984696	1828546	1706620
	2-way	1814944	1665922	1607158	1593880
	4-way	1628348	1596194	1590830	1590526
	8-way	1596362	1590488	1590488	1590488
128 bytes	1-way	2828552	2032332	1861822	1713204
	2-way	1863044	1670916	1594700	1575970
	4-way	1629444	1579778	1571522	1570804
	8-way	1583142	1570804	1570696	1570696
256 bytes	1-way	3429232	2540678	2310646	2096820
	2-way	2296536	2008726	1883960	1854832
	4-way	1959882	1862572	1847310	1845934
	8-way	1875210	1846106	1845590	1845590
512 bytes	1-way	5236066	3617854	3291772	2975828
	2-way	3258584	2763034	2556098	2487928
	4-way	2717898	2501536	2463180	2460148
	8-way	2540968	2462264	2459698	2459698
1024 bytes	1-way	9002772	5799182	5230136	4671908
	2-way	4857050	3981482	3650842	3507632
	4-way	4012218	3557596	3460148	3448980
	8-way	3686510	3458480	3448702	3448702

Table 1: Bus traffic in systems with 4 CPUs and a 32-bit bus.

Cache Block Size	Set Assoc.	Cache Size in Bytes			
		1M	2M	4M	8M
16 bytes	1-way	4595716	4010702	3863200	3757440
	2-way	3786122	3696442	3663954	3656904
	4-way	3665668	3656318	3655088	3655062
	8-way	3655192	3655036	3655036	3655036
32 bytes	1-way	3445384	2857406	2736876	2643242
	2-way	2675848	2594042	2563714	2557556
	4-way	2567352	2557206	2555984	2555984
	8-way	2556176	2555954	2555954	2555954
64 bytes	1-way	2928078	2396640	2277820	2184324
	2-way	2234550	2136286	2101396	2092414
	4-way	2106604	2092544	2090642	2090566
	8-way	2091076	2090528	2090528	2090528
128 bytes	1-way	3125978	2414470	2283688	2168748
	2-way	2245372	2112412	2067402	2054764
	4-way	2078402	2054506	2051830	2051598
	8-way	2053536	2051490	2051490	2051490
256 bytes	1-way	3709646	2934524	2758906	2588472
	2-way	2676362	2473632	2398874	2378916
	4-way	2432490	2379176	2373260	2372814
	8-way	2378968	2372540	2372540	2372540
512 bytes	1-way	5485160	4071132	3805144	3543716
	2-way	3629214	3275514	3150670	3102324
	4-way	3231170	3099438	3081348	3080148
	8-way	3108680	3079982	3079832	3079832
1024 bytes	1-way	9178184	6224792	5791680	5322228
	2-way	5148994	4543688	4339428	4231624
	4-way	4515872	4239670	4193544	4185466
	8-way	4290784	4185728	4185172	4185172

Table 2: Bus traffic in systems with 8 CPUs and a 32-bit bus.

Cache Block Size	Set Assoc.	Cache Size in Bytes			
		1M	2M	4M	8M
16 bytes	1-way	3629472	3000324	2816632	2689048
	2-way	2772036	2633076	2580212	2569652
	4-way	2595164	2569428	2566148	2566004
	8-way	2568224	2565964	2565964	2565964
32 bytes	1-way	2570592	1991416	1851002	1744334
	2-way	1818818	1704608	1658464	1650118
	4-way	1674542	1650590	1647432	1647302
	8-way	1650418	1647260	1647260	1647260
64 bytes	1-way	1996410	1513752	1389306	1291916
	2-way	1378528	1259722	1212686	1202104
	4-way	1229468	1203962	1199670	1199430
	8-way	1204066	1199400	1199400	1199400
128 bytes	1-way	1935368	1368684	1247182	1141044
	2-way	1247940	1111300	1056684	1043346
	4-way	1081220	1046034	1040146	1039636
	8-way	1048374	1039636	1039560	1039560
256 bytes	1-way	2097328	1531238	1384854	1248500
	2-way	1375768	1193206	1113592	1095088
	4-way	1161578	1100012	1090286	1089422
	8-way	1107882	1089530	1089206	1089206
512 bytes	1-way	2950818	2030638	1827348	1640916
	2-way	1805528	1519066	1398658	1359352
	4-way	1492106	1367328	1345100	1343348
	8-way	1389992	1344568	1343090	1343090
1024 bytes	1-way	4815636	3077774	2768184	2464932
	2-way	2567258	2092074	1911962	1834416
	4-way	2107962	1861596	1808692	1802644
	8-way	1931246	1807792	1802494	1802494

Table 3: Bus traffic in systems with 4 CPUs and a 64-bit bus.

Cache Block Size	Set Assoc.	Cache Size in Bytes			
		1M	2M	4M	8M
16 bytes	1-way	4204664	3662960	3526288	3428296
	2-way	3454972	3371812	3341636	3335052
	4-way	3343140	3334500	3333356	3333332
	8-way	3333452	3333308	3333308	3333308
32 bytes	1-way	2937640	2425546	2320308	2238534
	2-way	2267040	2195614	2169062	2163620
	4-way	2172128	2163306	2162232	2162232
	8-way	2162400	2162206	2162206	2162206
64 bytes	1-way	2251254	1828000	1733100	1658260
	2-way	1698382	1620070	1592068	1584822
	4-way	1596060	1584912	1583370	1583310
	8-way	1583716	1583280	1583280	1583280
128 bytes	1-way	2130474	1624726	1531240	1449196
	2-way	1503676	1409084	1376890	1367788
	4-way	1384466	1367546	1365606	1365438
	8-way	1366816	1365362	1365362	1365362
256 bytes	1-way	2258158	1765756	1653658	1545144
	2-way	1600938	1472352	1424762	1412004
	4-way	1445834	1412136	1408332	1408046
	8-way	1411928	1407868	1407868	1407868
512 bytes	1-way	3079400	2262364	2107992	1956772
	2-way	2006238	1801914	1729806	1701876
	4-way	1776130	1700206	1689732	1689044
	8-way	1705416	1688942	1688856	1688856
1024 bytes	1-way	4898632	3299224	3063744	2809076
	2-way	2715842	2387656	2276836	2218440
	4-way	2372384	2222774	2197768	2193402
	8-way	2250464	2193536	2193236	2193236

Table 4: Bus traffic in systems with 8 CPUs and a 64-bit bus.

Cache Block Size	Set Assoc.	Cache Size in Bytes			
		1M	2M	4M	8M
16 bytes	1-way	3461574	2857572	2681188	2558733
	2-way	2638499	2505040	2454264	2444117
	4-way	2468578	2443900	2440751	2440613
	8-way	2442744	2440574	2440574	2440574
32 bytes	1-way	2350188	1813836	1683708	1584880
	2-way	1653992	1548132	1505332	1497588
	4-way	1520180	1498028	1495100	1494980
	8-way	1497864	1494940	1494940	1494940
64 bytes	1-way	1699494	1278280	1169686	1084564
	2-way	1160320	1056622	1015450	1006216
	4-way	1030028	1007846	1004090	1003882
	8-way	1007918	1003856	1003856	1003856
128 bytes	1-way	1488776	1036860	939862	854964
	2-way	940388	831492	787676	777034
	4-way	807108	779162	774458	774052
	8-way	780990	774052	773992	773992
256 bytes	1-way	1431376	1026518	921958	824340
	2-way	915384	785446	728408	715216
	4-way	762426	718732	711774	711166
	8-way	724218	711242	711014	711014
512 bytes	1-way	1808194	1222030	1092436	973460
	2-way	1079000	897082	819938	795064
	4-way	879210	800224	786060	784948
	8-way	814504	785720	784786	784786
1024 bytes	1-way	2722068	1717070	1537208	1361444
	2-way	1422362	1147370	1042522	997808
	4-way	1155834	1013596	982964	979476
	8-way	1053614	982448	979390	979390

Table 5: Bus traffic in systems with 4 CPUs and a 128-bit bus.

Cache Block Size	Set Assoc.	Cache Size in Bytes			
		1M	2M	4M	8M
16 bytes	1-way	4009138	3489089	3357832	3263724
	2-way	3289397	3209397	3180477	3174126
	4-way	3181876	3173591	3172490	3172467
	8-way	3172582	3172444	3172444	3172444
32 bytes	1-way	2683768	2209616	2112024	2036180
	2-way	2062636	1996400	1971736	1966652
	4-way	1974516	1966356	1965356	1965356
	8-way	1965512	1965332	1965332	1965332
64 bytes	1-way	1912842	1543680	1460740	1395228
	2-way	1430298	1361962	1337404	1331026
	4-way	1340788	1331096	1329734	1329682
	8-way	1330036	1329656	1329656	1329656
128 bytes	1-way	1632722	1229854	1155016	1089420
	2-way	1132828	1057420	1031634	1024300
	4-way	1037498	1024066	1022494	1022358
	8-way	1023456	1022298	1022298	1022298
256 bytes	1-way	1532414	1181372	1101034	1023480
	2-way	1063226	971712	937706	928548
	4-way	952506	928616	925868	925662
	8-way	928408	925532	925532	925532
512 bytes	1-way	1876520	1357980	1259416	1163300
	2-way	1194750	1065114	1019374	1001652
	4-way	1048610	1000590	993924	993492
	8-way	1003784	993422	993368	993368
1024 bytes	1-way	2758856	1836440	1699776	1552500
	2-way	1499266	1309640	1245540	1211848
	4-way	1300640	1214326	1199880	1197370
	8-way	1230304	1197440	1197268	1197268

Table 6: Bus traffic in systems with 8 CPUs and a 128-bit bus.

Cache Block Size	Set Assoc.	Cache Size in Bytes			
		1M	2M	4M	8M
16 bytes	1-way	3293676	2714820	2545744	2428418
	2-way	2504962	2377004	2328316	2318582
	4-way	2341992	2318372	2315354	2315222
	8-way	2317264	2315184	2315184	2315184
32 bytes	1-way	2239986	1725046	1600061	1505153
	2-way	1571579	1469894	1428766	1421323
	4-way	1442999	1421747	1418934	1418819
	8-way	1421587	1418780	1418780	1418780
64 bytes	1-way	1551036	1160544	1059876	980888
	2-way	1051216	955072	916832	908272
	4-way	930308	909788	906300	906108
	8-way	909844	906084	906084	906084
128 bytes	1-way	1265480	870948	786202	711924
	2-way	786612	691588	653172	643878
	4-way	670052	645726	641614	641260
	8-way	647298	641260	641208	641208
256 bytes	1-way	1098400	774158	690510	612260
	2-way	685192	581566	535816	525280
	4-way	562850	528092	522518	522038
	8-way	532386	522098	521918	521918
512 bytes	1-way	1236882	817726	724980	639732
	2-way	715716	586090	530578	512920
	4-way	572762	516672	506540	505748
	8-way	526760	506296	505634	505634
1024 bytes	1-way	1675284	1036718	921720	809700
	2-way	849914	675018	607802	579504
	4-way	679770	589596	570100	567892
	8-way	614798	569776	567838	567838

Table 7: Bus traffic in systems with 4 CPUs and a 256-bit bus.

Cache Block Size	Set Assoc.	Cache Size in Bytes			
		1M	2M	4M	8M
16 bytes	1-way	3813612	3315218	3189376	3099152
	2-way	3123822	3047182	3019318	3013200
	4-way	3020612	3012682	3011624	3011602
	8-way	3011712	3011580	3011580	3011580
32 bytes	1-way	2556832	2101651	2007882	1935003
	2-way	1960434	1896793	1873073	1868168
	4-way	1875710	1867881	1866918	1866918
	8-way	1867068	1866895	1866895	1866895
64 bytes	1-way	1743636	1401520	1324560	1263712
	2-way	1296256	1232908	1210072	1204128
	4-way	1213152	1204188	1202916	1202868
	8-way	1203196	1202844	1202844	1202844
128 bytes	1-way	1383846	1032418	966904	909532
	2-way	947404	881588	859006	852556
	4-way	864014	852326	850938	850818
	8-way	851776	850766	850766	850766
256 bytes	1-way	1169542	889180	824722	762648
	2-way	794370	721392	694178	686820
	4-way	705842	686856	684636	684470
	8-way	686648	684364	684364	684364
512 bytes	1-way	1275080	905788	835128	766564
	2-way	789006	696714	664158	651540
	4-way	684850	650782	646020	645716
	8-way	652968	645662	645624	645624
1024 bytes	1-way	1688968	1105048	1017792	924212
	2-way	890978	770632	729892	708552
	4-way	764768	710102	700936	699354
	8-way	720224	699392	699284	699284

Table 8: Bus traffic in systems with 8 CPUs and a 256-bit bus.

Bus Width	32-bit	64-bit	128-bit	256-bit
Optimal Block Size	128 Bytes	128 Bytes	256 Bytes	512 Bytes

Table 9: Optimal choice of block size for systems with optimal choice of cache size and degree of set associativity.

4 Conclusion

In this paper, a comprehensive study on optimal design of megabyte second-level caches for minimizing bus traffic in shared-memory shared-bus multiprocessors is presented. This study focuses on the common multiple-user/multiple-task workload with no fine-grain or medium-grain parallel applications since the majority of multiprocessors are installed for improving system throughput under this kind of environment. The most important findings from this study are summarized in the following:

1. A good design of megabyte second-level caches could mean a 55% to 85% reduction in bus traffic.
2. The following three combinations of cache size and degree of set associativity are in general the most cost-effective choices for minimizing bus traffic: (1) 1 Megabyte with 8-way set associativity, (2) 2 Megabyte with 4-way set associativity, or (3) 4 Megabyte with 2-way set associativity.
3. The optimal choice of block size is a function of bus width. For systems with an optimal choice of cache size and degree of set associativity as listed above, the optimal block size is 128 bytes if the system has a 32-bit or 64-bit bus, 256 bytes if the system has a 128-bit bus, and 512 bytes if the system has a 256-bit bus.

Appendix A

The characteristics of the 15 Sparcsim traces used in the simulation are presented in Table 10, where

- AC is an ADA compiler.
- BISON is a Yacc-like package.
- CHESS is a chess program.
- COMPRESS is a program that performs data compression.
- CPP is a C compiler preprocessor.
- DIFF is an UNIX utility that compares two files and list their difference.
- GAS is the GNU assembler.
- GCC is the GNU C compiler.
- GO is a Chinese chess program.
- INDENT is a formatter.
- LD is a line editor.
- NM is an UNIX utility that lists the symbol table of a object file.
- QS is a quick sort program.
- TEXI2ROFF is a program that converts Tex input files into ROFF input files.
- YACC is a compiler compiler.

References

[1] J. Archibald and J.-L. Baer, "Cache Coherence Protocols: Evaluation Using a Multiprocessor Simulation Model", ACM Trans. on Computer Systems, Vol. 4, No. 4, November 1986.

[2] A. R. Karlin, M. S. Manasse, L. Rudolph, and D. D. Sleator, "Competitive Snoopy Caching", Algorithmica, 3, 1988.

[3] R. Sites and A. Agarwal, "Multiprocessor Cache Analysis Using ATUM", Proc. of the 15th Annual International Symposium on Computer Architecture, 1988.

[4] S. J. Eggers and R. H. Katz, "Evaluating the Performance of Four Snooping Cache Coherency Protocols", Proc. of the 16th Annual International Symposium on Computer Architecture, 1989.

Traces Name	Total references	Instruction fetch	Data read	Data write	Kernel space	User space
AC	2082744	1639913	251117	191714	1394	2081350
BISON	4146439	3429828	523252	193359	23394	4123045
CHESS	5680199	4704530	625760	349909	16875	5663324
COMPRESS	3318816	2703702	392463	222651	558	3318258
CPP	2962890	2604096	238196	120598	1308	2961582
DIFF	1861934	1391640	342244	128050	3711	1858223
GAS	2792623	2254186	384051	154386	26158	2766465
GCC	858296	719161	90368	48767	11350	846946
GO	2562089	2078233	348907	134949	1537	2560552
INDENT	2356008	1814327	423251	118430	1274	2354734
LD	2219329	1945869	242882	30578	2341	2216988
NM	3485622	2799800	468085	217737	1468	3484154
QS	1141261	1007580	78056	55625	3136	1138125
TEXI2ROFF	2308983	1805190	401938	101855	711	2308272
YACC	4029474	3477732	473335	78407	1053	4028421

Table 10: Characteristics of the traces used in the simulation

[5] J. Hennessy and D. Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann Publishers, San Mateo, California, 1990.

[6] H. O. Bugge, E. H. Kristiansen, and B. O. Bakka, "Trace-Driven Simulations for a Two-Level Cache Design in Open Bus Systems", Proc. of the 17th Annual International Symposium on Computer Architecture, 1990.

[7] Cypress Semiconductor, *Sparc MBus Interface Specification*, Cypress Semiconductor, Jan., 1991.

[8] E. Hahn, E. H. Kristiansen, and B. O. Bakka, "Performance Consideration in Futurebus+", Buscon East, 1989.

[9] Sun Microsystems, *SPARCSim Manual Set*, Sun Microsystems, 1989.

[10] S. Przybylski, *Cache and Memory Hierarchy Design* Morgan Kaufmann, San Mateo, California, 1990.

[11] R. Katz, S. Eggers, D. Wood, C. L. Perkins, and R. Sheldon, "Implementing a Cache Consistency Protocol", Proc. of the 12th Annual International Symposium on Computer Architecture, 1985.