Early Cost/Performance Cache Analysis of a Split MCM-Based MicroSparc CPU

Peyman Dehkordi, Karthi Ramamurthi, Donald Bouldin[1]
and Howard Davidson[2]

Abstract

Optimization of a microelectronic system is a difficult task involving a number of different disciplines. Often, an optimization in one discipline will result in a sub-optimal solution in other areas and the overall system. This paper looks into the optimization of a microelectronics system by concurrent consideration of the micro-architecture, package, and logic partitioning. This approach will attempt to identify an optimized design by helping the designer to explore the multi-dimensional solution space and evaluate the design candidates based on their system-level cost/performance. As a demonstration vehicle, we have evaluated the SUN MicroSparc CPU for possible MCM packaging based on sets of smaller dies using this approach. Cost/performance figure-of-merits are presented for various cache sizes using cost-optimized partitioning for flip-chip MCM-D packaging..

Introduction

Sub-micron IC technologies have low-latency device characteristics promoting higher clock speeds than in the past. Hence, interconnection delays are now becoming as important as the device delays. Microelectronic system designers have been taking advantage of these higher clock speeds to achieve higher performing systems by integrating as much functionality as possible into a single IC to reduce the interconnection penalties. However, the manufacturing yields of the resulting larger dies have been imposing constraints on the integration level. These constraints force the designers to perform various cost/performance trade-offs on the possible design candidates.

Advanced microelectronic packaging technology such as the multichip module (MCM) offers higher wiring density and shorter interconnect delays then the more traditional technologies such as printed circuit boards (PCBs). Conventionally, package technology is utilized by the package designer towards the end of the design cycle. Our previous work has shown that the packaging-related issues need to be considered throughout the design cycle by the system, IC, and package designers [1]. As we start to consider the packaging issues at an early stage of the design, we need to understand and evaluate the impact of various packaging choices. The appropriate evaluation should be made at the system level by considering the packaging effects at various stages of the design cycle. As a designer makes this design paradigm shift, he will be faced with a different set of issues and constraints. This paper addresses these issues and outlines the approach that we have adopted in an attempt to solve this problem.

Case Study

Today's high performance microprocessors are highly integrated to achieve the best cost/performance. The designers are frequently faced with selection and trade-offs of various architectural issues [2]. As MCMs are becoming more available, the design architects are tempted to split a highly integrated "large" monolithic die into a set of smaller dies to achieve a better cost/performance [3][4]. This requires the designer to *include* cost/performance trade-offs between the IC and the MCM in the traditional analysis. As a demonstration vehicle, we have been evaluating and re-designing an MCM version of the SUN MicroSparc CPU based on a set of smaller dies. The monolithic baseline design is a 40 MHz RISC CPU with 2K of data cache and 4K of instruction cache as shown in Figure 1. The design consists of 750K transistors and the die measures 15 mm by 15 mm.

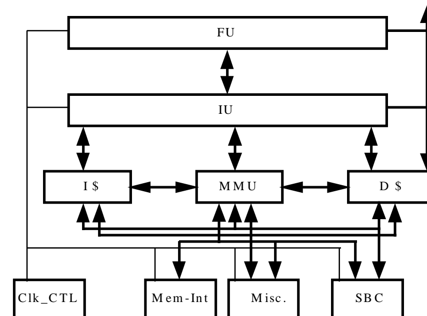


Figure 1. The MicroSparc CPU Block Diagram.

In this paper, we have extended our previous study by considering various sizes of first-level cache. We have evaluated a cost/performance figure of merit to assess the

[1] University of Tennessee, Knoxville, TN.
[2] Sun Microsystems, Mountain View, CA.
This work was sponsored in part by ARPA under grant DAAH04-94-G0004

best design candidates. Our efforts have focused on answering the following issues:

1) What is the near-optimum partition size (given a design and a packaging choice)?
2) What is the content of each partition?
3) What cache size will result in best cost/performance ?
4) Should the ICs be fabricated using a combined logic/memory process or separate logic and memory processes?
5) How should all of the above issues be considered concurrently for a near-optimum design?

We have used *early-analysis* methods to evaluate various design choices by predicting their cost/performance as shown in Figure 2. Previous works such as [5] have looked at cache-analysis for MCM-based CPUs at a conceptual stage (pre-netlist phase of the design) using estimation-based models. Other work such as [6] evaluated the impact of dielectric and bonding technologies on GaAs MCM-based CPU for a fixed partitioning. Our approach also uses estimation-based models of [7] to predict the die and the package cost and performance in conjunction with a cost-optimized partition. This approach permits us to evaluate various designs quickly and to explore the solution space for a globally optimized design. We are perfoming detailed designs to validate the estimations models.

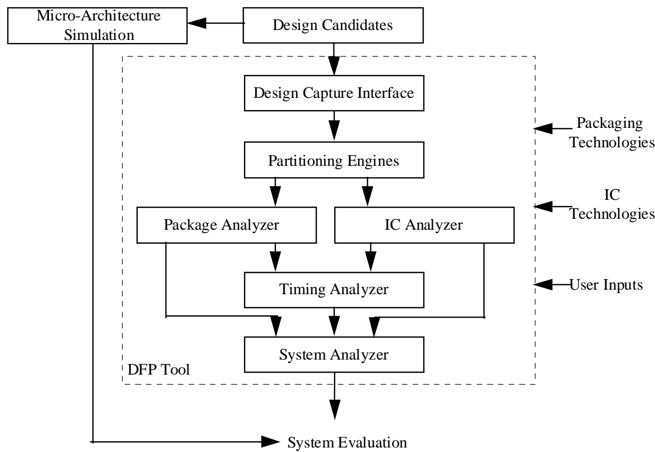


Figure 2. The Adopted Design Approach .

MCM Crossing Penalties

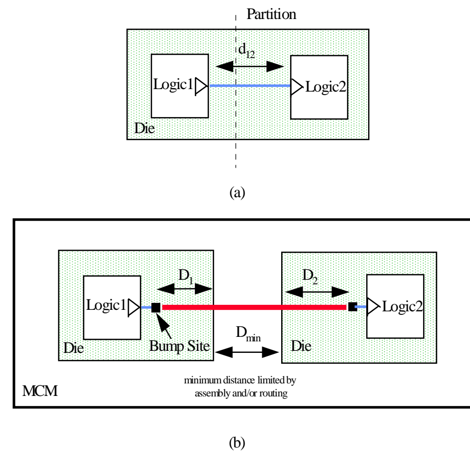
System performance is measured in terms of MIPS (Millions of Instructions per Second) and is defined as:

$$MIPS = \frac{1000}{T_C \times CPI}$$

where T_c is the cycle time (nS) and CPI is clock cycles per instruction. T_c can be impacted if the critical path is partitioned across multiple dies (as compared to the monolithic case). Figure 3a and 3b show the penalty for breaking a net and crossing the MCM substrate. The interconnections on the MCM-Ds are characterized with wider traces, less resistive and lower dielectric constant as compared to the IC technologies. It has been argued that delays on the MCM interconnections are comparable to the IC and even better for longer nets [8]. The delay model is shown in Figure 3c and is similar to the ones in [9]. In order to manage properly the testing complexity for the final MCM, we have assumed all the dies have JTAG capabilities. The JTAG I/O cells add at least one 2-1 mux on each I/O pins which adds to the MCM crossing delay. Therefore the *minimum* delay for a partitioned-net across the MCM is:

$$T_{net\text{-}breaking} = 2 * T_{jtag} + T_{buffer} + T_{mcm}$$
$$T_{net\text{-}breaking} = 2 * T_{jtag} + T_{buffer} + T_{mcm(Dmin)} + T_{mcm(D1+D2)}$$

Where D_{min} is the minimum assembly spacing between the dies and D_1 and D_2 are distance between the I/O pad and the edge of the die. The first three terms are fixed overhead (for a given MCM and IC technologies) associated with the breaking of *any* signal net and the last term is a function of the die size, placement, and the MCM technology. The $T_{net\text{-}breaking}$ delay penalty can be reduced by optimizing the I/O buffers to reduce the T_{buffer} [10]. The timing simulations in this paper are based on the I/O buffer similar to one used in [11].



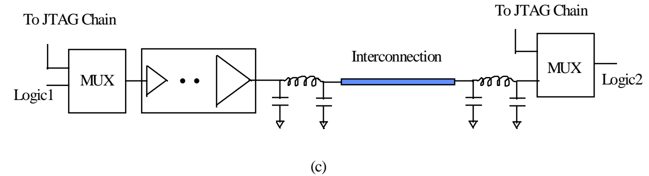


Figure 3. MCM Crossing Penalty.

Results

We have evaluated 2K/4K, 4K/8K, 8K/16K, and 16K/32K data/instruction cache sizes for the MicroSparc. We evaluated both the “combined logic/ memory” (LM) as

well as "separate logic/ memory processes" (L/M). The cost of the wafer and access time is assumed to be the same for both processes and the memory optimized process has double the density for SRAM implementation. There were a total of 8 different design candidates which were partitioned onto a MCM-D using flip-chip bonding. The partitioning stage was instructed to result in the lowest-cost set of dies for both the LM and L/M processes. For the LM process, the partitioner was allowed to mix or separate the logic and memory in order the reduce the cost (while using the LM cost model and density). For the L/M process, the partitioner was instructed to keep the logic and memory separate from each other (using both logic and memory cost models and densities). The partitioned results and the partition contents are shown in Table 1.

The results for the LM process show that the partitioner tends to separate the cache memory from the logic as the cache size becomes larger (to reduce the complete system cost: die + substrate + testing). Interestingly, this is also true for the 8K/16K and 16K/32K partitions based on the LM process in which the LM cost models were used (cache units are separated from logic). We decided not to consider 16K/32K LM because the partition result is the same as the L/M and one can always use the separate memory process for the stand-alone cache units.

The performance is measured in terms of the processor speed and CPI (cycle per instruction). The CPI measurements were provided by SUN using their trace simulator. The processor speed was calculated based on the delay which would occur if the critical path were broken. The MicroSparc CPU has a five-stage pipeline in which the critical path is between the cache and the TLB (translation lookaside buffer). We decided to take a pessimistic approach by considering every stage of the pipeline to be the critical path. We have also assumed the delay of the monolithic case is *all* due to logic (a pessimistic assumption but this is done for sake of simplicity and to offset any noise-related delays for the MCM cases). Therefore, a break in any of the pipeline stages (due to partitioning) will result in a longer machine cycle. The delay calculations are based on the estimated die-set size and their anticipated placement on the substrate. The die-set placements for all the cases are shown in Figure 3 along with the critical path representations.

The relative performance is shown in Figure 4 where all the figures of performance are normalized to the performance of the monolithic MicroSparc. Some important observations can be made from this analysis. The maximum clock frequencies in all the cases are lower than the monolithic CPU due to the substrate crossing penalty. Spice simulation shows that the worst-case delay is about 1 nS for the longest critical path on the substrate. For the 2K/4K MCM, both LM and L/M MCM-CPUs have 4% less performance (i.e. MIPS) than the monolithic design due to the substrate/buffer penalty. For all other cases, the MCM versions are higher performing by a factor of 3%, 7% and 12% for the 4k/8K, 8K/16K, and 16K/32K designs. The performance results also point out that there is not much difference between MCMs using combined and separate processes (although they resulted in different partitions).

The relative cost of the MCM CPUs are shown in Figure 5 where all the costs are normalized to the cost of the monolithic MicroSparc. All MCM-CPUs have a much lower cost than the monolithic CPU (even the one with 16K/32K cache is 24% the cost of the monolithic CPU). Interestingly, the LM MCM-CPUs show a lower cost as compared to the L/M MCM-CPUs for 2K/4K and 4K/8K cache sizes. The L/M 2K/4K and 4K/8K MCM CPUs have the same cost. This is because of the high I/O connection to the cache units as compared to its small die size. Therefore, the die size was dominated by the I/Os rather than the cache size (we used a 250 micron pitch for the flip-chip bonding). This suggests that a more aggressive flip-chip technology (i.e. less than 250 micron) should be investigated since it *may* result in a lower cost system. We have also assumed that the I/O pads can be placed throughout the area of the memory dies. This may also not be a good practice because of the soft error due C^4 α emission and regular SRAM layout. Therefore, the I/O pads may have to be placed on the perimeter of the die causing the size of the dies to be determined by the number of I/O rather than the memory cells for smaller cache memories.

Figure 6 shows the cost/performance figure-of-merit (i.e. cost per MIPS) where all the figures are normalized to the monolithic MicroSparc. The LM process shows a monotonic trend as the cache size increases. This suggests that the cost/performance gets worse for a combined logic/memory process as you increase the cache size. The L/M process reveals a curve where the minimum (i.e. lowest cost per MIPS) is at 8K/16K cache size. Interestingly the best and worst figure-of-merit is for 2K/4K and 8K/16K combined logic/memory CPUs respectively. Based on a given application requirement, one may select one of these MCM CPUs. For example, for a cost-sensitive application, the 2K/4K combined logic/memory process will be the choice. For highest performance, the 16K/32K based on separate logic/memory will be selected. For moderate cost/performance, the 8K/16K using separate logic/memory will be the choice.

Conclusions

As microelectronics packaging has been becoming the bottleneck of most high-performance systems, usage of advanced packaging such as MCMs is becoming inevitable. The packaging choices and their impact need to be considered and *evaluated* at an early stage of the design for the best system cost/performance.

The results of this work suggest that an MCM-based CPU may result in a *comparable* performance and *better* cost as compared to the monolithic CPU. Physical design of the substrate and optimized I/O drivers are the key to achieve cycle times comparable to the monolithic implementation. Signal integrity issues such as cross talk and ground bounce have to be evaluated carefully since partitioning results in more I/Os and interconnections on the substrate as shown in Figure 7.

Testing continues to be an important issue with MCMs in general. In this paper, the testing cost was modeled as a function of the I/Os. A more appropriate model will include the test coverage as well as the number of required test vectors. MCM-based split CPUs with JTAG dies may result in a better "controllability" and "observability" making testing cost *comparable* to the monolithic ones.

A separate memory process seems to result in a lower cost for *larger* cache sizes for this design. An optimized memory process results in a smaller die for a given memory size than the combined logic/memory process. As the feature size is scaled down, a small cache size will have a smaller footprint in a memory optimized process. In that case, the size of the memory die may be driven by the number of I/Os instead of the memory size.

Acknowledgments

The MSDA (Multichip System Design Advisor) tool has been extensively used throughout this work. The authors gratefully acknowledge the contribution of the MSDA tool which was originally developed by Peter Sandborn at MCC and is now being commercialized through Savantage, Inc.

References

[1] Dehkordi, P., Ramamurthi, K., Bouldin, D., Davidson, H. and P. Sandborn, "Impact of Packaging Technology on System Partitioning: A Case Study", *Proceedings of the MCM Conference*, March, 1995.

[2] Hennessy, J. L. and D. Patterson, *Computer Architecture, A Quantitative Approach*, Morgan Kaufmann Publisher, Inc., 1990.

[3] Mendelsohn, A., "Will Monolithic or Multichip processor Win The Performance Race?", *Computer Design*, May, 1991.

[4] Sechler, R., "RISC System/6000 Central Processing Unit (CPU) MCM", *Proceedings of International Conference and Exhibition Multichip Modules*, Denver, CO, April, 1993.

[5] Roberts, J.D. and W. Dai, "Early System Analysis of Cache Performance for RISC Systems: MCM Design Trade-offs", *Technical Report UCSC-CRL-92-02*, University of California, Santa Cruz, March, 1992.

[6] Kayssi, A., Sakallah, K., Brown, R., Lomax, R., Mudge, T., and T. Huff, "Impact of MCM's on System Performance", *Proceedings of Multichip Module Workshop*, Santa Cruz, CA, March, 1991.

[7] Sandborn, P. and H. Moreno, *Conceptual Design of Multichip Modules and Systems*, Kluwer Academic Publishers, 1994.

[8] Franzon, P.,Stanaski, A., Tekman, Y., and S. Banerjai, "System Design Optimization for MCM", *Proceedings of Multichip Module Conference*, Santa Cruz, CA, January, 1995.

[9] Bakoglu, H.B., *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley Publishing Company, Inc., 1990.

[10] Gabara, T., Fisher, W., Knauer, S., Frye, R., Tai, K., and M. Lau, "An I/O CMOS Buffer Set for Silicon Multi Chip modules (MCM)", *Proceedings of Multichip Module Conference*, Santa Cruz, CA, March, 1993.

[11] Taylor, M, Helman, D. and W. Dai, "AMP, A Bus-based Packet Switch Design Using MCM", *Proceedings of Multichip Module Workshop*, Santa Cruz, CA, March, 1991.

Cache Size	Logic/memory combined process (LM)		Logic/memory separate process (L/M)	
	Contents	Perf. Cost Cost/Perf.	Contents	Perf. Cost Cost/Perf.
2K/4K D$/I$	1 die **(monolithic)**: (D$, I$, MMU, FU, IU, MEM-INT, SBC, CLK, MISC)	1.000 MIPS 1.000 $ 1.000 $/MIPS	N/A	
2K/4K D$/I$	3 dies: D$, I$, MMU FU, IU MEM-INT, SBC, CLK, MISC	0.961 MIPS 0.144 $ 0.150 $/MIPS	3 dies: D$, I$ FU, IU, MMU MEM-INT, SBC, CLK, MISC	0.960 MIPS 0.204 $ 0.212 $/MIPS
4K/8K D$/I$	3 dies: D$, I$, MMU FU, IU MEM-INT, SBC, CLK, MISC	1.025 MIPS 0.178 $ 0.173 $/MIPS	3 dies: D$, I$ FU, IU, MMU MEM-INT, SBC, CLK, MISC	1.024 MIPS 0.204 $ 0.199 $/MIPS
8K/16K D$/I$	4 dies: D$ I$ MMU, FU, IU MEM-INT, SBC, CLK, MISC	1.069 MIPS 0.238 $ 0.223 $/MIPS	3 dies: D$, I$ FU, IU, MMU MEM-INT, SBC, CLK, MISC	1.071 MIPS 0.213 $ 0.198 $/MIPS
16K/32K D$/I$	4 dies: D$ I$ MMU, FU, IU MEM-INT, SBC, CLK, MISC	N/A	4 dies: D$ I$ FU, IU, MMU MEM-INT, SBC, CLK, MISC	1.115 MIPS 0.238 $ 0.214 $/MIPS

Table 1. The Results of The Partitioned CPUs (All Values Are Normalized to the Monolithic CPU).

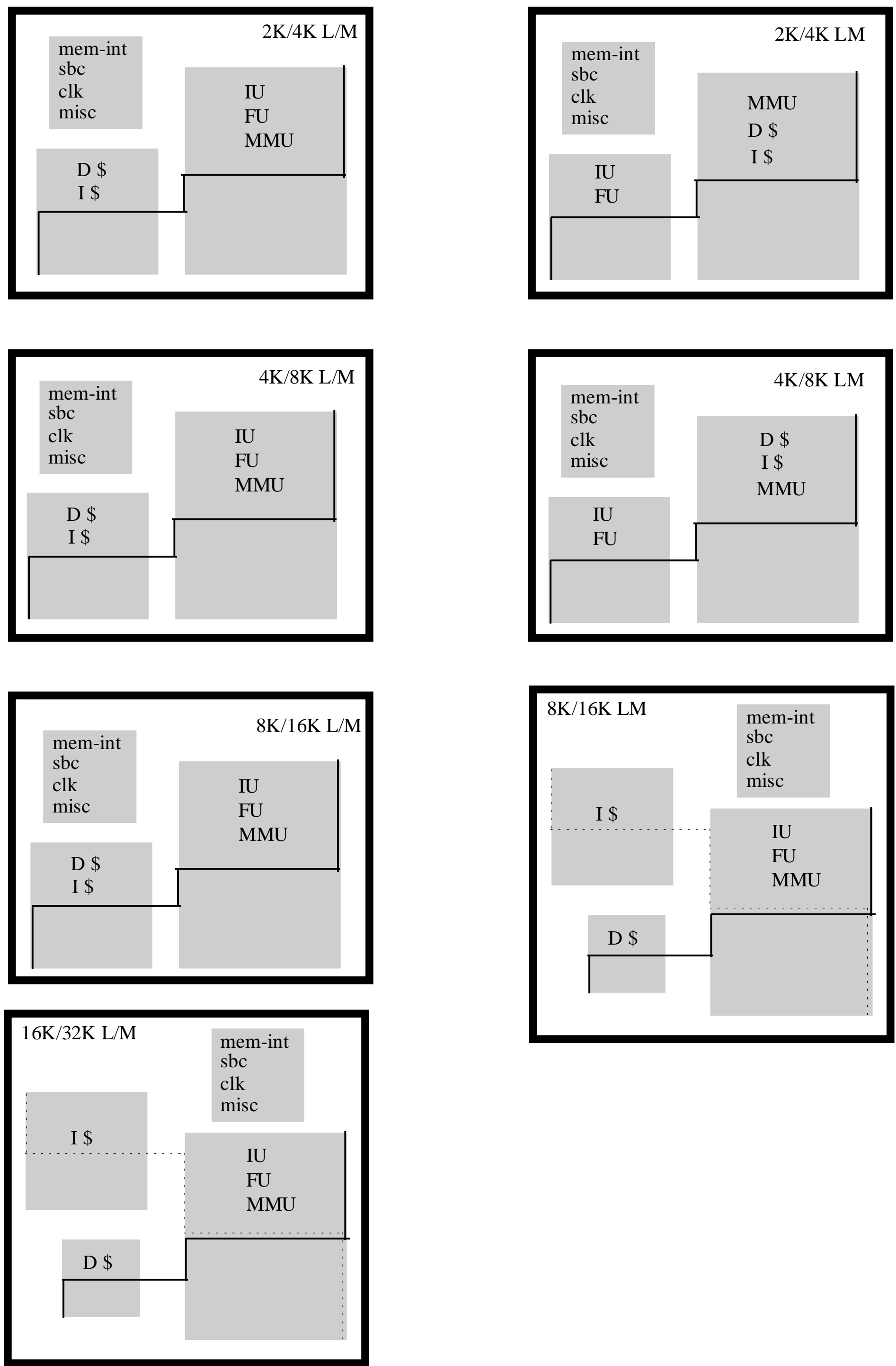


Figure 3. The Tentative Die-Set Placement on the MCM Substrate (Not to Scale).

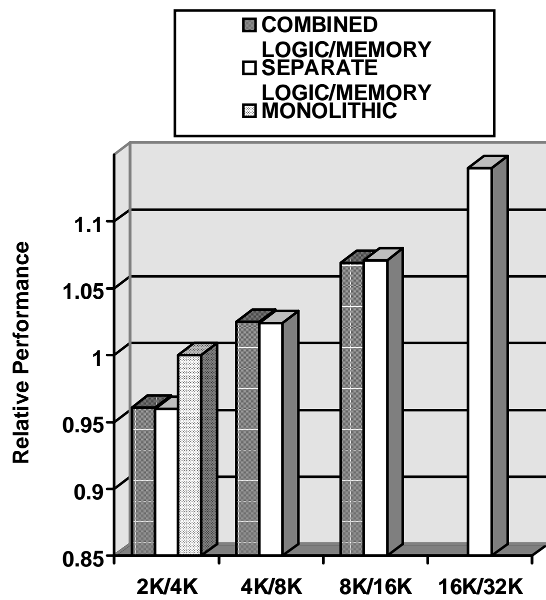


Figure 4. The Relative Performance Analysis.

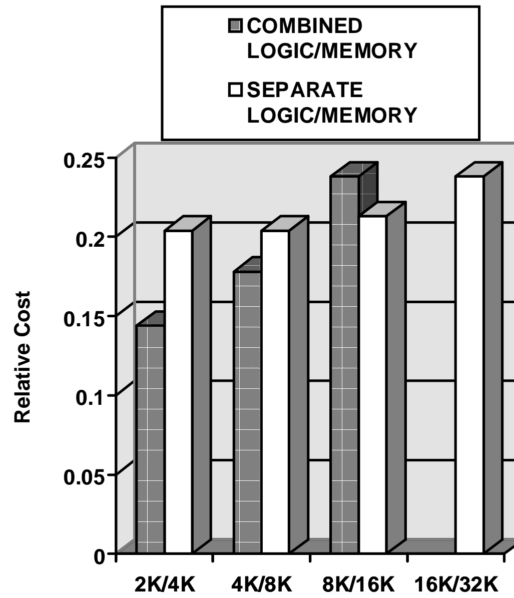


Figure 5. The Relative Cost Analysis.

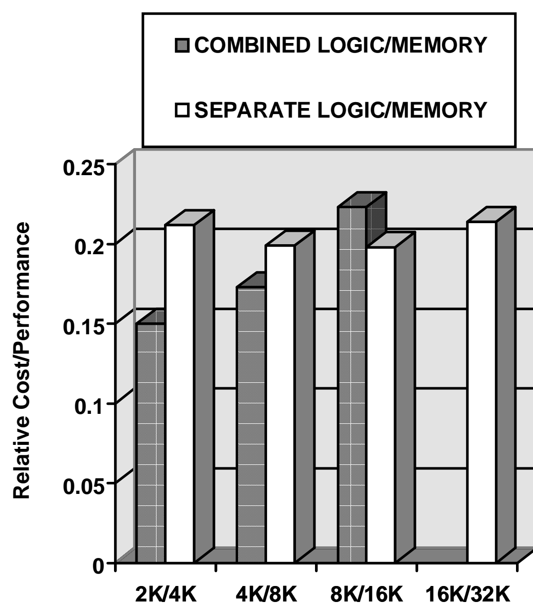


Figure 6. The Relative Cost/Performance Analysis.

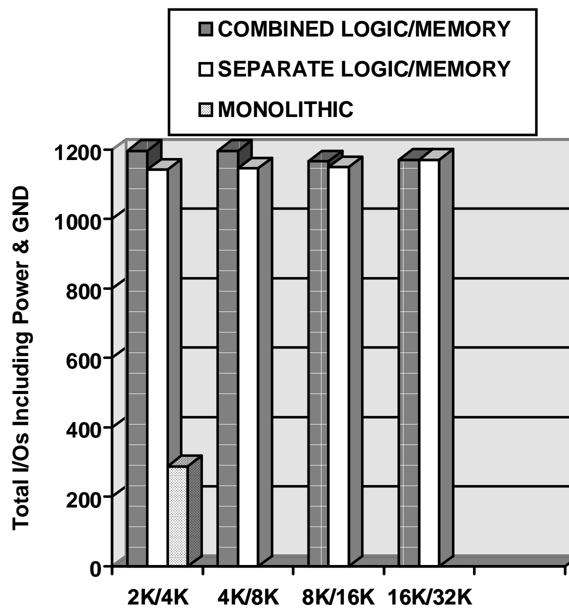


Figure 7. The Total Number of I/Os Including Power & Ground..

PERFORMANCE CALCULATIONS WORK-SHEET

CPI Computation

$CPI = CPI_{execution} + Misses / Instruction * Miss\ Penalty$

$CPI = CPI_{execution} + Memory\ access / Instruction * Miss\ rate * Miss\ Penalty$

$Average\ Memory\text{-}access\ time = Hit\ time + Miss\ rate * Miss\ Penalty$

Based on the information provided on the "Tsunami Specification" and trace results from SUN:

Data Cache	Instruction Cache	Average Miss rate	Average Memory cycle	Overall CPI
2 K	4 K	6.5 %	33 nS	2.07
4 K	8 K	4.8 %	29.6 nS	1.92
8 K	16 K	3.6 %	27.2 nS	1.82
16 K	32 K	2.6 %	25.2 nS	1.73

[1] Computation is based on 50 % of all the memory reference are to data and 50% to instructions.
[2] Miss penalty = (11 + 5) / 2 = 8 cycles first term is for I$ and second term is for D$
[3] Memory access / Instruction = 1.1
[4] $CPI_{execution} = 1.5$
[5] Hit time = 1 cycle = 25 nS , 40 MHz.
[M]aximum possible improvement due to the cache increase is 28%.

Cycle Time Computation

The critical path is most probably during the "execute" or "Result" stages of the pipeline:
TLB unit ---> cache ---> IU/FU

For the base design (2k/4k), the cycle time is 25 nS for the 40 MHz. MicroSparc. As we increase the cache size, we assume memory access times varies. We assume an increase of 0.25 nS in the memory access time for each additional 6K due to the increase in the size of the memory. We also assume that the cycle time is not impacted much due to the extra logic for larger cache (since we have a direct map cache). Therefore, the effective cycle time is computed as follows:

$T_{new\ design} = T_{base\ design} + T_{extra\ cache} + T_{jtag} + \sum$ (breaks in critical path per pipe stage * $T_{buffer + mcm}$)
$T_{new\ design} =$ 25 nS+ 0.25 * (D$+I$ - 6K) + 0.5 nS+ $\sum$ (breaks in critical path per pipe stage * $T_{buffer + mcm}$)

Cache Size D$/I$	Cost ($)	Max. Net-Length (mm) and delay (nS)		# I/O	Cycle Time (nS)
2K/4K LM	57.45	21.3	0.53	1197	26.03
4K/8K LM	71.08	23.9	0.55	1197	26.30
8K/16K LM	95.34	28.2	0.59	1167	26.59
16K/32K LM	136.15	n/a	n/a	1171	n/a
2K/4K L/M	81.48	23.5	0.54	1142	26.04
4K/8K L/M	81.48	23.5	0.54	1146	26.31
8K/16K L/M	85.02	25.1	0.56	1150	26.56
16K/32K L/M	95.34	28.2	0.59	1171	26.84

The overall performance is computed as:

$$MIPS = \frac{1000}{T_C \times CPI}$$

Cache Size D$/I$	Cycle Time (nS)	CPI	MIPS	Cost ($)	Cost/MIPS
2K/4K MONO	25.00	2.07	19.32	400.05	20.71
2K/4K LM	26.03	2.07	18.56	57.45	3.10
4K/8K LM	26.30	1.92	19.80	71.08	3.59
8K/16K LM	26.59	1.82	20.66	95.34	4.61
16K/32K LM	n/a	n/a	n/a	n/a	n/a
2K/4K L/M	26.04	2.07	18.55	81.48	4.39
4K/8K L/M	26.31	1.92	19.79	81.48	4.12
8K/16K L/M	26.56	1.82	20.69	85.02	4.10
16K/32K L/M	26.84	1.73	21.54	95.34	4.43