Access Time and Energy Tradeoffs for Caches in High Frequency Microprocessors[*]

Eugene B. John and Stefan Petko
Electrical and Computer Engineering Dept
The University of Texas at San Antonio
San Antonio, TX 78249
ejohn@utsa.edu

Lizy Kurian John and Jason Law
Electrical and Computer Engineering Dept
The University of Texas at Austin
Austin, TX 78712
{ljohn,law}@ece.utexas.edu

Abstract

This paper investigates the cache sizes and configurations that can be supported by a high frequency processor of the next generation. Based on the SIA roadmap prediction that a 0.1u processor of the next generation will run at 3.5GHz, we model caches of that technology using the CACTI tool. Access times as well as energy consumption are modeled for caches in the 8k-4M range, for various associativities. Impact of having multiple ports as well as that of varying block sizes is also studied.

1. Introduction

Advances in integrated circuit techniques have provided huge increases in the number of transistors on microprocessor chips. This increase has allowed computer architects and microprocessor designers to incorporate large amounts of on chip caches. As an example, consider the Pentium 4, which contains 42-55 million transistors. The 42-million version of the chip includes a 256KB L2 cache and the 55 million chip includes a 512KB L2 cache. This may be contrasted with early microprocessors such as the Motorola MC68020 which contained 256 bytes of cache.

While transistors are plentiful and we are able to include large caches on the microprocessors, their sizes and configurations have tremendous implications. It can take several cycles to access large caches, while small caches can be accessed relatively quickly. Such tradeoffs become even more important with the increasing clock frequency of microprocessors.

A next generation microprocessor produced using a 0.1 micron technology will run at 3.5GHz according to projections from the Semiconductor Industry Association (SIA) [1]. Single cycle access for such a microprocessor means that the access has to occur in 0.2857 ns. How big can on-chip caches be for such a microprocessor? What associativities can such caches have? Most state-of-the-art microprocessors execute multiple instructions per cycle in a superscalar manner. Issuing and executing multiple instructions necessitate multiple ports for caches. What is the impact of multiporting on cache access time? One of the objectives of this paper is to identify access time considerations for caches in a high performance, high frequency microprocessor.

Power and energy consumption have become critical design metrics in the recent past. The power consumption of on-chip caches for StrongARM SA110 is 43% of the total chip power [2]. In the 300MHz bipolar CPU reported by Jouppi et al. [3], 50% of power is dissipated by caches. How does cache power consumption vary as cache sizes and configurations change? What is the impact of associativity and multiple ports on power? Investigating this issue is another objective of the paper.

A technique to reduce conflict misses and improve cache hit ratio without increasing associativity is to use a cache assist such as the victim cache [3] or annex cache [4]. In addition to improving performance, such cache assists have been shown to reduce energy consumption. But most studies in the past assume a 1 cycle access latency for the cache assists, which are often fully associative or highly associative. Are such assumptions valid? What are reasonable cycle time assumptions to make in cache simulation studies? This paper investigates this issue as well.

The rest of this paper is organized as follows. Section 2 describes background and related research. Section 3 describes the experimental environment. Section 4 presents experimental results and tradeoff analysis. Section 5 presents a summary of the paper and offers concluding remarks.

2. Background and Motivation

Caches may be direct mapped or associative. Direct-mapped caches have more conflict misses due to their lack of associativity, and it is generally perceived that they have better access times than their associative counter parts. However, quantitative information on the access time increase due to increased associativity is not available.

Cache assists such as the victim cache [3] and the annex cache [4] are small fully associative caches used in conjunction with directly mapped L1 caches to improve hit ratios. In general, smaller direct-mapped cache benefits the most from the addition of the cache assist. As the direct-mapped cache increases in size, the relative size of the cache assist becomes smaller. This is because the

[*] This research is supported in part by a State of Texas Advanced Technology Program grant and a UTSA Summer research grant.

likelihood of a mapping conflict which would be easily removed by the assist cache is reduced. Systems with victim caches can benefit from longer line sizes more than systems without victim caches, since the victim caches help remove misses caused by conflicts that result from longer cache lines.

Bahar et al. [2] found that the time required by the processor to perform the swapping, due to a victim cache hit, was detrimental to performance. John et al [4,10] proposed the annex cache which does not perform the swapping on every assist cache hit. Instead only items that have proved their likelihood of reuse get swapped.
Bahar et al. [2] analyze the power/performance tradeoffs in microprocessors, however, assuming a single cycle access to the on-chip caches. We consider the access times and clock frequencies of the processor to determine the actual number of cycles that are needed to access caches of specific sizes in high frequency processors.

Albera et al. [5] present the advantages of using a victim buffer, albeit assuming a single cycle access to the victim cache. Various past researchers assume that a 16 block cache assist can be accessed in a single cycle. While it might be true in a low frequency processor, it might not be a reasonable assumption to make in a fast processor.

3. Experimental Methodology

3.1. CACTI

We utilized CACTI, a tool developed by DECWRL [6] to evaluate cache structures. The cache timing model was originally developed by Wada et al [7]. The CACTI tool was developed by Wilton and Jouppi [8] and later augmented by Reinman and Jouppi [9] to include power. The model was originally validated using SPICE simulations for a 0.8 um process. If a different feature size is used, the transistor capacitances needed are recomputed. The CACTI 3.0 version that we used is the latest upgrade over Reinman and Jouppi's [9] tool.

An m-way set associative cache consists of three main parts: a data array, a tag array and the necessary control logic. The data array consists of S rows containing m lines. Each line contains L bytes of data and a tag T which is used to uniquely identify the data. Upon receiving a data reference, the address is divided into three parts. The first part indexes one row in the cache, the second selects the bytes or words desired, and the last is compared to the entry in the tag to detect a hit or a miss. In a cache of size C, the number of sets (S= $\frac{C}{BxA}$) , where B is block size (in bytes), A is the associativity. These organizations could result in an array that is much larger in one direction than the other, causing either the bitlines or wordlines to be slow. To alleviate this problem, sub arraying is used. Parameters N_{dwl} and N_{dbl} are defined to describe how the array can be broken horizontally and vertically. The parameter N_{dwl} indicates how many times the array has been split with vertical cut lines, while N_{dbl} indicates how many times the array has been split with horizontal cut lines. The total number of sub arrays is N_{dwl} x N_{dbl}

Energy dissipation in CMOS technology circuit is mainly from charging and discharging gate capacitances, that is, every transition dissipate $E_q = \frac{1}{2} \cdot C_{eq} \cdot Vdd^2$ Watts. CACTI use the technology feature size as an input parameter. Aside from being used to scale the access time reported by CACTI, this parameter scales the capacitances and the value of V_{DD} used by the power model. The value of V_{DD} is scaled by

$$V_{DD} = \frac{4.5V}{(0.8/TECH)^{.67}}$$

Where TECH is the feature size of the technology. This means that voltage will scale at a slower rate than capacitance and therefore than access time. The voltage level to which the bitlines are charged is calculated as a fraction of the scaled value of V_{DD}. The 0.8 um process uses 4.5V and once scaled using the above equation, the 0.1 um model uses 1.1V.

3.2 Simulation Cache Model

Using CACTI, we have investigated the access time and power consumption for direct-mapped and associative caches. Original CACTI uses subarraying and finds optimal N-values for best access time and power consumption. The results in this paper are with the default optimizations, targeted at simultaneously optimizing for both access time and power.

We investigate the access times and power for 0.1 um technology. Caches of sizes ranging from 8k bytes to 4M bytes are studied. Direct mapped, set associative and full associative caches with single and multiple ports are studied. The block size is 32 bytes unless otherwise indicated, however, in studies on block size, 16, 32, 64 and 128 bytes per block are studied.

4. Simulation Results and Analysis

Figure 1 indicates the access times and energy per access for various cache sizes. As mentioned before, a next generation microprocessor produced using a 0.1 micron technology will run at 3.5GHz according to projections from the Semiconductor Industry Association (SIA) [1]. Single cycle access for such a microprocessor means that the access has to occur in 0.2857 ns. The x-axis indicates access time, in ns on the left side and cycles for a 3.5GHz processor on the right side. It can easily be seen that none of these caches can be accessed in a single cycle. Single port caches up to 128k or dual port caches smaller than 64k can be accessed in 4 processor cycles, while larger caches and caches with multiple ports will take very large number of cycles to access. Thus even if on chip transistor densities allow the creation of huge caches, one needs to restrict sizes and number of ports to small values in order to provide fast access.

Typically L2 caches on modern processors are accessible in 5-10 cycles and main memories in 50-100 cycles. Looking at figure 1, we can see that L2 caches cannot be much bigger than the current sizes such as 256k or 1M, even if transistors are plentiful. A single ported 4M memory will take approximately 30 cycles to access prohibiting such large L2 caches on ultra high frequency caches.

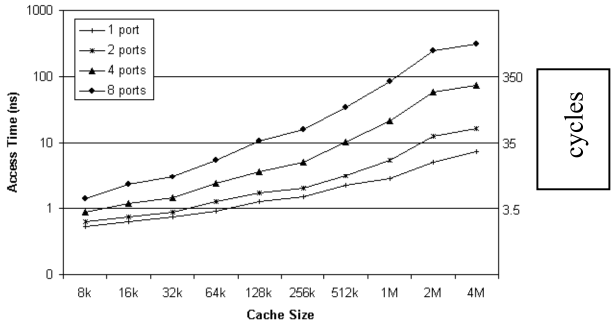

Figure 1 *Access times per access for various cache sizes. The number of cycles corresponding to a 3.5 GHz processor is indicated on the right side.*

Figure 2 illustrates growth in access times with increase in cache associativity. Increase in associativity is accompanied by increases in access times. These increases are more pronounced for small caches, however, for larger caches, associativity up to 16-way does not make any significant differences in access times. The variations in access times are negligible because the optimizations that CACTI do adjust the sub arrays in such a way that in most caches, a nearly square structure is obtained. In a naïve implementation, this may not be the case. It is extremely important to do cache array optimizations in order to get cache structures with feasible access times.

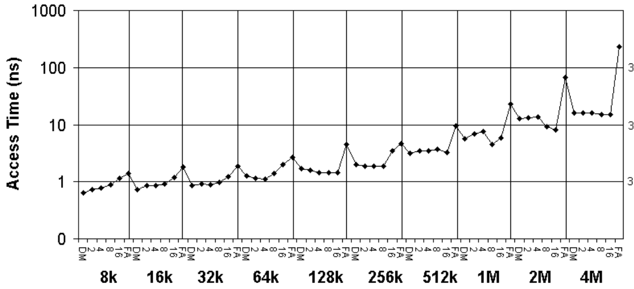

Figure 2 *Access times for various cache associativities, in caches of size 8k-4M. All caches are single ported. For each cache size, the 6 points plotted are direct mapped, 2-way, 4-way, 8-way, 16-way and fully associative.*

Figure 3 illustrates the variation in access times with increase in number of ports. It is seen that increase in ports is very expensive. The rate of growth in access times is very significant compared to impact of change in associativity. Doubling the ports increases the access tie in a manner similar to the impact of doubling the cache size.

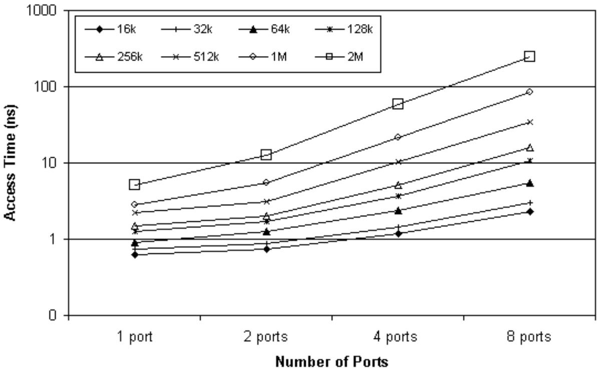

Figure 3 *Access time increase with increase in ports*

Figure 4 illustrates the energy consumption for single element access in caches of sizes in the 16k-2M range. In L1 caches, there are one or more accesses in every cycle. In L2 caches and main memories, the accesses are more infrequent, since they are accessed only when the upper layer caches result in a miss.

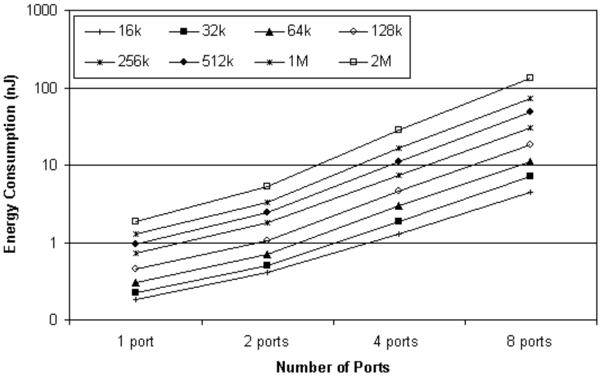

Figure 4 *Cache energy for single element access*

Figure 5 gives light on access times for assist caches such as victim caches. Assist caches up to 4k size can be accessed in 4 cycles on a 3.5GHz processor. Comparing access times in figure 5 and figure 2, it can be seen that their access times are comparable to that of direct mapped caches up to 64k. While small fully associative caches are faster than big direct mapped caches bigger than 256k, the performance improvement obtained for such caches from victim caching is very small [4].

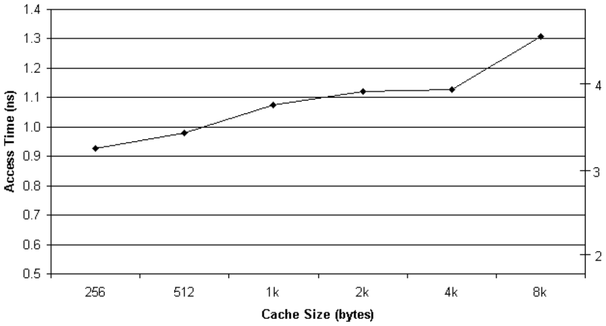

Figure 5. *Access times for small fully associative caches typically used as assist caches. Cycles for a 3.5 GHz processor are indicated on the right.*

In all of the aforementioned experimental configurations, the block size of the cache is fixed at 32 bytes. Figure 6 investigates the impact of block sizes on cache access times and energy consumption. It can be observed that block size does not have any significant impact on access time or power. The number of blocks changes with the change in block size, and in a naïve implementation, can change length of word lines and bit lines. However, optimizations can be done to maintain an approximately square structure. Hence cache size alone is seen to be the major determinant of access time and power, once adequate optimizations are performed.

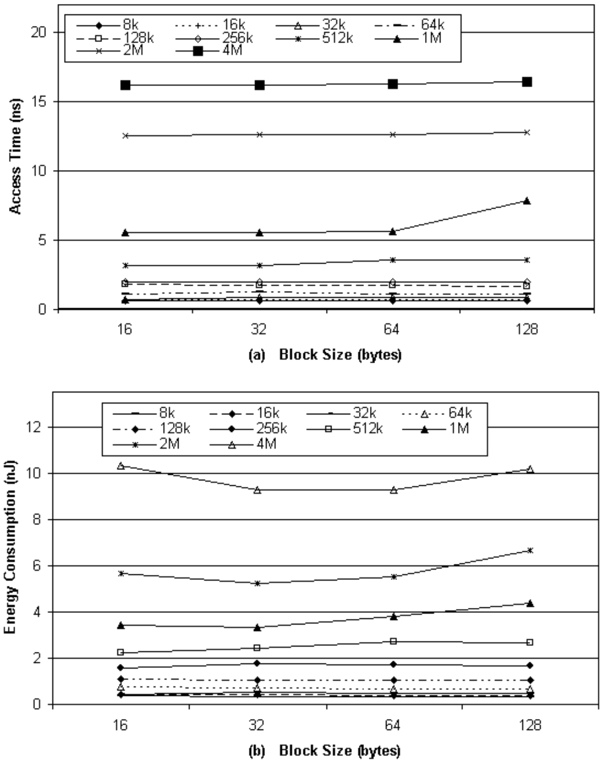

Figure 6. *Impact of block size on cache access times and energy consumption*

5. Conclusion

This paper investigates design tradeoffs for caches in future high frequency microprocessors using an access time and power model implemented in a tool called CACTI. While future processor chips will have an abundance of transistors available for on-chip caches, these caches will need to be kept small in order to guarantee fast access. For instance, single cycle cache access will be impossible in a 3.5GHz processor. Caches in the 8k-128k range would need approximately 4 processor cycles for a hit access. Associativity and block size are seen to have smaller effect on access time and power, whereas increase in ports has a significant impact. Cache size is seen to be the major determinant of access time and power, along with port numbers.

References

1. Semiconductor Industry Association, The national technology roadmap for semiconductors, 2001, www.semichips.org
2. R. I. Bahar, G. Albera, S. Manne, "Power and Performance tradeoffs using Various Caching Strategies," In Proc. Of the 1998 International Symposium on Low Power Electronics and Design, pp. 64-69, Aug. 1998
3. N. Jouppi, "Improving Direct-mapped Cache Performance by the Addition of a Small Fully-Associative Cache and Prefetch Buffers," ISCA-17: ACM/IEEE International Symposium on Computer Architecture, pp. 364-373, May 1990
4. L. K. John, T. Li and A. Subramanian, "Annex Cache: A cache assist to implement selective caching", Microprocessors and Microsystems, 23 (1999), 537-551.
5. G. Albera and I. Bahar, "Power/Performance Advantages of Victim Buffer in High-Performance Processors", International Symposium on Low Power Electronics and Design, Monterey, CA, August 1998.
6. P. Shivakumar and N. Jouppi, "CACTI 3.0: An Integrated Cache Timing Power, and Area Model", DEC Western research Lab Report 2001/2
7. T. Wada, S. Rajan and S. A. Przbylski, "An Analytical Access Time Model for On-Chip Cache Memories", IEEE Journal of Solid State Circuits, Vol. 27, No. 8, Aug 1992, p. 1147-1156
8. S. J. Wilton, N. P. Jouppi,, "CACTI: an enhanced cache access and cycle time model Solid-State Circuits," IEEE Journal of, Volume: 31 Issue: 5, pp. 677-688 May 1996
9. G. Reinman and N. Jouppi, An integrated cache timing and power model, 1999, COMPAQ Western research Lab.
10. L. John and A. Subramanian, "Design and Performance Evaluation of a Cache Assist to Implement Selective Caching", Proceedings of the International Conference on Computer Design, 1997, pp. 510-518.
11. A. Chandrakasan, W. J. Bowhill, F. Fox, Design of High Performance Microprocessor Circuits, IEEE Press, 2001