Analysis of Power Consumption in Memory Hierarchies

Patrick Hicks, Matthew Walnock, Robert Michael Owens
Pennsylvania State University

ABSTRACT

In this paper, we note and analyze a key trade-off: as the complexity of caches increases (higher set-associativity, larger block size, and larger overall size), the power consumed by a cache access increases. However, because the hit rate also increases, the number of main memory accesses decreases and thus the power consumed by a memory access decreases. Recent papers which consider the power consumption of caches tend to ignore hit rates. This is unfortunate, because it is undesirable to have energy-efficient caches which are also very slow. Hit rates also play a key role in truly evaluating the energy efficiency of a cache, because low hit rates lead to more frequent main memory accesses which consume more power than cache accesses.

1 INTRODUCTION

With the advent of mobile computing and communications, the power consumption of microprocessors becomes an increasingly important issue. In particular, current research indicates that power consumed during memory accesses accounts for a significant percentage of the total power consumption in microprocessors [7, 8], making the power consumption of caches and main memory an important issue.

Although memory hierarchies have been a hot topic for many years, little attention has been given to power consumption in caches. Traditionally, the great variety of caches has only been compared to each other with respect to hit rates (i.e. speed), ignoring power consumption. In contrast, recent papers which consider the *power consumption* of caches tend to ignore hit rates. This is unfortunate, because it is undesirable to have energy-efficient caches which are also very slow. We also believe that hit rates play a key role in truly evaluating the energy-efficiency of a cache, because low hit rates also lead to more frequent main memory accesses which consume considerably more power than cache accesses [6].

In this paper, we compare the energy consumption of various sizes and styles of caches. We consider not only the energy consumed by a cache hit, but also the energy consumed on cache misses, generating a model that allows comparison of the power consumed by the entire memory hierarchy. We also consider the relative speeds of these caches in order to shed more light on the trade-off between power and speed in caches.

2 MODELING POWER CONSUMPTION

Much work has been done in recent years to model the energy consumed by various microprocessor components [5, 6, 8]. This is an especially difficult problem, because it is difficult to obtain experimental data for many modern-day CPUs. And even when it is possible, it is often undesirable. Because many fabrication and implementation details greatly affect the power consumption for a particular component, working exclusively with one architecture makes improvements in the general case near impossible.

Caches are no exception. Although some work has been focused on particular caches [4, 9], other research efforts have focused on modeling the power consumption of caches in general. One such model is proposed by Su and Despain in [8]. Because of the generality and intuitiveness of this model, we decided that it would meet our needs as a basis for modeling memory access power consumption.

To generalize the model in order to include main memory, we take into account the amount of power consumed on a cache miss. This will be the amount of power consumed in the decode path of the cache in addition to the amount of power consumed by an access to main memory. Because there seems to be a great deal of disagreement concerning the ratio of power consumed by the cache to the power consumed by main memory, we generate all of our energy values over a range of reasonable ratios. By providing data sets which consider many different ratios, we will be aiding future designers who will probably see many different ratios as on-chip fabrication technologies continue to differ from off-chip technologies.

After accounting for main memory accesses, our function, which models energy consumption for memory hierarchies, is as follows. The equations and explanations for Energy_{dec}, Energy_{cell}, and Energy_{io} are borrowed directly from [8]. We include an explanation here only as an aid to the reader.

$$
\begin{aligned}
\text{Energy} &= \text{Energy}_{\text{dec}} + \\
&\quad (\text{hit\_rate}) \cdot (\text{Energy}_{\text{cell}} + \text{Energy}_{\text{io}}) + \\
&\quad (1 - \text{hit\_rate}) \cdot (\text{Energy}_{\text{main\_mem}} \cdot \text{CacheMemRatio}) \\
\text{Energy}_{\text{dec}} &= \alpha \cdot Addr\_bus\_bs \\
\text{Energy}_{\text{cell}} &= \beta \cdot Word\_line\_sz \cdot Bit\_line\_sz \cdot Bit\_line\_sb \\
\text{Energy}_{\text{io}} &= \gamma \cdot (Addr\_pad\_bs + Data\_pad\_bs)
\end{aligned}
$$

hit_rate	hit rate of the cache.
Energy$_{\text{main\_mem}}$	base energy value consumed by a main memory.
CacheMemRatio	ratio of power consumption of a main memory access to an on-chip cache access.
$Addr\_bus\_bs$	Number of bits switched on address buses per instruction
$Word\_line\_sz$	Number of memory cells in a word line
Bit_line_sz	Number of memory cells in a bit line
Bit_line_sb	Number of switching bit lines per instruction
$Addr\_pad\_bs$	Number of bit switches on address pads per instruction
$Data\_pad\_bs$	Number of bit switches on data pads per instruction
$\alpha.\beta, \gamma$	Constants depending on VLSI implementation (0.001, 2, 20 are used for $0.8\,\mu$m CMOS technology)

To measure cache speeds, we use the following simple equation, which is based on a 20:1 ratio between main memory speeds and cache speeds.

$$
\text{Speed} = \text{hit\_rate} + (1 - \text{hit\_rate}) \cdot 20
$$

3 DATA COLLECTION

We have observed the energy consumption of a variety of popular cache designs over a range of sizes (1K, 4K, 8K, 16K, 32K, 64K). Specifically, we consider varying ranges of associativity (direct mapped, 2-way and 4-way) and block size (1,2,4,8 word blocks) for each cache size. We limit our focus to write-back, split caches for this particular investigation, but our data set could easily be extended to include other paradigms.

We generated hit rates for these caches by simulating the SPECint92 benchmarks **gcc**, **spice**, **tex** using the software cache simulation tool, **dinero** [3]. Each memory trace contains over a million entries. We then averaged the results. These can be found in the tables in [2].

Based on the findings presented in current research papers [1], we have chosen values for the cache-to-main memory power consumption ratio which reflect a one to two order of magnitude difference.

With these input parameters and a spreadsheet, we were able to create a table of data points. The data points in this table were then used in the 3D plotting tool **Surfer** to create graphs (these can be found in [2]) that allow us to visually analyze the results of our survey. After analyzing the results, we were then inspired to graph a few hand-picked caches with the best energy consumption behavior against there respective speeds. This allows us to understand the trade-off between optimizing memory hierarchies for energy consumption and for speed.

4 ANALYSIS OF RESULTS

The results from our analysis shed some light on the issue of power consumption in memory hierarchies. It seems that 8K and 16K caches with large blocks and 2-way set-associativity perform best with respect to all tested power consumption ratios. In contrast, 64K caches perform extremely badly in all situations with all different cache styles. 32K caches also perform quite badly in all situations also. 1K caches perform well as long as main memory accesses consume only about one order of magnitude more energy than cache accesses, but perform extremely poorly as the ratio between the cache and the main memory grows to two orders of magnitude.

A full graphical summary of our results can be found in [2]. Speed comparisons can also be found in [2].

4.1 DATA CACHES

Our analysis reveals that when considering both power consumption and speed, the optimal first-level data cache is an 8K 2-way set-associative cache with a block size of 8 words. However, 16K 4-way caches with a block size of 8 are not much less energy efficient, but are somewhat faster. Because of this, we consider both 8K and 16K caches in the following graphs.
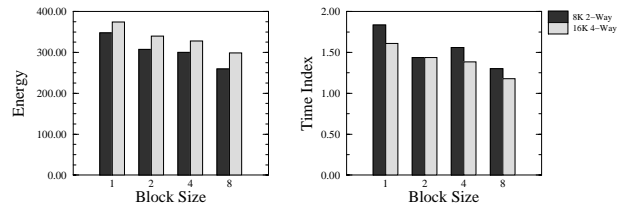

Figure 1: Data cache energy and speed versus block size.

It appears that data caches consume consistently less power as block size increases (see figure 1[1]). We find that an 8K 2-way set-associative cache with a block size of 8 consumes 1%–20% less power and is 33% faster than the same cache with a 4 word block. This trend extends to block sizes of 2 and 1 words as well. Apparently, although a larger block size leads to an increase in power for every miss and every hit, the great improvement in hit rate outweighs this penalty.
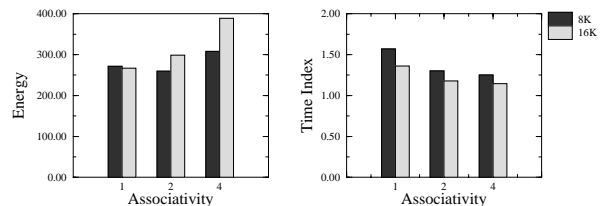

Figure 2: Data cache energy and speed versus associativity.

A surprising result of our analysis is that 2-way set-associativity appears to be optimal for many cache sizes(see

---

[1]for all bar charts, lower time indices and lower energy values indicate faster and more energy efficient caches respectively

figure 2). Although 2-way caches perform slightly more slowly than 4-way caches (about 4% slower for an 8K cache with block size of 8), the power savings easily compensate for the loss (13%–33% better depending on the cache-to-main memory power consumption ratio). In contrast, 2-way caches perform better than direct mapped caches for high cache-to-main memory ratios, while performing somewhat worse for low ratios. However, the difference in speed is significant–the 8K 2-way cache with block size of 8 words performs about 21% faster than the direct mapped cache. For the 16K cache, the direct mapped cache performs more energy-efficiently, but is much slower than both the 2-way and 4-way caches.
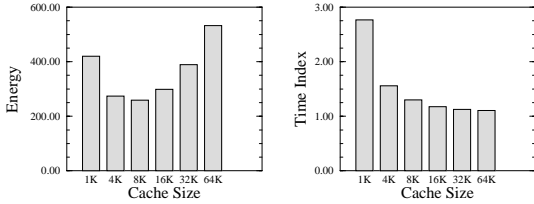
Figure 3: Data cache energy and speed versus cache size.

Finally, we note that for data caches, 8K seems to be the optimal size based on power and speed statistics (see figure 3). Although 8K caches are somewhat slower than 16K caches (about 10% for 2-way caches with block size 8 words), they are significantly more energy efficient (up to 33% less energy is consumed). For low cache-to-main memory power consumption ratios, it may be desirable to use the larger cache, but for high ratios, the 8K cache is clearly the best choice. Caches larger than 16K and smaller than 8K tend to perform very inefficiently, although very small caches can perform well when the power cost for accessing main memory is small.
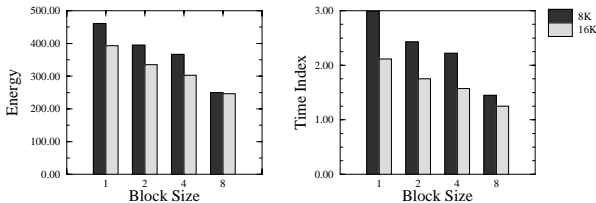
4.2 INSTRUCTION CACHES

Figure 4: Instruction cache energy and speed versus block size.

For instruction caches, the analysis is not as clear, but it seems that a 16K direct mapped cache with block size 8 performs optimally. Again, larger block sizes are indisputedly better (see figure 4), but the tradeoff between speed and power with respect to cache size becomes more pronounced (see figure 5). For example, 8K 4-way caches with block size 8 perform 14%–37% more efficiently than 16K caches, but the 16K cache performs 15% faster. The difference becomes less pronounced for lower set-associativities. The 16K direct

mapped cache with block size 8 words actually performs consistently more efficiently as well as significantly faster than the 8K version.
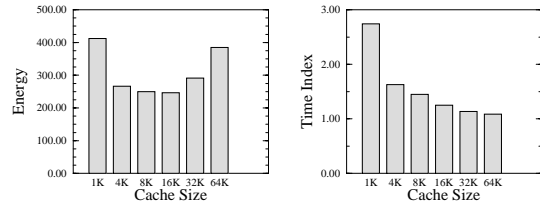

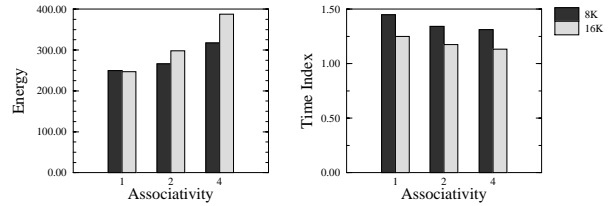Figure 5: Instruction cache energy and speed versus cache size.

Figure 6: Instruction cache energy and speed versus associativity.

Finally, we note that lower set-associativity seems better for instruction caches (see figure 6). Although the 16K direct mapped cache with block size 8 performs approximately 10% slower than the 4-way version, it also performs between 44% and 87% more efficiently. If the compromise in speed for using a direct mapped cache seems too severe, the 2-way may be a good compromise, only performing 4% slower but still achieving a 26%–37% savings in power.

5 CONCLUSION

In this paper, we have compared the energy consumption of various sizes and styles of caches. We have considered not only the energy consumed by a cache hit, but also include the energy consumed on cache misses, generating a model that allows comparison of the power consumed by the entire memory hierarchy. The result of this investigation is that 8K 2-way data caches with large block sizes tend to perform with the best energy efficiency while maintaining access times comparable to larger, much less energy-efficient caches. For instruction caches, 16K direct mapped caches with large block sizes appear to perform the best. As a point of comparison with modern computer architectures, we note that the Power PC uses 16K 4-way data and instruction caches with 8 word block sizes, while the Pentium Pro uses 8K 4-way data and instruction caches with 8 word block sizes. Our analysis concludes that both of these schemes are excellent with respect to power, but optimally, a combination of the schemes should be considered while also considering lower set-associativity.

We also discovered that the ratio of power consumed by main memory accesses to that of cache accesses has a significantly different affect on different caches, making it a

crucial consideration when choosing the optimal cache to fit in a memory hierarchy. Specifically, for large ratios(two orders of magnitude), we found that highly set-associative caches outperformed direct mapped caches both in energy consumption and speed, while direct-mapped caches were more energy efficient for lower ratios. This work should help to guide two groups in the computer engineering community. It should help to focus the attention of researchers who are working on improving cache energy usage, while also providing a guideline for system designers who wish to optimize for energy consumption.

6   FUTURE WORK

The power analysis presented in this paper could be extended to consider additional levels of caching, including a second-level unified cache. It would also be helpful to compare the values predicted by this analysis to those found in real systems to test the accuracy of the model used here. Finally, this model could be used as a component in a complete power estimation system such as that developed in [5].

ACKNOWLEDGEMENT

REFERENCES

[1] Meng-Fan Chang, Mary Jane Irwin, and Robert Michael Owens. Power-area trade-offs in memory arrays with dual word lines. Submitted to IEEE Symposium on Low Power Electronics 1997.

[2] Patrick Hicks, Matthew Walnock, and Robert Michael Owens. Analysis of power consumption in memory hierarchies. Technical Report CSE-97-003, Pennsylvania State University Department of Computer Science and Engineering, June 1997.

[3] Mark Hill. Dinero III cache simulator. online document available via http://www.cs.wisc.edu/~markhill, 1989.

[4] Uming Ko, Poras T. Balsara, and Ashwini K. Nanda. Energy optimization of multi-level processor cache architectures. In *IEEE Symp on Low Power Electronics*, pages 45–49, 1995.

[5] Huzefa Mehta. *System Level Power Analysis*. PhD thesis, Penn State University, 1996.

[6] T. Sato, Y. Ootaguro, M. Nagamatsu, and H. Tago. Evaluation of architecture-level power estimation for CMOS RISC processors. In *IEEE Symp on Low Power Electronics*, pages 44–45, 1995.

[7] Y. Shimazaki et al. An automatic-power-save cache memory for low-power RISC processors. In *IEEE Symposium on Low Power Electronics*, pages 58–59, 1995.

[8] C. Su and A. Despain. Cache design trade-offs for power and performance optimization: A case study. In *IEEE Symposium on Low Power Electronics*, pages 63–68, 1995.

[9] Nestoras Tzartzanis and William C. Athas. Energy recovery for the design of high speed, low- power static RAMs. In *IEEE Symposium on Low Power Electronics*, pages 55–60, 1996.