Leakage Power Optimization Techniques for Ultra Deep Sub-Micron Multi-Level Caches

Nam Sung Kim, David Blaauw, Trevor Mudge
Advanced Computer Architecture Lab
Department of Electrical Eng. and Computer Sci., University of Michigan, Ann Arbor
{kimns, blaauw, tnm}@eecs.umich.edu

Abstract

On-chip L1 and L2 caches represent a sizeable fraction of the total power consumption of microprocessors. In deep sub-micron technology, the subthreshold leakage power is becoming the dominant fraction of the total power consumption of those caches. In this paper, we present optimization techniques to reduce the leakage power of on-chip caches assuming that there are multiple threshold voltages, VTH's, available. First, we show a cache leakage optimization technique that examines the trade-off between access time and leakage power by assigning distinct VTH's to each of the four main cache components — address bus drivers, data bus drivers, decoders, and SRAM cell arrays with sense-amps. Second, we show optimization techniques to reduce the leakage power of L1 and L2 on-chip caches without affecting the average memory access time. The key results are: 1) 2 VTH's are enough to minimize leakage in a single cache; 2) if L1 size is fixed, increasing the L2 size can result in much lower leakage without reducing average memory access time; 3) if L2 size is fixed, reducing L1 size can result in lower leakage without loss of the average memory access time; and 4) smaller L1 and larger L2 caches than are typical in today's processors result in significant leakage and dynamic power reduction without affecting the average memory access time.

1. Introduction

As semiconductor process technology moves below 0.1μm, subthreshold leakage power is becoming a dominant fraction of total power. A potentially important source of this power loss is on-chip caches, because larger and larger on-chip caches are being integrated on the chip. For example, Intel's Madison processor has 1MB and 6MB on-chip L2 and L3 caches respectively [1].

To alleviate this problem, transistors in caches could be designed for low leakage, for example, by assigning them a high threshold voltage, VTH, or by controlling the VTH with adaptive body biasing or, if a better balance of speed and power is required, by employing dual VTH [2-8]. Traditionally, only two VTH's have been available in high performance process technologies, allowing cache designers limited flexibility to suppress leakage current. To further improve the leakage, several dynamic circuit and microarchitectural techniques [9-12] have therefore been proposed targeted at leakage power reduction of L1 caches. However, due to the increasing importance of subthreshold leakage current, the number of available VTH's in future process technologies will increase. Next generation 65nm processes are expected to support 3 VTH's and future processes are likely to provide designers with even more VTH choices. This increase provides new flexibility for leakage power reduction methods, allowing new trade-offs between the VTH of different parts of a cache and between different levels in the cache hierarchy. The availability of additional VTH's suggests a new examination of the trade-off between cache size and VTH to reduce subthreshold leakage power loss.

In this paper, we investigate combinations of circuit and microarchitectural techniques to minimize leakage and dynamic energy in microprocessor memory hierarchies under access time constraints. We present systematic approaches to VTH assignment and memory hierarchy configuration to minimize leakage and dynamic energy consumption. Our study is limited to hierarchies consisting of L1, L2 caches, and main memory. However, our approach is readily extended to systems with more cache levels.

ICCAD'03, November 11-13, 2003, San Jose, California, USA.

First, we examine the optimization of leakage power of individual on-chip cache memories that can be achieved if more than one VTH is used to optimize leakage power dissipation. We show how many independent VTH's are needed for effective leakage power reduction and how much VTH can be increased effective without sacrificing the access time of caches. Second, we show that cache miss characteristics of L1 and L2 caches under SPEC2000 workloads allow us to reduce leakage as well as total dynamic energy dissipation while maintaining the same overall average memory access time in the microprocessor memory system.

The next section of this paper explains the circuit and microarchitectural simulation methodology used in this research. Section 3 and Section 4 present our proposed leakage power optimization techniques for individual and multi-level cache systems. Section 5 discusses future direction of this research and adds some concluding remarks.

2. Methodology

2.1 Circuit simulation

To examine trade-offs between leakage power dissipation and access time of a microprocessor memory system, we need SRAM access time and leakage power models. Rather than starting from the scratch, we could have built on a widely used SRAM cache memory model called "CACTI" [13]. This model estimates access time, dynamic energy dissipation, and area of caches for the given configuration parameters such as total size, line size, associativity and number of ports. However, it is based on 0.8μm CMOS technology and applies linear scaling to obtain the figures for smaller technologies. Also, it does not support access time, and leakage power models for multiple VTH's. To address these short-comings, we designed SRAM's with 70nm technology [14] and used HSPCE simulations to derive our leakage and access time models.

Caches were designed with sizes ranging from 16KB to 1024KB. Bit-line and word-lines were segmented to improve access time, and sub-banks were employed to reduce dynamic power dissipation [15] (see Table 1 for the cache sub-bank configuration). The caches were broken into four components for the purposes of assigning distinct VTH's: address bus drivers, data bus drivers, decoders, and 6T-SRAM cell arrays with sense-amps. We employed an "H-tree" topology for the address and data bus routing and inserted repeaters on each branch of the buses to optimize the

TABLE 1. SRAM organizations for each cache size, showing sub-bank organization (sbank).

cache size	# of sbanks	sbank size	sbank organization	
			bit-lines	word-lines
16KB	4	4KB	256	128
32KB	8			
64KB	4	16KB	512	256
128KB	8			
256KB	4	64KB	1024	512
512KB	8			
1024KB	16			

access time of the caches. We assume that the circuits are designed with the 70nm technology in anticipation of the next generation of process technology. The circuit styles and the "W/L" ratios of transistors for the circuits are based on the CACTI model. We also include the interconnect capacitance and resistance for the long wires such as bit-lines, word-lines, address, and data bus wires based on the values predicted in [16].

HSPICE simulations were run to obtain access times, dynamic, and leakage power dissipations for various cache sizes and for various VTH's for their four components. We considered VTH's between 0.2 and 0.5V in steps of 0.05V at 1V supply voltage. In addition, we measured the delay time, dynamic power, and leakage power dissipation of each memory component separately. Figure 1 and Figure 2 show VTH vs. leakage power and delay time of the 7x128, 8x256, and 9x512 row decoders that we designed. The HSPICE simulation results shown in Figure 1 agree with the exponential decay in leakage power with VTH that is characteristic of CMOS circuits:

$$P_{leakage} \propto e^{-V_{TH}} \quad (1)$$

The CMOS circuit delay of ultra deep sub-micron short-channel transistors are:

$$T_d = \frac{k \cdot L \cdot V_{DD}}{(V_{DD} - V_{TH})^{\alpha}} \quad (2)$$

where k is a constant and α is about 1.3 depending on the technology [17]. Figure 2 shows the HSPICE measurement results for the circuit delay of the decoders, which agrees with the Equation (2). However, the circuit delay or access time also fits well to an exponential growth function with a very small exponent over our range of interest. It was convenient in some of our optimizations to model delay this way.

To obtain approximated analytic equations for the leakage power and access time as a functions of VTH and cache size, we applied exponential decay and growth curve fitting techniques after measuring leakage power and access time for each VTH point and each cache size. Assuming that we can apply four distinct VTH's, the analytic approximated equations for leakage power, *LP*, and access time, *AT* are:

$$LP(V_{TH1}, \ldots, V_{TH4}) = A_0 + A_1 e^{-V_{TH1}/a_1} + \ldots + A_4 e^{-V_{TH4}/a_4} \quad (3)$$

$$AT(V_{TH1}, \ldots, V_{TH4}) = B_0 + B_1 e^{V_{TH1}/b_1} + \ldots + B_4 e^{-V_{TH4}/b_4} \quad (4)$$

where VTH1, VTH2, VTH3 and VTH4 represent the VTH's for address bus drivers, data bus drivers, decoders and 6T-SRAM cell arrays, respectively. Each exponential term evaluates the leakage power dissipation of one of the four components. Each coefficient (e.g., A1 and a1) in the equation is extracted using the Origin 6.1 curve fitting software based on HSPICE simulation measurement results. We also define baseline caches as those with all low-VTH's (0.2V). According to the HSPICE measurements, the access time and the leakage power trends of the designed baseline caches agree with those of earlier studies.

2.2 Microarchitectural simulation

We use the SimpleScalar cycle simulator [18] to obtain L1 and L2 cache miss rates, which were used to estimate the *average memory access time* (AMAT) [19] for a two-level cache memory hierarchy system. The SPEC2000 benchmark suite was used and compiled with GCC 2.6.3 using O4 level optimizations. It was also statically linked with library code. To get reliable L2 cache miss rates, we completed the execution for each benchmark application and each given input. The reason we completed the executions is that L2 cache accesses are far less frequent than L1 cache accesses and an insufficient number of L2 accesses may result in unrepresentatively higher L2 cache miss rates.

Table 2 shows the L1 and L2 cache miss rates for 16KB, 32KB, and 64KB L1 caches respectively. We assume that we have two L1 caches, one each for instruction and data, but a unified L2 cache for each configuration. The L1 instruction caches are direct mapped, and the L1 data caches are 4-way set associative. Also, the L2 caches are 8-way set associative. Each L1 cache miss rate is obtained by the sum of the number of total instruction and data cache misses divided by the sum of total instruction and data cache accesses. A 16KB L1 means instruction and data caches are each 16KB in size.

3. A single cache leakage optimization

Table 3 shows the dynamic energy per access, and the dynamic and leakage power dissipation of the baseline caches used in this research. First, the dynamic energy dissipation is measured for an access using HSPICE, then we divide it by the access time of the cache to estimate the average dynamic power dissipation during the access. If caches are designed with the same size sub-bank, there is not much difference in the average dynamic energy dissipation, because we assume that only one sub-bank is accessed during the cache access and the sub-bank size is the same regardless of the cache size in a certain range (e.g, 256~1024KB). The only difference in energy dissipation of the different caches with the same sub-bank size is caused by the energy dissipation of address and data bus drivers. However, the access time increases as the cache size grows due to the propagation delay caused by longer address

FIGURE 1. VTH vs. leakage power of 7x128, 8x256, and 9x512 row decoder logic.

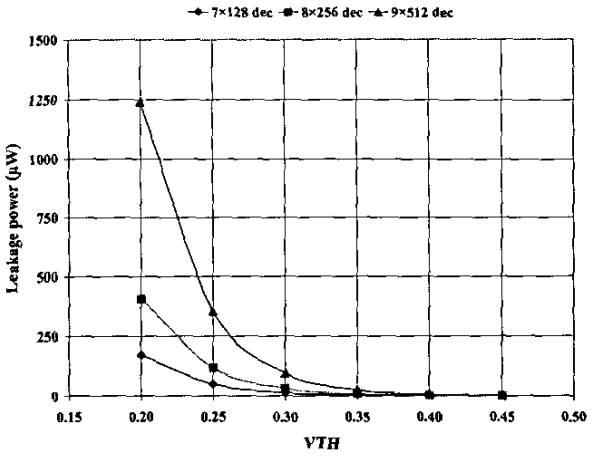


FIGURE 2. VTH vs. delay of 7x128, 8x256, and 9x512 row decoder logic.

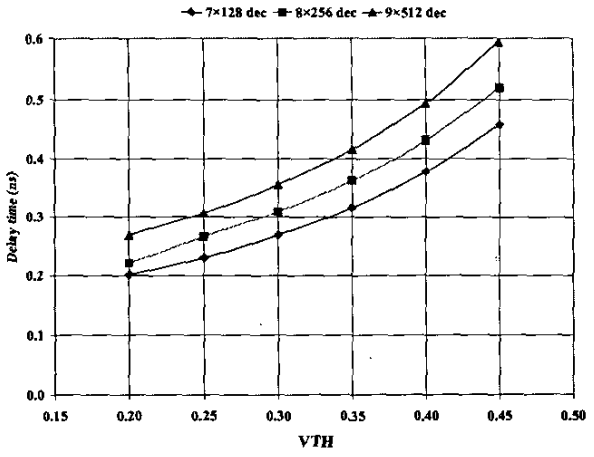


TABLE 3. **Dynamic and leakage power dissipation of baseline caches (VTH = 0.2V).**

cache size	dyn. energy (pJ)	dyn. power (mW)	leak. power (mW)	leakage fraction
16KB	10.9	16.4	10.7	0.396
32KB	14.5	18.3	22.1	0.547
64KB	30.4	30.6	41.9	0.579
128KB	51.3	43.6	85.0	0.792
256KB	139.6	90.3	165.8	0.671
512KB	159.9	78.6	332.7	0.664
1024KB	201.3	75.2	666.7	0.871

and data interconnect wires. This results in smaller average dynamic power dissipation in some cache sizes (compare the dynamic power dissipation of 512KB and 1024KB cache with that of 256KB cache in Table 3).

The experimental results also illustrate the leakage power problem in large caches with all low-VTH's (0.2V). In 1024KB caches, the percentage of the leakage can be as much as 87%. Furthermore, leakage power is dissipated all the time while the dynamic power is consumed only when the cache is accessed. For the L2 caches, they are only accessed when a L1 cache miss occurs. Therefore, the actual percentage of leakage power averaged over the long run is much higher than the numbers appeared in the Table 3.

3.1 Leakage power optimization with multiple VTH assignments

Assuming that we can assign different VTH's to each component of the cache, it is important to determine how many VTH's are cost-effective because an extra mask and process step are needed for each additional VTH.

To find the minimum leakage power of caches using four different VTH's under a specified target access time constraint, we formulate the problem as follows:

$$min\left\{LP = A_0 + A_1 e^{-V_{TH1}/a_1} + \ldots + A_4 e^{-V_{TH4}/a_4}\right\} \quad (5)$$

$$constraint\ AT = B_0 + B_1 e^{V_{TH1}/b_1} + \ldots + B_2 e^{V_{TH4}/b_4} \quad (6)$$

$$0.2 \le V_{TH1}, V_{TH2}, V_{TH3}, V_{TH4} \le 0.5 \quad (7)$$

where VTH1, VTH2, VTH3 and VTH4 represent the VTH's for address bus drivers, data bus drivers, decoders and 6T-SRAM arrays.

There exits numerous combinations of VTH1, VTH2, VTH3 and VTH4 satisfying a specific target access time. Among these combinations, we find a quadruple of VTH1, VTH2, VTH3 and VTH4 producing minimum leakage power using a numerical optimization method (e.g., Matlab's *fmincon* function) that satisfies a specified access time error range within 5%. We can repeat this with modified objective and constraint functions to find an optimal VTH combination for the cache memories that have only 2 or 3 VTH's.

To examine the dependence of the optimization results on access time, we sweep the target access time from the fastest possible (all low-VTH's) to the slowest possible (all high VTH's). The followings are the VTH assignment schemes we examined in this study:

- *Scheme I*: assigning a high-VTH to all the cache circuit components including address bus drivers, data bus drivers, decoders and 6T-SRAM cell arrays.
- *Scheme II*: assigning a high-VTH only to 6T-SRAM cell arrays and assigning a default- or low-VTH (0.2V) to the rest of the transistors.
- *Scheme III*: assigning a high-VTH to 6T-SRAM cell arrays and assigning another high-VTH to the peripheral circuit components of the cache (address bus drivers, data bus drivers and decoders).
- *Scheme IV*: assigning four different high VTH's to all four circuit components of the cache.

In Figure 3, we plot the normalized minimum leakage power values at different target access times (105%, 110%, 115% and so on). In the graph, the normalized delay and leakage of 100% correspond to the access time and leakage of a cache with all default-VTH's for all the four cache components. Also, the 115% access time means that it is 15% slower than the baseline cache.

Figure 4 shows the VTH trends of each cache component for the normalized delay of 32KB cache with scheme IV. According to this simulation result, the VTH of 6T-SRAM cell array starts increasing first, because it has the most significant impact on the leakage power reduction but has the least significant impact on the overall access time. In contrast, the decoder has the least significant impact on the leakage but has the most significant impact on the overall access time. The address and data bus drivers show mid-

TABLE 2. **L1 and L2 cache miss rates for 16KB - 64KB L1 caches and 128KB ~ 4096KB L2 caches. A 16KB L1 means instruction and data caches are each 16KB in size.**

L1 size	Miss rate	L2 size	Miss rate
16KB	3.3%	128KB	34.2%
		256KB	32.2%
		512KB	30.6%
		1024KB	25.5%
32KB	2.5%	256KB	40.3%
		512KB	38.2%
		1024KB	31.8%
		2048KB	19.1%
64KB	1.5%	512KB	41.6%
		1024KB	35.6%
		2048KB	21.8%
		4096KB	16.3%

FIGURE 3. **Normalized leakage vs. delay of 32KB caches with the scheme I, II, III, and IV.**

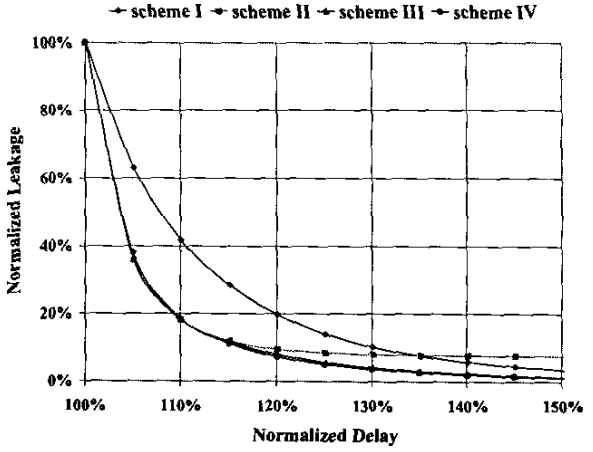
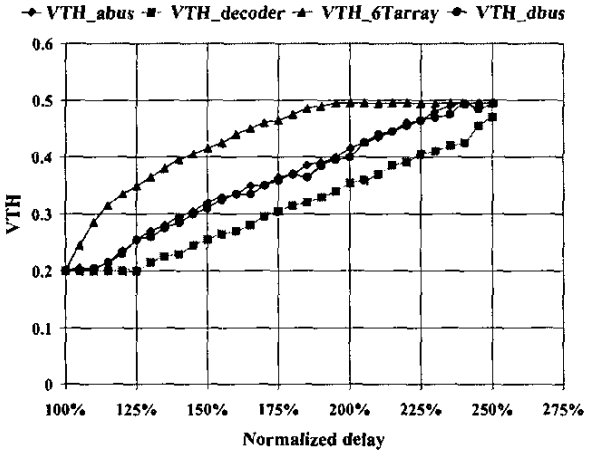


FIGURE 4. VTH trends of each cache component vs. normalized delay of 32KB caches with the scheme IV.



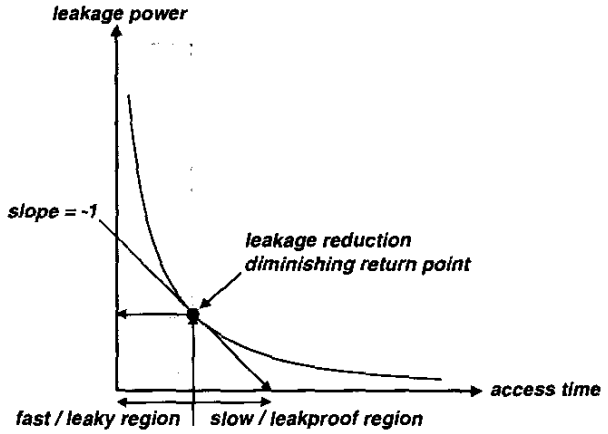
dling impact on both leakage and access time compared to the 6T-SRAM cell array and the decoder. This trend suggests that we should give the top optimization priority to the 6T-SRAM cell array to achieve the lowest leakage power of caches for a given access time constraint.

Table 4 compares the leakage power of schemes II, III, and IV for each cache size against scheme I. As expected, we can reduce more leakage power while achieving the same access time by having more VTH's to control. However, as the target access time increased to more than the 150% point of the scheme II, the caches dissipate more leakage power than those employing the scheme I caches, see both Table 4 and Figure 3. The address and data bus drivers and decoders — cache peripheral circuits consume non-negligible leakage power. Also, the leakage power by those components becomes substantial when we cut down the leakage power of 6T-SRAM array. Furthermore, the slowest delay point of the scheme II ends around 150% in small caches. This means that the peripheral circuits also play important roles in both leakage power and access time of caches. In other words, increasing the VTH of the 6T-SRAM cell array alone gives us diminishing return at some point without reducing the leakage power further. This is why scheme I caches give even better results than scheme II caches with increase of VTH. Other noticeable results are that there is a negligible difference between schemes III and IV in terms of leakage power reduction, which implies that 2 distinct VTH's or scheme III for caches are enough for the leakage reduction.

TABLE 4. Normalized leakage power of scheme II, III, and IV caches for each cache size against scheme I at the target access times (125%, 150%, and 175%).

cache size	125%			150%			175%		
	II	III	IV	II	III	IV	II	III	IV
16KB	0.50	0.41	0.40	1.64	0.40	0.38	N/A	0.42	0.40
32KB	0.61	0.41	0.37	2.23	0.37	0.32	N/A	0.35	0.34
64KB	0.48	0.46	0.45	0.94	0.47	0.44	3.07	0.49	0.46
128KB	0.46	0.42	0.39	1.23	0.41	0.37	N/A	0.41	0.38
256KB	0.57	0.57	0.57	0.79	0.56	0.54	2.21	0.60	0.62
512KB	0.47	0.46	0.44	0.81	0.44	0.43	2.69	0.46	0.48
1024KB	0.33	0.32	0.30	0.82	0.28	0.26	N/A	0.30	0.29

FIGURE 5. An optimal leakage / access time trade-off point.



3.2 Trade-off between leakage and access time

One interesting point from these experimental results is that we may not need to use very high VTH (e.g. 0.5 for 1V supply voltage) since this impedes the circuit speed unnecessarily without reducing the leakage power further. Figure 5 shows a general trend in a leakage power vs. access time graph. In the "fast / leaky" region, we can reduce the leakage power dramatically with a small increase of the access time. On the other hand, we cannot reduce the leakage power very much by increasing VTH after some points, while access time increases rapidly. Based on this observation, we can calculate a point whose tangential slope equals negative "1" in the graph shown in Figure 5, and we call this the inflexion point of the leakage power — in other words, this point can be regarded as an optimal leakage / access time trade-off point.

Table 5 shows the normalized access times vs. leakage power of caches at their inflexion point. This is also good indication for how many VTH's are good enough to control the leakage power effectively. The normalized access time and leakage power are based on the fastest access time and leakiest leakage power values when using all low-VTH's (0.2V). According to this result, we can achieve more leakage power reduction with a faster access time as we increase the number of distinct VTH's, but we quickly reach a point of diminishing returns.

TABLE 5. The inflexion points of leakage power reduction when increasing VTH's.

Cache size	Normalized access time				Normalized leakage power			
	I	II	III	IV	I	II	III	IV
16KB	1.24	1.15	1.15	1.16	0.151	0.118	0.115	0.111
32KB	1.24	1.13	1.14	1.15	0.152	0.135	0.122	0.111
64KB	1.25	1.18	1.18	1.19	0.154	0.116	0.116	0.112
128KB	1.24	1.16	1.16	1.17	0.154	0.118	0.115	0.108
256KB	1.24	1.20	1.20	1.20	0.147	0.117	0.118	0.113
512KB	1.23	1.18	1.18	1.18	0.145	0.107	0.108	0.103
1024KB	1.23	1.15	1.15	1.16	0.143	0.091	0.092	0.089

4. Two-level cache leakage optimization

In a microprocessor memory system, the average memory access time (AMAT) is a key issue when considering the overall microprocessor performance. We can estimate AMAT for a multi-level cache hierarchy as follows [19]:

$$AMAT(L1) = AT(L1) + MR(L1) \times AMAT(L2) \quad (8)$$

$$AMAT(L2) = AT(L2) + MR(L1, L2) \times AT(mainmem) \quad (9)$$

where *AT* and *MR* are access time and miss rate for the specific size of caches, respectively. In (9) we use the local miss rate *MR*(*L1*, *L2*), i.e., the miss rate of the L2 seen by the accesses that first miss the L1. This depends on the size of L1 caches, because the number of accesses of the L2 cache is equal to the number of misses of L1 caches (of course, the total misses remains the same).

In addition, to compare the dynamic energy dissipation of each memory hierarchy configuration, we define the *average memory access energy* (AMAE) similarly to the AMAT. Assuming that the L1 cache is accessed every cycle, the AMAE represents the average energy dissipation per access in the entire microprocessor memory system that includes L1, L2 and main memory. We can estimate AMAE, as follows:

$$AMAE(L1) = AE(L1) + MR(L1) \times AMAE(L2) \quad (10)$$

$$AMAE(L2) = AE(L2) + MR(L1, L2) \times AE(mainmem) \quad (11)$$

where AE is the average energy dissipation per access of memory structures.

To obtain the main memory access time and dynamic energy dissipation per access, we use a 2-channel 1066Hz 256MB RAMBUS DRAM RIMM module whose sustained transfer rate is 4.2GB/s [20]. Though the sustained transfer rate is quite high, we should also consider the RAS/CAS latency of the memory, which is about 20ns. For the energy dissipation per access, we used the number given in [21] — 3.57nJ per access.

4.1 L2 cache leakage power optimization

We will examine the leakage power optimization of L2 caches first, because their contribution to leakage power dominates due to their size. Consider a conventional cache hierarchy of 16KB and 128KB for L1 and L2 caches respectively, designed with low-VTH (0.2V) devices. If we fix the L1 and reduce the leakage of the L2 by increasing VTH the cache system becomes slower. However, we can maintain the same AMAT and reduce the leakage power of the L2 by increasing its size to reduce its miss rate. Since the main memory access penalty is quite significant, even a slight reduction of L2 cache miss rates results in a significant improvement of the AMAT. We note that although area was one of the most important design constraints in the past, this trend is changing and power is becoming an equally important constraint in many situations [22].

Figure 6 shows the leakage power vs. AMAT of L2 caches with the fixed size L1 cache of 16KB. Assuming that the AMAT of a 128KB L2 cache as a base, we compare the leakage power of other caches at the same AMAT point (see the dotted vertical line in Figure 6). As can be seen from the graphs, the AMAT can be maintained while the leakage power can be reduced by replacing a 128KB L2 with a 256KB L2 cache that is intentionally slowed down by increasing its VTH to reduce leakage. Similarly, the use of a 512KB L2 cache can further reduce leakage compared to the 256KB cache (see the dashed vertical line in Figure 6).

FIGURE 6. Leakage vs. AMAT for 128KB, 256KB, and 512KB L2 caches with a fixed 16KB L1 cache.

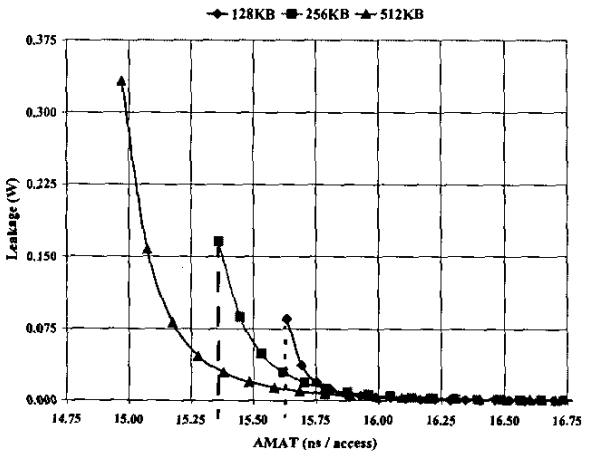


Table 6 shows the results for normalized leakage power and AMAE for each L1 cache size designed using scheme III at a fixed AMAT. To compare leakage power and AMAE, the following standard cache configurations were used: 128KB L2 with 16KB L1, 256KB L2 with 32KB L1, and 512KB L2 with 64KB L1. Table 6 gives the somewhat counter intuitive results that we can reduce both leakage power and AMAE by employing larger L2 caches while maintaining a constant AMAT.

TABLE 6. Normalized leakage power and AMAE for each L2 cache using scheme III at fixed AMAT's.

L1	L2	Normalized leakage	Normalized AMAE
16KB	128KB	1.00	1.00
	256KB	0.31	1.01
	512KB	0.15	0.99
32KB	256KB	1.00	1.00
	512KB	0.11	0.98
	1024KB	~0.00	0.89
64KB	512KB	1.00	1.00
	1024KB	0.01	0.95

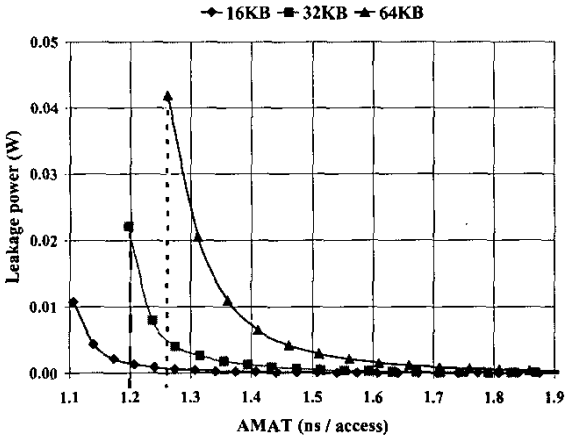
4.2 L1 cache leakage power optimization

It is hard to improve the L1 cache miss rates further because they are already quite low for 16KB, 32K, and 64KB caches when using SPEC200 benchmarks. Hence, the access time of caches is the dominant factor in determining AMAT. However, the access time of 64KB L1 cache can increase by 48% compared to a 16KB L1 cache, because access time is very sensitive to size for small caches. Essentially, cache access time increases logarithmically with size, but has a steeper slope for smaller caches than for larger caches.

This observation confirms why the AMAT of a cache hierarchy with a smaller L1 cache can be faster than one with a larger L1 caches for a certain range of cache sizes (e.g., 16KB~64KB). Figure 7 shows the leakage power vs. the AMAT of 16KB, 32KB, and 64KB L1 caches using scheme III each with a fixed L2 cache of size 512KB. Like the comparison performed in Section 4.1, the leakage power of different caches is compared at the same AMAT point. The graphs show that leakage power can be reduced by replacing a 64KB L1 cache with a 32KB L1 cache that is intentionally slowed down by increasing its VTH's to reduce the leakage power (see the dotted vertical line in Figure 7), but the resulting hierarchy has the same AMAT. Similarly, a slowed 16KB cache with increased VTH's can replace a 32KB without changing the AMAT of the L1/L2 hierarchy. The new system consumes much less leakage power (see the dashed vertical line in Figure 7).

Table 7 shows the results for normalized leakage power and AMAE for each fast but leaky L1 cache sizes using scheme III with fixed AMAT's. The comparisons were performed in the same manner as Table 6. According to the comparisons, we can reduce both leakage power and AMAE by employing smaller L1 caches. This is therefore in contrast to the case for L2 caches, where the leakage of the overall hierarchy can be reduced by increasing their size. It should be noted, these results are only valid within the specific set of sizes given in this paper. A 4KB L1 cache will have a cache miss rate that is much higher than a 16KB cache, but its access time will

FIGURE 7. **Leakage vs. AMAT for 16KB, 32KB, and 64KB L1 caches with a fixed 512KB L2 cache.**



not be sufficiently smaller to make the trade-off worthwhile. Also, the normalized AMAE is rather high because the total power fraction of L1 caches is relatively small compared to L2 caches.

5. Conclusion and future work

In this paper, we examined the leakage power and. access time trade-off trends where multiple VTH's are allowed. We used curve fitting techniques to model leakage power and access time. Our results show that 2 distinct VTH's for caches are sufficient to yield a significant reduction in leakage power. Such an arrangement can reduce the leakage power up to 91% (see scheme III in Table 5) for an 1MB SRAM cache without significantly increasing access time. We also show that smaller L1 and larger L2 caches than are typical in today's processors result in significant leakage and dynamic power reduction without affecting AMAT. Given that the processor core may need a distinct VTH, and each of the caches may need up to two VTH's (scheme III) we could require up to five distinct VTH's.

In this work, we assume that we have two-level on-chip caches. Recently, however, microprocessors with three-level caches are being deployed, and their L2 and L3 cache sizes are much larger than the caches discussed here. For future work, we will investigate leakage power optimization in a multi-level cache hierarchy that include L2 and L3 caches.

TABLE 7. **Normalized leakage power and AMAE for each L1 cache using scheme III at fixed AMAT's.**

L2	L1	Normalized leakage	Normalized AMAE
256KB	32KB	1.00	1.00
	16KB	0.06	0.95
512KB	64KB	1.00	1.00
	32KB	0.19	0.62
	16KB	0.02	0.59
1024KB	64KB	1.00	1.00
	32KB	0.12	0.64
	16KB	0.02	0.62

Acknowledgement

This work was supported by Intel Graduate Fellowship, by DARPA contract number F3361500-C-1678, and by SRC 2001-HJ-904.

Reference

[1] http://www.intel.com

[2] S. Mutoh, et al., "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *IEEE J. Solid-State Circuits*, Vol 30, pp. 847~854, Aug. 1995.

[3] T. Douseki, N. Shibata, and J. Yamada, "A 0.5-1 V MTCMOS/ SIMOX SRAM macro with multi-Vth memory cells," *IEEE Intl. SOI Conf.*, pp. 24~25, 2000.

[4] K. Nii, et al., "A low power SRAM using auto-backgate-controlled MT-CMOS," *IEEE Intl. Symp. Low Power Electronic Device*, pp. 293~298, 1998.

[5] H. Mizuno, et al., "An 18-μA standby current 1.8-V, 200-MHz microprocessor with self-substrate-biased data-retention mode," *IEEE J. Solid-State Circuits*, Vol 34, pp. 1492~1500, Nov. 1999.

[6] F. Hamzaoglu, et al., "Analysis of dual-VT SRAM cells with full-swing single-ended bit line sensing for on-chip cache," *IEEE Tran. Very Large Scale Integration (VLSI) Systems*, Vol 10, pp. 91~95, Apr. 2002.

[7] M. Powell, et al., "Gated-*VDD*: A circuit technique to reduce leakage in deep-submicron cache memories," *IEEE Intl. Symp. on Lower Power Electronics & Design*, pp. 90~95, 2000.

[8] A. Agarwal, L. Hai, K. Roy, "A single-Vt low-leakage gated-ground cache for deep submicron," *IEEE J. Solid-State Circuits*, Vol 38, pp. 319~328, Feb. 2003.

[9] N. S. Kim, et al., "Drowsy instruction caches," *IEEE Intl. Symp. on Microarchitecture*, pp. 219~230, 2002.

[10] S. Yang, et al., "An integrated circuit/architecture approach to reducing leakage in deep-submicron high-performance I-caches," *IEEE Intl. Symp. High-Performance Computer Architecture*, 2001, pp. 147~157.

[11] S. Kaxiras, et al., "Cache decay: exploiting generational behavior to reduce cache leakage power," *IEEE Intl. Symp. on Computer Architecture*, pp. 240~251, 2001.

[12] H. Zhou, et al., "Adaptive mode-control: a static-power-efficient cache design", *IEEE Parallel Architecture and Compilation Techniques*, pp. 61~70, 2001.

[13] P. Shivakumar, et al., "An integrated cache timing, power, and area model," *WRL Research Report*, Feb. 2002.

[14] *http://www-device.eecs.berkeley.edu*

[15] K. Ghose, and M. Kamble, "Reducing power in superscalar processor caches using subbanking, multiple line buffers and bit-line segmentation", *IEEE Intl. Symp. on Low Power Electronic & Design*, pp. 70 -75, 1999.

[16] R. Ho, et al., "The future of wires", *Proceedings of the IEEE*, Vol. 89, pp. 490 -504, Apr. 2001.

[17] T. Kuroda et al., "A 0.9-V, 150-MHz, 10-mW, 4 mm2, 2-D discrete cosine transform core processor with variable threshold-voltage scheme," *IEEE J. Solid-State Circuits*, Vol.31, pp.1770-1779, Nov. 1996.

[18] D. Burger et al., "The SimpleScalar toolset version 2.0", *Tech. Rept. TR-97-1342*, Univ. of Wisconsin-Madison, Jun. 1997.

[19] J. Hennessy et al., "Computer architecture - A quantitative approach," 3rd ed., Morgan Kaufmann, pp. 406~408, 2003.

[20] Rambus Inc., "800/1066MHz RDRAM advanced information," *http://www.rambus.com*, Ver. 0.6, 2002.

[21] V. Delaluz, et al., "Compiler-directed array interleaving for reducing energy in multi-bank memories," *IEEE Asia South Pacific Design Automation Conf.*, pp. 288~293, 2002.

[22] T. Mudge, "Power: A first class design constraint," *IEEE Computers*, Vol. 34, pp. 52~57, Apr. 2001.