Quantitative Analysis and Optimization Techniques for On-Chip Cache Leakage Power

Nam Sung Kim, David Blaauw, and Trevor Mudge, *Fellow, IEEE*

***Abstract*—On-chip L1 and L2 caches represent a sizeable fraction of the total power consumption of microprocessors. In nanometer-scale technology, the subthreshold leakage power is becoming one of the dominant total power consumption components of those caches. In this study, we present optimization techniques to reduce the subthreshold leakage power of on-chip caches assuming that there are multiple threshold voltages, V_T's, available. First, we show a cache leakage optimization technique that examines the tradeoff between access time and subthreshold leakage power by assigning distinct V_T's to each of the four main cache components—address bus drivers, data bus drivers, decoders, and static random access memory (SRAM) cell arrays with sense amplifiers. 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) two additional high V_T's are enough to minimize leakage in a single cache—3 V_T's if we include a nominal low V_T for microprocessor core logic; 2) if L1 size is fixed, increasing L2 size can result in much lower leakage without reducing average memory access time; 3) if L2 size is fixed, reducing L1 size may result in lower leakage without loss of the average memory access time for the SPEC2K benchmarks; 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.**

***Index Terms*—Microprocessor memory hierarchy, multiple threshold voltage, on-chip caches, SRAM, subthreshold leakage power.**

I. Introduction

UNTIL VERY recently, only dynamic power has been a significant source of power consumption, and Moore's law has helped to control it. Shrinking processor technology below 100 nm has allowed, and actually required, reducing the supply voltage to reduce dynamic power consumption. However, smaller geometries with a low-threshold voltage exacerbate leakage, so static power is beginning to dominate the power consumption equation [1]. For example, a 90-nm Pentium 4 consumes 110 W, and roughly 40% of the total power dissipation is consumed by leakage power [2]. The excessive heat dissipation by the leakage power in the high-end 90-nm Pentium 4 processor forced Intel Corporation to adopt more expensive power delivery, cooling, and packaging systems.

Manuscript received November 10, 2003; revised July 26, 2004. This work was supported in part by the Intel Corporation, the National Science Foundation, and ARM plc, Cambridge, U.K.

N. S. Kim is with the Circuit Research Laboratory, Intel Corporation, Hillsboro, OR 97124 USA (e-mail: nam.sung.kim@intel.com).

D. Blaauw and T. Mudge are with the Advanced Computer Architecture Laboratory, University of Michigan, Ann Arbor, MI 48109 USA.

Digital Object Identifier 10.1109/TVLSI.2005.859476

A potentially important source of this power dissipation is on-chip caches, because larger on-chip caches are being integrated onto the chip. For example, an Intel processor for server applications has 1 and 6 MB on-chip L2 and L3 caches, respectively[1]; subthreshold leakage power is dissipated by all of the subbanks even if they are not accessed, while dynamic power is dissipated when a cache subbank is accessed. To alleviate this problem, transistors in caches could be designed for low subthreshold leakage, for example, by assigning them a higher threshold voltage V_T or by controlling the V_T with adaptive body biasing or, if a better balance of speed and power is required, by employing dual V_T [3]–[7]. Traditionally, at most two V_T's—one low and one high V_T—have been available in high-performance process technologies, allowing cache designers only limited flexibility for suppressing subthreshold leakage current. To further improve the subthreshold leakage, several circuit and microarchitectural techniques [8]–[13] have therefore been proposed targeted at the subthreshold leakage power reduction of L1 caches.

One consequence of the increasing importance of subthreshold leakage current is that, the number of available V_T's in future process technologies will increase. Next-generation 65-nm processes are expected to support three V_T's (one low and two high V_T's) and future processes are likely to provide designers with even more V_T choices. This increase provides new flexibility for subthreshold leakage power reduction methods, allowing new tradeoffs between the V_T of different parts of a cache and between different levels in the cache hierarchy. The availability of additional V_T's suggests a new examination of the tradeoff between cache size and V_T to reduce power loss from subthreshold leakage current.

In this study, we present systematic techniques for assigning multiple V_T's to memory hierarchies to minimize power dissipation, in particular subthreshold leakage [14]. Based on our techniques, we provide a detailed *quantitative* tradeoff analysis between access time and subthreshold leakage power of on-chip caches as a function of the number and the strength of V_T. Although the *qualitative* trends of subthreshold leakage power versus access time tradeoff are well known, this paper provides a detailed *quantitative* analysis to determine the optimal number of V_T's for given design constraints and to justify the cost of extra V_T's. First, we examine optimal leakage power dissipation for various access times in on-chip SRAM caches, when more than one high V_T is available. Then, we show how many high V_T's are needed, in addition to a nominal V_T required for the processor's general logic circuits and how much V_T should be increased for effective leakage power reduction for

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

TABLE I
CACHE ORGANIZATIONS FOR EACH CACHE SIZE

cache size (KB)	# of sub-banks	sub-bank size (KB)	sub-bank organization	
			bit-lines	word-lines
16	4	4	256	128
32	8			
64	4	16	512	256
128	8			
256	4	64	1024	512
512	8			
1024	16			

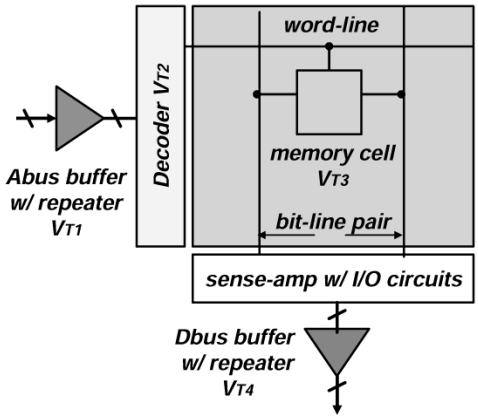

Fig. 1. Cache subbank organization.

various cache access time points. Second, we present how cache leakage power can be reduced while maintaining the same average memory access time of a processor memory system using L1 and L2 cache access statistics for SPEC2K workloads [15].

The reminder of this study is organized as follows. Section II explains our on-chip cache subthreshold leakage power and access time-modeling methodologies. Section III presents a subthreshold leakage power optimization technique for a given access time constraint and provides a quantitative tradeoff analysis of on-chip cache subthreshold leakage power and access time. Section IV presents two-level cache leakage power optimization techniques using cache access statistics. Section V discusses future directions for this line of work and adds some concluding remarks.

II. ON-CHIP CACHE LEAKAGE POWER AND ACCESS TIME MODELS

To examine tradeoffs between subthreshold leakage power and access time of a processor cache memory system, we need circuit models to estimate the subthreshold leakage power and access time of caches. Rather than starting from scratch, we could have built on a widely used cache memory model called "CACTI" [16]. This model estimates access time, dynamic energy dissipation, and area of caches for given cache configuration parameters such as total size, line size, associativity, and number of ports. However, it is based on an outdated 0.8-μm CMOS technology and it applies linear scaling to obtain the figures for smaller process technologies. Furthermore, it does not provide access time and leakage power when multiple V_T's are available. To address these shortcomings, we designed caches with the 70-nm Berkeley predictive technology model (BPTM)[2] in anticipation of the next generation of process technology. Then, we derived our subthreshold leakage power and access time models based on the HSPICE simulations of the designed cache circuits.

The designed caches ranged from 16 to 1024 KB in size. The bitlines and wordlines were segmented to improve access time, and subbanks were employed to reduce dynamic power dissipation [17] as well; see Table I for the cache subbank organization used in this study. The caches were broken into four components for the purposes of assigning distinct V_T's: *address bus drivers, data bus drivers, decoders*, and *6T-SRAM cell arrays with sense amplifiers*. Fig. 1 illustrates the cache subbank organization used in this study.

The circuit topology and the W/L ratios of transistors in the decoder circuits are based on the CACTI model but optimized for the 70-nm technology. In addition, modern techniques for lower voltage are employed for the bitline precharge and sense-amplifier circuits. For the address and data bus interconnects, we employed an *H-tree* topology and inserted repeaters on each branch of the buses to optimize the interconnect delay of cache buses. To obtain the interconnect capacitance and resistance of long wires such as bitlines, wordlines, address, and data buses, the lengths of the interconnects are estimated using SRAM cell dimensions of 1.42 μm $\times$ 0.72 μm and the cache organizations in Table I. Then, for given interconnect length, the predictor provided in footnote 2 is used to estimate the interconnect capacitance and resistance.

HSPICE simulations were run extensively to obtain leakage power and access time (or delay) models for wide ranges of cache sizes and V_T's for their four components. We considered V_T's between 0.2 and 0.5 V in steps of 0.05 V at 1-V nominal supply voltage. We measured the leakage power and the delay of each cache component separately.

A. Leakage Power Models

Fig. 2 shows V_T versus leakage power of the 7×128, 8×256, and 9×512 row decoders that we designed. The HSPICE simulation results shown in Fig. 2 agree with the exponential decay in leakage power with a linear increase of V_T that is characteristic of general CMOS circuits

$$P_{\text{leadage}} \propto e^{-V_T}. \tag{1}$$

To obtain an approximated analytic equation for leakage power as a function of V_T, we measured the leakage power of the decoders at each discrete V_T point, and we applied an exponentially decaying curve fitting method to the measured leakage power as follows:

$$P_{\text{leadage}}(V_T) = A_0 + A_1 e^{-V_T/a_1} \tag{2}$$

where A_0, A_1, and a_1 are constants derived from using Origin 6.1, which is a scientific graphing and analysis software curve-

[2][Online]. Available: http://www-device.eecs.berkeley.edu/ptm

TABLE II
CACHE COMPONENT LEAKAGE POWER MODEL COEFFICIENTS AT 70 °C DIE TEMPERATURE AND A TYPICAL CORNER FOR EACH CACHE SIZE

		Cache Size (KB)						
		16	32	64	128	256	512	1024
Address bus	$A0(\times10^{-3})$	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	$A1(\times10^{-3})$	34.33	71.61	39.24	81.84	44.14	92.07	187.92
	$a1(\times10^{-3})$	40.86	40.90	40.86	40.90	40.86	40.90	40.91
Decoder	$A0(\times10^{-3})$	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	$A1(\times10^{-3})$	97.39	194.82	243.70	487.46	765.56	1531.12	1062.24
	$a1(\times10^{-3})$	40.41	40.41	40.00	40.00	39.70	39.70	39.70
SRAM array	$A0(\times10^{-3})$	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	$A1(\times10^{0})$	1.25	2.50	5.01	10.01	19.86	39.72	79.44
	$a1(\times10^{-3})$	41.60	41.60	41.58	41.58	41.66	41.66	41.66
Data bus	$A0(\times10^{-3})$	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	$A1(\times10^{-3})$	13.55	54.18	54.18	216.73	108.35	433.40	1083.54
	$a1(\times10^{-3})$	41.62	41.62	41.62	41.62	41.62	41.62	41.62

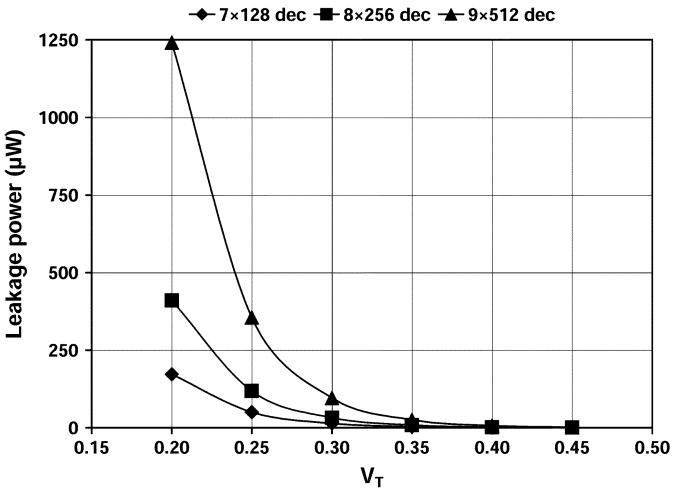

Fig. 2. Leakage power dissipation of the 7×128, 8×256, and 9×512 decoders.

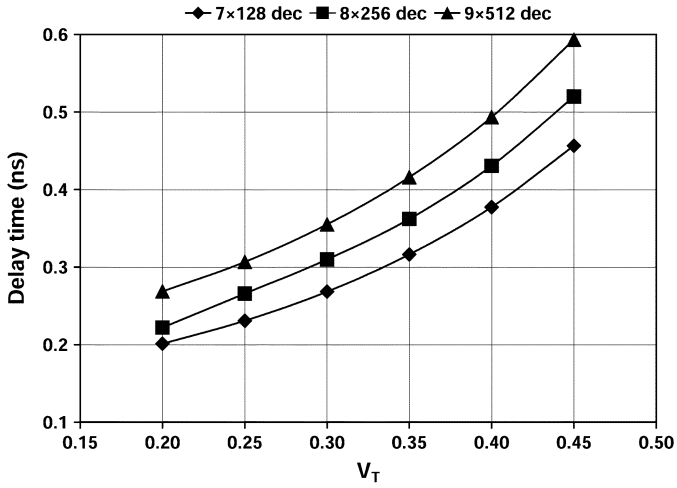

Fig. 3. Delay time of 7×128, 8×256, and 9×512 decoders.

fitting package[3]—the *R-squared error* is less than 0.001 for each fitted curves.

The rest of the cache components—address driver, data driver, and 6T SRAM cell array—show the same leakage power trend characteristics as the decoder of Fig. 2; leakage power decreases exponentially with the linear increase of V_T. Hence, an identical curve-fitting method can be applied for these components to derive leakage power models like (2). The coefficients for all of the components in (2) can be found in Table II.

Once all of the approximated analytic leakage power models for each component are derived for a cache size, the total leakage power of the cache can be approximated as the sum of the leakage power of all the components. Assuming that we apply four distinct V_T's, the analytic approximated equation for leakage power (LP) is

$$\mathrm{LP}(V_{T1}, \ldots, V_{T4}) = A_0 + A_1 e^{-V_T 1/a_1} + \cdots + A_4 e^{-V_T 4/a_4} \tag{3}$$

where V_{T1}, V_{T2}, V_{T3}, and V_{T4} represent the V_T'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.

B. Access Time Models

Fig. 3 shows V_T versus delay time of the 7×128, 8×256, and 9×512 row decoders that we designed. Basically, the

[3][Online]. Available: http://www.originlab.com

TABLE III
CACHE COMPONENT DELAY MODEL COEFFICIENTS AT 70C° DIE TEMPERATURE AND A TYPICAL CORNER FOR EACH CACHE SIZE

		Cache Size (KB)						
		16	32	64	128	256	512	1024
Address bus	B0($\times 10^{-12}$)	74.00	106.92	89.02	145.75	133.63	254.46	486.24
	B1($\times 10^{-12}$)	29.72	39.29	32.82	42.90	35.89	43.79	66.96
	b1($\times 10^{-3}$)	218.76	208.18	216.49	197.87	207.89	179.25	166.30
Decoder	B0($\times 10^{-12}$)	94.52	94.52	67.93	67.93	130.71	130.71	130.71
	B1($\times 10^{-12}$)	37.246	37.246	54.36	54.36	52.61	52.61	52.61
	b1($\times 10^{-3}$)	199.49	199.49	218.48	218.48	207.02	207.02	207.02
SRAM array	B0($\times 10^{-12}$)	60.98	60.98	121.96	121.96	243.92	243.92	243.92
	B1($\times 10^{-12}$)	103.17	103.17	206.34	206.34	412.69	412.69	412.69
	b1($\times 10^{-3}$)	279.83	279.83	279.83	279.83	279.83	279.83	279.83
Data bus	B0($\times 10^{-12}$)	26.30	58.93	41.32	96.97	85.62	203.24	433.07
	B1($\times 10^{-12}$)	10.25	20.04	13.34	24.27	16.67	26.98	51.53
	b1($\times 10^{-3}$)	221.36	201.37	215.10	188.59	199.68	168.73	160.31

CMOS circuit delay of ultra deep submicrometer short-channel transistors is

$$T_{\text{delay}} = \frac{k \cdot L \cdot V_{DD}}{(V_{DD} - V_T)^{\alpha}} \tag{4}$$

where k, L, and α[4] are constants depending on the technology and transistor sizes. The measured delay time trends in Fig. 3 agree with (4). However, the circuit delay or access time also fits very well to an exponential growth function with a very small exponent over our range of interest. It was convenient for some of our optimizations to approximate delay this way.

To obtain an approximated analytic equation for delay time as a function of V_T, we measured the delay time of the decoders at each discrete V_T point, and we fit the following exponential curve to the measured delay time:

$$V_{\text{delay}}(V_T) = B_0 + B_1 e^{-V_T/b_1} \tag{5}$$

where B_0, B_1, and b_1 are constants derived using the same technique as that used for the leakage power models.

The rest of the cache components show the same delay trend characteristics as the decoder case of Fig. 3. Hence, the same curve-fitting technique can be applied for those components to derive approximated delay time models as functions of V_T like (5). The coefficients for all the components in (5) can be found in Table III.

Once all of the approximated delay time models for each component are extracted for a specific cache size, total delay or access time of the cache can be approximated as a sum of the delay times of all the cache components. Assuming that we can apply four distinct V_T's, the analytic approximated equation for the access time (AT) is

$$\text{AT}(V_{T1}, \ldots, V_{T4}) = B_0 + B_1 e^{V_T 1/b_1} + \cdots + B_4 e^{-V_T 4/b_4} \tag{6}$$

where V_{T1}, V_{T2}, V_{T3}, and V_{T4} represent the V_T's for address bus drivers, data bus drivers, decoders, and 6T-SRAM cell arrays, respectively. Each exponential term corresponds to the delay time of one of the four components.

We also define baseline caches in which the V_T of all the cache components is set to a low-V_T (0.2 V). Fig. 4 shows the access time and the leakage power of the baseline caches. The cache access time grows logarithmically and the leakage power increases linearly with the cache size. Those trends agree with those of earlier studies on SRAM design. In Fig. 4, we assume a direct-mapped cache organization and consider only the leakage

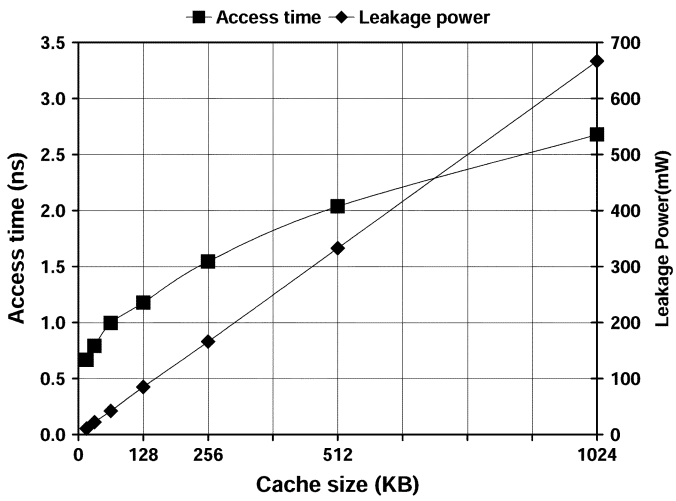

Fig. 4. Access time and leakage power versus cache size of the baseline caches.

[4]α was around 2 in submicrometer technology, but it has been decreased to about 1.3 in the current generation deep-submicrometer technology.

power of data arrays, disregarding the leakage of the tag comparators and other cache control logic.

III. Cache Leakage Optimization With Multiple V_T Assignments

A. Methodology

In this section, we present a leakage power optimization technique assuming that we can assign multiple V_T's to a cache. To find the minimum leakage power of caches using a maximum of four distinct V_T's under a specified target access time constraint, we formulate the problem as follows:

$$\min\{\mathrm{LP}(V_{T1},\ldots,V_{T4}) = A_0 + A_1 e^{-V_{T1}/a_1} + \cdots + A_4 e^{-V_{T4}/a_4}\} \quad (7)$$

$$\text{constraints}: LP(V_{T1},\ldots,V_{T4}) = B_0 + B_1 e^{-V_{T1}/a_1} + \cdots + B_4 e^{-V_{T4}/a_4},$$
$$0.2 \le V_{T1}, V_{T2}, V_3, V_4, \le 0.5 \quad (8)$$

where V_{T1}, V_{T2}, V_{T3}, and V_{T4} represent the V_T's for address bus drivers, data bus drivers, decoders, and 6T-SRAM arrays, respectively.

There exist numerous combinations of V_{T1}, V_{T2}, V_{T3}, and V_{T4} satisfying a specific target access time. Among those V_T combinations, we find a quadruple of V_{T1}, V_{T2}, V_{T3}, and V_{T4} producing minimum leakage power using a numerical optimization method (e.g., Matlab's *fmincon* function). We allowed the V_T combination that satisfies a specified access time error range within 5%. We can repeat this procedure with modified objective and constraint functions to find an optimal V_T combination for cache memories that have only two or three distinct V_T's.

Assuming that we can assign distinct V_T's to each component of the cache, it is important to determine how many V_T's are cost-effective because an extra mask and process step are needed for each additional V_T. To examine the dependence of the optimization results on access time, we sweep the target access time from the fastest possible (assigning a low V_T of 0.2 V to all the cache components) to the slowest possible (assigning a high V_T of 0.5 V to all the cache components). We present here the summary of the V_T assignment schemes we examined in this study.

- *Scheme I*: Assigning a high-V_T to all of the cache components including address bus drivers, data bus drivers, decoders, and 6T-SRAM cell arrays. This requires 2 V_T's if we include a nominal or low V_T for the processor's general logic circuits.
- *Scheme II*: Assigning a high-V_T only to the 6T-SRAM cell arrays that dominates leakage power but not the overall cache delay and assigning a default- or low-V_T (0.2 V) to the rest of the transistors. This requires at least two V_T's if we include a nominal or low V_T for the processor's logic.
- *Scheme III*: Assigning a high-V_T to the 6T-SRAM cell arrays and assigning another high-V_T to the peripheral components—address bus drivers, data bus drivers, and decoders of the cache. This requires at least three V_T's if we include a nominal or low V_T for the processor.

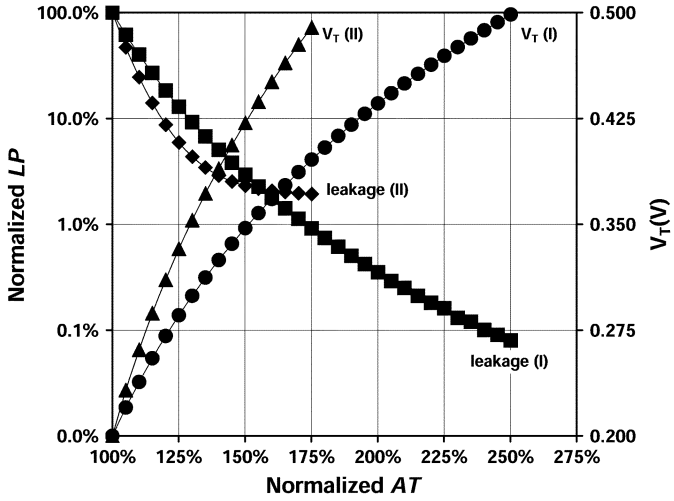

Fig. 5. Normalized optimum LP and V_T versus normalized AT of 512-KB caches—schemes I and II.

- *Scheme IV*: Assigning four distinct high V_T's to all four cache components. This requires at least five V_T's if we include a nominal or low V_T for the processor logic.

B. Leakage Power Optimization and Quantitative Tradeoff Analysis

In Fig. 5, we plot the normalized optimum leakage power and V_T at different target access times (125%, 150%, 175%, and so forth) of 512-KB caches employing schemes I and II. The optimum leakage power and the V_T are obtained using (7) and (8) of Section III-A. The parenthesized I and II in Fig. 5 represent the schemes I and II, respectively. In the graph, the normalized minimum leakage power and the access time of 100% correspond to the access time and the leakage power of a 512-KB baseline cache designed with a low V_T (0.2 V) for all four cache components—the *fasted* but *leakiest* cache. The 125% access time in the x axis means that the cache is 25% slower than the baseline cache.

According to the trends shown in Fig. 5, the leakage power decreases exponentially as the V_T increases linearly; note that the y axis is a logarithmic scale. The optimization results for the different cache sizes show almost the same normalized optimum leakage power and V_T trends as those of the 512-KB caches in Fig. 5 as long as the same V_T assignment scheme is applied; see Table IV for the normalized leakage power of all the cache sizes. Comparing two schemes—scheme I and II—the 512-KB cache with scheme II dissipates less leakage power than the one with scheme I at the same access time point when the normalized access time constraint is less than 155%. For example, at the 125% access time point, scheme II shows 6% leakage dissipation of the baseline 512-KB cache and scheme I shows 13% leakage dissipation—a 2 × difference. However, scheme I shows better leakage power reduction beyond a 155% normalized access time point.

Fig. 6 shows the normalized optimum leakage power and V_T versus normalized access time trends for a 512-KB cache of scheme III. The optimum leakage power and the V_T's are obtained using (7) and (8) of Section III-A. In Fig. 6, the V_T of

TABLE IV
PERCENTAGE LEAKAGE POWER OF SCHEMES I–IV NORMALIZED TO LEAKAGE POWER OF EACH CACHE SIZE AT THE 100% AT POINT

cache size (KB)	125% AT				150% AT				175% AT			
	I	II	III	IV	I	II	III	IV	I	II	III	IV
16	14.0	7.0	5.7	5.5	3.2	5.2	1.3	1.2	0.8	N/A	0.3	0.3
32	14.0	8.6	5.8	5.2	3.3	7.3	1.2	1.1	0.9	N/A	0.3	0.3
64	14.8	7.1	6.9	6.7	3.4	3.2	1.6	1.5	0.9	2.9	0.5	0.4
128	14.5	6.7	6.1	5.7	3.4	4.2	1.4	1.3	1.0	N/A	0.4	0.4
256	13.5	7.8	7.8	7.7	3.0	2.3	1.7	1.6	0.8	1.7	0.5	0.5
512	12.8	6.0	5.9	5.7	2.8	2.3	1.2	1.2	0.7	2.0	0.3	0.3
1024	12.0	3.9	3.8	3.7	2.7	2.3	0.7	0.7	0.7	N/A	0.2	0.2

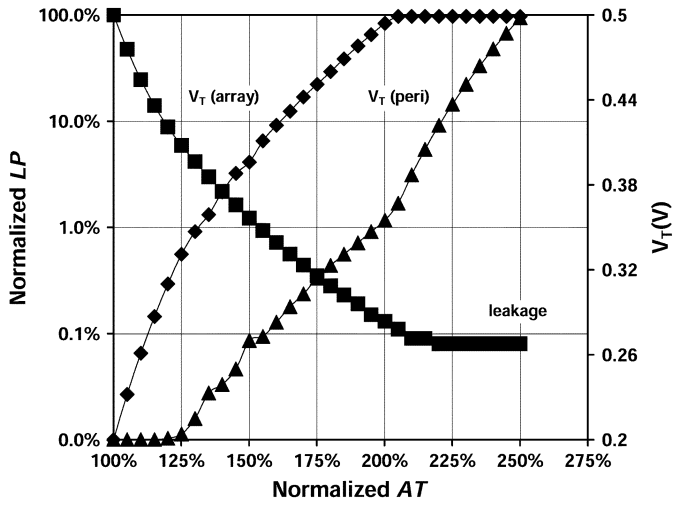

Fig. 6. Normalized optimum LP and V_T versus normalized AT of 512-KB caches—schemes I and III.

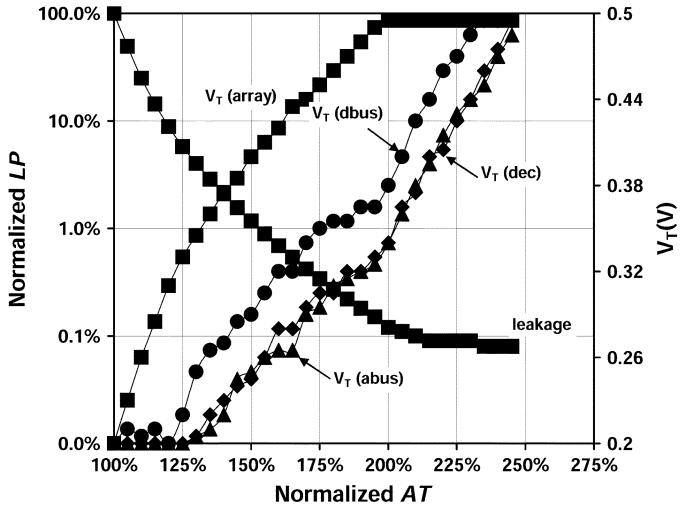

Fig. 7. Normalized optimum LP and V_T versus normalized AT—scheme IV.

the SRAM cell array, denoted as *array* in the graph, starts to increase first. This implies that the SRAM cell array is responsible for the most significant fraction of total cache leakage power, but it has the least impact on increasing the total cache access time. After the V_T of the SRAM cell arrays are saturated to the maximum allowed point (0.5 V), the V_T of the peripheral components labeled as *peri* in the graph is increased further to reduce further leakage power in the peripheral components. However, this just increases the access time without much further cache leakage reduction. For example, the leakage power is not decreased over the 215% access time point where the V_T for the peripheral circuit has not reached the maximum value (0.5 V) in this 512-KB cache case.

This leakage power and V_T versus access time trends also explain the leakage optimization results shown in Fig. 5: scheme II shows a better optimization result than scheme I does when the normalized access time is less than 155%, but it does not beyond 155% access time point. Recall that scheme I assigns a high-V_T to all the cache components. It sacrifices more access time unnecessarily by increasing the V_T of the peripheral components with little leakage reduction at the same access time. However, scheme II assigns the high-V_T to just the SRAM cell arrays that are responsible for a greater fraction of total cache leakage power but affects access time less. However, scheme II cannot reduce leakage power beyond the 155% access time point, because the leakage power of the peripheral components, where a low V_T is used, becomes substantial beyond this point.

Fig. 7 shows the normalized optimum leakage power and V_T versus normalized access time trends for a 512-KB cache of scheme IV. The optimum leakage power and the V_T's are obtained again using (7) and (8) of Section III-A. In scheme IV, we can assign up to 4 distinct V_T's for leakage power optimization. According to the results shown in Fig. 7, the V_T of the 6T-SRAM cell arrays starts to increase first similar to the scheme III case. Among the peripheral components, the V_T for the data bus starts to increase first. This implies that the data bus consisting of 128 b—the assumed bus width between the L2 and L1 caches—has the second most significant impact on the leakage power. Even though the address bus has the same structure, the number of bits in the address bus is much smaller than the data bus. Hence, the leakage power impact of the address bus much less than the data bus. However, in the case of smaller caches (e.g., 16–64 KB caches) where the data bus width is 32 b, both the data and address bus have almost the same impact on the leakage power. Therefore, the V_T trends for both the data and address buses will be the same. These V_T trends suggest the direction of V_T optimizations that reduce cache leakage power.

Table IV summarizes the normalized cache leakage power of schemes I–IV. As expected, we can reduce more leakage power

TABLE V
CACHE DYNAMIC ENERGY CONSUMPTION PER ACCESS AND LEAKAGE POWER DISSIPATION AT 70 °C DIE TEMPERATURE AND A TYPICAL CORNER FOR EACH CACHE SIZE

	Cache Size (KB)						
	16	32	64	128	256	512	1024
Dynamic energy / access (pJ)	10.9	14.5	30.4	51.3	139.6	159.9	201.3
Static leakage power (mW)	10.7	22.1	41.9	85.0	165.8	332.7	666.7

while achieving the same access time by having more V_T's to control. If the access time is fixed, the caches of schemes III and IV always show 38%–72% better leakage optimization results than those of scheme I. There are a few things we should note from this comparison study. First, as the target access time is increased to more than the 150% point in scheme II, caches dissipate more leakage power than those employing scheme I. This implies that the cache peripheral components consume nonnegligible leakage power. The leakage power of those components becomes substantial when we cut down the leakage power of the 6T-SRAM arrays significantly. Second, the slowest cache access time of scheme II ends around 150% in small-size caches. This means that the peripheral components also play important roles in both cache leakage power and access time. In other words, increasing the V_T of 6T-SRAM cell arrays alone gives us diminishing returns at some point without reducing the leakage power further. This is why the caches of scheme I give even better results than those of scheme II as V_T increases. Finally, there is a negligible difference between caches of schemes III and IV in terms of leakage power reduction. This implies that scheme III employing two *distinct high* V_T's—three V_T's if we include a nominal or low V_T for the processor—is enough to minimize leakage. Finally, as illustrated in Figs. 5–7 and Table IV, each cache shows a wide range of optimal leakage power consumption depending on target access time. Hence, the right tradeoff point between the leakage power and the access time of the caches will be determined by either system design specifications or constraints.

IV. LEAKAGE OPTIMIZATION TECHNIQUES FOR TWO-LEVEL CACHES

A. Methodology

In a processor memory system, the average memory access time (AMAT) [18] is a key metric for measuring the overall memory system performance. To evaluate the performance or AMAT, it is essential to examine the cache miss characteristics of realistic applications, because the performance or AMAT is a function of L1 and L2 cache miss rates and cache access times. In our study, we assume that the memory system hierarchy consists of separate L1 instruction and data caches with a unified L2 cache. Then, the average performance of the processor memory system can be measured or compared with the AMAT represented by

$$\begin{aligned}&\text{Average Memory Access Time}\\&\quad=\text{HitTime}_{L1}+\text{MissRate}_{L1}\\&\qquad\times(\text{Hit Time}_{L2}+\text{MissRate}_{L2}\times\text{MissPenalty}_{L2})\end{aligned}\tag{9}$$

where HitTime_{L1} and HitTime_{L2} are the access time of L1 and L2 caches, Miss Rate_{L1} and Miss Rate_{L2} are the miss rate of L1 and L2 caches, and Miss Penalty_{L2} is the external memory access and data transfer time. Note that the local miss rate[5] is used as the Miss Rate_{L2}.

Similarly, we measure the average memory access energy (AMAE) to compare the dynamic energy dissipation of each memory system configuration. 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 average memory access energy, as follow:

$$\begin{aligned}&\text{Average Memory Access Energy}\\&\quad=\text{HitEnergy}_{L1}+\text{MissRate}_{L1}\\&\qquad\times(\text{Hit Energy}_{L2}+\text{MissRate}_{L2}\times\text{MissEnergy}_{L2})\end{aligned}\tag{10}$$

where Hit Energy is average energy dissipation per access given in Table V. We assume a two-channel 1066-MHz 256-MB RAMBUS DRAM RIMM whose sustained transfer rate is 4.2 GB/s [19] to derive the main memory access time and dynamic energy dissipation per access. Though the sustained transfer rate is quite high, we should also consider the RAS/CAS latency of the memory, which is about 20 ns. For the energy dissipation per access, we used the number given in [20], which is 3.57 nJ per access. The dynamic energy dissipation per access can vary depending on the number of RIMMs. We assume that one RIMM is installed. See Section IV–B and note that more RIMMs are favorable for our optimization technique, because our technique prefers a larger L2 cache to a smaller one for leakage power reduction. The larger L2 cache accesses DRAM less frequently than the smaller one, resulting in less energy consumption for accessing the external DRAM. Hence, if more RIMM modules are installed implying more energy dissipation per DRAM access, a larger L2 cache will allow even more energy to be saved.

To obtain L1 and L2 cache miss rates, we use the SimpleScalar/Alpha 3.0 tool set [21], which is a suite of functional and timing simulation tools for the Alpha AXP ISA. In addition, we collected the results from all 25 of the SPEC2K benchmarks [15] to perform our evaluation. All SPEC programs were compiled for a *Compaq* Alpha AXP-21 264 processor using the Compaq C and Fortran compilers under the OSF/1 V4.0 operating system using full compiler optimizations ($-O4$). We completed the execution for each benchmark application to get reliable L2 cache miss rates, because L2 cache accesses are far less frequent than

[5]This rate is simply the number of misses in a cache divided by the total number of memory accesses to this cache.

TABLE VI
AVERAGE L1 AND L2 CACHE MISS RATES FROM THE ENTIRE SPEC2K BENCHMARKS

L1 size (KB)	miss rate (%)	L2 size (KB)	miss rate (%)
16	3.3%	128KB	34.2%
		256KB	32.2%
		512KB	30.6%
		1024KB	25.5%
32	2.5%	256KB	40.3%
		512KB	38.2%
		1024KB	31.8%
64	1.5%	512KB	41.6%
		1024KB	35.6%

L1 cache accesses; an insufficient number of L2 accesses may result in unrepresentatively higher L2 cache miss rates.

Table VI shows the average L1 and L2 cache miss rates from the entire SPEC2 K benchmarks for 16-, 32-, and 64-KB L1 caches, respectively. We used direct-mapped L1 instruction caches and four-way set associative L1 data caches. Also, we used eight-way set associative L2 caches. For simplicity, each L1 cache miss rate is obtained by taking the sum of the number of total instruction and data cache misses and dividing by the sum of total instruction and data cache accesses; a 16-KB L1 means instruction and data caches are each 16 KB in size. Since an L2 miss rate is a function of the L1 cache miss rate, we measure the separate L2 cache miss rates for each L1 cache size configuration. Those cache miss characteristics will definitely affect the leakage optimization direction of two-level cache memory systems.

B. L2 Cache Leakage Power Optimization

Since an L2 cache's contribution to leakage power dominates due to their size, we will examine the leakage power optimization of the L2 cache first. Consider caches designed with low-V_T (0.2 V) devices and a baseline cache memory system consisting of 16 and 128 KB for L1 and L2 caches, respectively. Then, we have leakage power consumption and AMAT corresponding to this configuration. Increasing V_T of the 128-KB L2 cache will reduce the leakage power of the L2 cache, but it will increase the AMAT of the cache memory system because of the increased access or hit time. However, there is a way to reduce the leakage power of the cache memory system without increasing the AMAT that significantly impacts on the execution time of the system.

The key to reducing leakage power without increasing AMAT is to compensate for the increased L2 access time by reducing the cache miss rate of the cache memory system. To reduce the miss rate, we can increase the L2 cache size. The main memory access penalty is quite significant in term of both time and energy. Hence, even a slight reduction of L2 cache miss rates results in a significant improvement in 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]. In this argument, we assume that the same AMAT will approximately give us the same execution time for a fixed processor core, L1 cache size, and benchmark program, so that we can fairly compare the total leakage energy consumption as well.

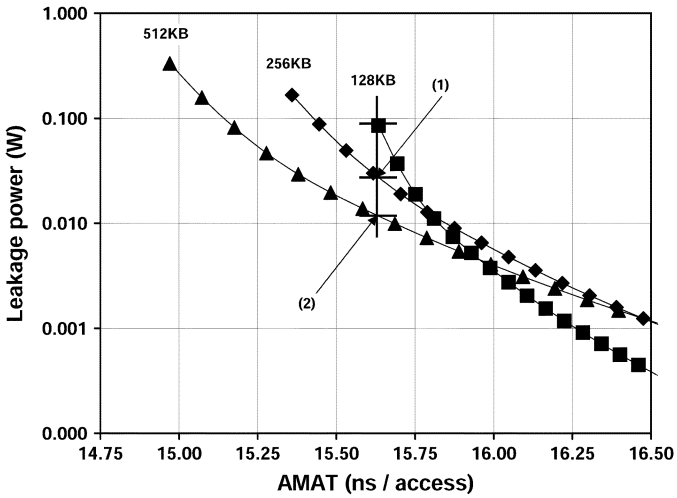

Fig. 8. L2 leakage power optimization at a fixed L1 size (16 KB). (1) and (2) are the leakage power consumption of the 256- and 512-KB caches at the same AMAT as the baseline 128-KB cache, respectively.

Fig. 8 shows the leakage power versus AMAT of L2 caches with a fixed L1 cache size—16 KB. The leakage power optimization for individual caches is based on scheme III that requires two additional distinct high V_T's for L2. Assuming the AMAT of the fastest 128-KB L2 cache designed with low-V_T (0.2 V) as a baseline, we compare the leakage power of other caches at the same AMAT point; see the (1) and (2) points in Fig. 8. The (1) and (2) points are the leakage power consumption of the cache system with the 256- and 512-KB caches at the same AMAT as the baseline 128-KB cache system. As can be seen from the plots, the AMAT can be maintained while the leakage power can be reduced by replacing the baseline 128-KB L2 cache with a 256-KB L2 cache that is intentionally slowed down by increasing its V_T's to reduce leakage.

This replacement with the double-sized L2 cache reduces the leakage power by 70% compared to the fastest but leakiest 128-KB L2 cache with the same AMAT. Similarly, the use of a 512-KB L2 cache can further reduce leakage compared to the 256-KB cache; see the vertical line in Fig. 8.

Finally, the employment of larger L2 caches also reduces the average dynamic power of the memory system, because the larger L2 caches reduce the number of external memory accesses that consume a significant amount of dynamic energy. Table VII summarizes the results for the normalized leakage power and normalized average memory access energy 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: 128-KB L2 with 16-KB L1, 256-KB L2 with 32-KB L1, and 512-KB L2 with 64-KB L1. The shaded numbers represent the baseline L2 configuration, leakage power, and AMAE. Table VII shows the counterintuitive results that we can reduce both leakage power and AMAE by employing larger L2 caches while maintaining the same AMAT.

TABLE VII
L2 CACHE NORMALIZED LEAKAGE AND AMAE AT THE FIXED L1 SIZE (16 KB) AND AMAT

L1 size (KB)	L2 size (KB)	normalized leakage (%)	normalized AMAE (%)
16	128	100	100
	256	31	101
	512	15	99
32	256	100	100
	512	11	98
	1024	~0	89
64	512	100	100
	1024	1	95

TABLE VIII
L1 CACHE NORMALIZED LEAKAGE AND AMAE AT THE FIXED L2 SIZE (512 KB) AND AMAT

L2 size (KB)	L1 size (KB)	normalized leakage (%)	normalized AMAE (%)
256	32	100	100
	16	6	95
512	64	100	100
	32	19	62
	16	2	59
1024	64	100	100
	32	12	64
	16	2	62

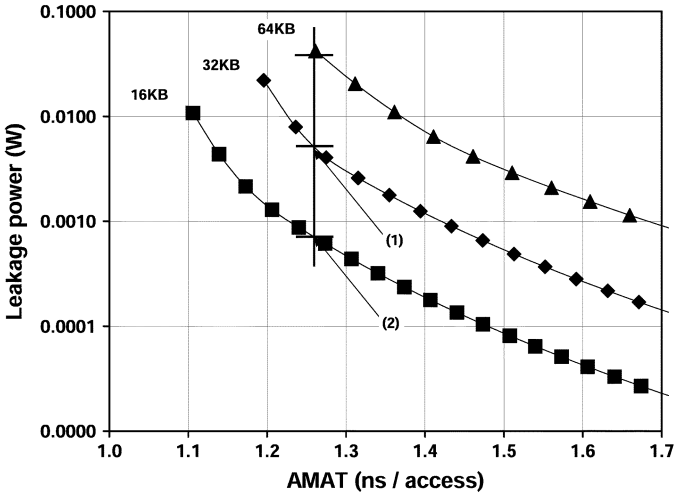

Fig. 9. L1 leakage power optimization at a fixed L2 size (512 KB). (1) and (2) are the leakage power consumption of the 32- and 16-KB caches at the same AMAT as the baseline 64-KB cache, respectively.

C. L1 Cache Leakage Power Optimization

It is rather difficult to improve the L1 cache miss rates further, because they are already very low for 16-, 32-, and 64-KB caches in the case when SPEC2K benchmarks are run. Hence, the access time of caches become a dominant factor in determining the AMAT. For example, the access time of a 64-KB L1 cache increases by 48% compared to the fastest 16-KB L1 cache, because the access time is very sensitive to size in small caches. Essentially, cache access time increases logarithmically with size, but has a steeper slope for smaller caches than for larger caches. This observation explains 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., 16 or 64 KB).

Fig. 9 shows the leakage power versus the AMAT of 16-, 32-, and 64-KB L1 caches using scheme III each with a fixed L2 cache of size 512 KB. Like the comparison performed in Section IV–B, the leakage power of different caches is compared at the same AMAT point. The plots show that leakage power can be reduced by replacing the fastest 64-KB L1 cache with a 32-KB L1 cache that is intentionally slowed down by increasing its V_T's to reduce the leakage power—the resulting cache memory system still has the same AMAT. Similarly, a slowed 16-KB cache with increased V_T's can replace a 32-KB cache without changing the AMAT of the L1/L2 hierarchy. The new system consumes much less leakage power; see points (1) and (2) in Fig. 9, which are the leakage power consumption of the cache system with the 32- and 16-KB caches at the same AMAT as the baseline cache system.

Table VIII shows the results for normalized leakage power and AMAE as a percentage of each fast but leaky L1 cache size using scheme III with fixed AMATs. The comparisons were performed in the same manner as Table VII. The shaded numbers represent the baseline L1 configuration, leakage power, and AMAE. According to the comparisons, we can reduce both leakage power and AMAE by employing smaller L1 caches. This is the inverse of the case for L2 caches, where the leakage of the overall memory system can be reduced by increasing their size. However, it should be noted that these results are only valid within the specific set of sizes and simulation environment given in this discussion. First, a 4-KB L1 cache will have a cache miss rate that is much higher than a 16-KB cache, but its access time will not be sufficiently smaller to make the tradeoff worthwhile. Also, the normalized AMAE is rather high because the total power fraction of L1 caches is relatively small compared to L2 caches. Second, many SPEC2K benchmark programs have very high locality compared to real-world larger size applications. This results in quite low cache miss rates for small-size L1 caches as shown in Table VI. Third, the operating system (OS) context switching was not modeled due to our limited simulation environment. The context switching typically increases cache miss rates, because cache flushing increases cold start misses. These factors must be considered if one is to perform realistic cache leakage power optimizations with the proposed techniques.

V. CONCLUSION

In this study, we examined the leakage power and access time tradeoff for caches where multiple V_T's are allowed. We used curve fitting techniques to model subthreshold leakage power and access time. Our results show that two extra distinct high V_T's for caches—3 V_T's including the V_T for the microprocessor core logic—are sufficient to yield a significant reduction in leakage power. Such an arrangement can reduce the leakage

power by as much as 91%. 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 the average memory access time. Given that the processor core may need a distinct V_T, and each of the caches may need up to two V_T's (scheme III) we could require up to five distinct V_T's for the leakage power optimization of two-level cache memory systems.

Even though the modeling and optimization techniques presented in this study have been performed using continuous-domain functions, the actual cache latencies are integer numbers of processor clock cycles. Cache designers or architects can choose an appropriate discrete point from the continuous-domain results depending on their target processor core clock frequency.

Furthermore, the circuit techniques combined with microarchitectural level controls exemplified by *drowsy* caches [10] are designed to reduce the leakage power of L1 caches when sacrificing access time is not an option. Such an approach is less attractive for L2 caches. The same effect can be obtained more simply by using high-V_T circuits.

REFERENCES

[1] N. S. Kim *et al.*, "Leakage current: Moore's law meets static power," *IEEE Computer* , vol. 36, no. 12, pp. 68–75, Dec. 2003.

[2] G. Sery, S. Borkar, and V. De, "Life is CMOS: Why chase life after?," in *Proc. IEEE Design Automation Conf.*, 2002, pp. 78–83.

[3] S. Mutoh *et al.*, "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *IEEE J. Solid-State Circuits*, vol. 30, no. 8, pp. 847–854, Aug. 1995.

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

[5] K. Nii *et al.*, "A low power SRAM using auto-backgate-controlled MT-CMOS," in *Proc. IEEE Int. Symp. Low Power Electronic Device*, 1998, pp. 293–298.

[6] 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, no. 11, pp. 1492–1500, Nov. 1999.

[7] F. Hamzaoglu *et al.*, "Analysis of dual-V_T SRAM cells with full-swing single-ended bit line sensing for on-chip cache," *IEEE Trans. Very Large Scale (VLSI) Syst.*, vol. 10, no. 2, pp. 91–95, Apr. 2002.

[8] M. Powell *et al.*, "Gated-$V_{\rm DD}$: A circuit technique to reduce leakage in deep-submicron cache memories," in *Proc. IEEE Int. Symp. Lower Power Electronics & Design*, 2000, pp. 90–95.

[9] A. Agarwal, L. Hai, and K. Roy, "A single-V_T low-leakage gated-ground cache for deep submicron," *IEEE J. Solid-State Circuits*, vol. 38, no. 2, pp. 319–328, Feb. 2003.

[10] N. S. Kim *et al.*, "Drowsy instruction caches," in *Proc. IEEE Int. Symp. Microarchitecture*, 2002, pp. 219–230.

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

[12] S. Kaxiras *et al.*, "Cache decay: Exploiting generational behavior to reduce cache leakage power," in *Proc. IEEE Int. Symp. Computer Architecture*, 2001, pp. 240–251.

[13] H. Zhou *et al.*, "Adaptive mode-control: A static-power-efficient cache design," in *Proc. IEEE Parallel Architecture and Compilation Tech.*, 2001, pp. 61–70.

[14] N. S. Kim *et al.*, "Leakage power optimization techniques for ultra deep sub-micron multi-level caches," in *Proc. IEEE Int. Conf. Computer Aided Design*, 2003, pp. 627–632.

[15] Standard Performance Evaluation Corporation [Online]. Available: http://www.specbench.org

[16] S. Wilton *et al.*, "An Enhanced Access and Cycle Time Model for On-Chip Caches,", Western Res. Lab. Res. Rep. 93/5, 1993.

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

[18] J. Hennessy *et al.*, *Computer Architecture—A Quantitative Approach*, 3rd ed. San Mateo, CA: Morgan Kaufmann, 2003, pp. 406–408.

[19] 800/1066 MHz RDRAM Advanced Information (2002). [Online]. Available: http://www.rambus.com

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

[21] T. Austin *et al.*, "SimpleScalar: An infrastructure for computer system modeling," *IEEE Computer*, vol. 35, no. 2, pp. 59–67, Feb. 2002.

[22] T. Mudge, "Power: A first class design constraint," *IEEE Computer*, vol. 34, no. 4, pp. 52–57, Apr. 2001.

Nam Sung Kim received the B.S. and M.S. degrees in electrical engineering from Korea Advanced Institute of Science and Technology, Taejon, Korea, in 1997 and 2000, respectively, and the Ph.D. degree in computer science and engineering from the University of Michigan, Ann Arbor, in 2004.

He is a Senior Research Scientist of the Circuit Research Laboratory with Intel Corporation's Microprocessor Technology Laboratories, Hillsboro, OR. He has authored more than 20 technical papers in refereed international conferences and journals. His research interests span low-power digital circuits and processor microarchitectures, circuit and microarchitecture co-designs, and computer-aided designs for variation-aware digital systems.

Dr. Kim was the recipient of the third place award at the 2001 IEEE Design Automation Conference (DAC) Student Design Contest in Las Vegas, NV, and the Best Paper Award at the 2003 IEEE International Conference on Microarchitecture (MICRO) in San Diego, CA, for his work on the low-power and robust microarchitecture. He was also the recipient of the Intel Fellowship in 2002.

David Blaauw received the B.S. degree in physics and computer science from Duke University, Chapel Hill, NC, in 1986, and the M.S. and Ph.D. degrees in computer science from the University of Illinois, Urbana, in 1988 and 1991, respectively.

He was with IBM Corporation as a Development Staff Member until August 1993. From 1993 until August 2001, he was with or Motorola Inc., Austin, TX, were he was the Manager of the High Performance Design Technology Group. Since August 2001, he has been on the faculty at the University of Michigan as an Associate Professor. His work has focused on VLSI design and computer-aided design with particular emphasis on circuit design and optimization for high-performance and low-power designs.

Prof. Blaauw was the Technical Program Chair and General Chair for the International Symposium on Low Power Electronic and Design in 1999 and 2000, respectively, and was the Technical Program Co-Chair and member of the Executive Committee the ACM/IEEE Design Automation Conference in 2000 and 2001.

Trevor Mudge (S'74–M'77–SM'84–F'95) received the Ph.D. degree in computer science from the University of Illinois, Urbana, in 1977.

Since then, he has been on the faculty of the University of Michigan, Ann Arbor. Three years ago, he was named the first Bredt Family Professor of Electrical Engineering and Computer Science after concluding a ten-year term as the Director of the Advanced Computer Architecture Laboratory—a group of a dozen faculty and about 80 graduate students. He is the author of numerous papers on computer architecture, programming languages, VLSI design, and computer vision. He has also chaired 33 theses in these research areas. His research interests include computer architecture, computer-aided design, and compilers. In addition to his position as a faculty member, he runs Idiot Savants, Ann Arbor, MI, a chip design consultancy.

Prof. Mudge is a member of the Association of Computing Machinery, the Institute of Electrical Engineers, U.K., and the British Computer Society.