Total Leakage Optimization Strategies for Multi-Level Caches

Robert Bai[1], Nam-Sung Kim[2], Dennis Sylvester[1], Trevor Mudge[1]
[1] University of Michigan, EECS Department, Ann Arbor, MI 48109 {rbai, dennis, tnm} @eecs.umich.edu
[2] Intel Corporation, Portland, Oregon, nam.sung.kim@intel.com

Abstract

Gate leakage current is fast becoming a major contributor to total leakage and will become the dominant leakage mechanism as gate oxide is scaled below 10Å. This has special relevance for caches, because they are often the largest component by area in state-of-the-art microprocessors, and leakage is their major contribution to overall chip power. In this paper, we investigate the impact of T_{ox} and V_{th} on power performance trade-offs for on-chip caches. We examine the optimization of the various components of a single level cache and then extend this to two level cache systems. In addition to leakage, our studies also account for the dynamic power expended as a result of cache misses. Our results show that, surprisingly, one can often reduce overall power by increasing the size of the L2 cache if we only allow one pair of V_{th}/T_{ox} in L2. We further show that two V_{th}'s and two T_{ox}'s are sufficient to get close to an optimal solution and that V_{th} is generally a better design knob than T_{ox} for leakage optimization, thus it is better to restrict the number of T_{ox}'s rather than V_{th}'s if cost is a concern. Finally, we show that optimal power performance points are remarkably robust to wide changes in ambient temperature.

Categories and Subject Descriptors:
B.3.2 [Cache Memories]: Design optimization for low power
General Terms: Design, Measurement, Performance
Keywords: Cache memory, Low power, Gate leakage

1. Introduction

Gate leakage current due to gate direct tunneling is quickly becoming an important issue in the total chip power dissipation. The gate leakage is only going to increase as we continue to scale down the thickness of gate oxide and eventually gate leakage will exceed subthreshold leakage since the gate oxide thickness is being scaled at a faster rate than the threshold voltage. According to ITRS projections [1], gate leakage has been identified as one of the challenges to future device scaling as the gate oxide will be scaled by 2007 to below 10Å. Before any successful market deployment of *high-k* dielectrics to replace gate oxide, we need to tackle the ever-increasing gate leakage problem. There have been several previous studies on cache leakage power reduction [2-8]; all of them focused on subthreshold leakage power. However, with aggressive T_{ox} scaling, gate leakage power can potentially surpass the subthreshold leakage at low T_{ox}. In this paper, we investigate various techniques to minimize total (gate + subthreshold) leakage power plus dynamic power under delay constraints for an entire microprocessor memory system consisting of L1, L2 cache and main memory. We present systematic approaches to assigning values for T_{ox} and V_{th} to minimize total leakage plus dynamic energy consumption. Although our study is limited to hierarchies consisting of L1, L2 caches, and main memory, it is readily extended to systems with more cache levels.

GLSVLSI'05, April 17–19, 2005, Chicago, Illinois, USA.

2. Circuit Evaluation Methodology

2.1. Technology File Generation

For our experiment, we have used the technology files from Berkeley Predictive Technology Model (BPTM) for a 65nm technology node [9]. We then characterize the technology files for a range of V_{th} and T_{ox} values. We let V_{th} vary from 0.2V to 0.5V, while allowing T_{ox} to scale from 10Å to 14Å. The lower limits of these ranges are chosen to reflect typical values of high-performance logic for the studied technology node. Such transistors would be required for the non-memory portion of a processor or system. While there is no physical reason for a V_{th} upper bound, we expect that values above 0.5V are unlikely in 65nm technology with approximately 1V supply. The upper limit we use for T_{ox} is determined primarily by the fact that the channel length is also scaled proportionately (see next section) — using Tox > 14 Å leads to device topologies that are vastly different than the baseline high-performance logic transistor in both horizontal and vertical dimensions, which may pose fabrication difficulties. Since changing T_{ox} can also alter the subthreshold leakage current, we have characterized the technology files such that at a given V_{th}, the same subthreshold leakage current, I_{sub}, is maintained across the entire range of T_{ox}. This is accomplished by adjusting the V_{th0} parameters in the HSPICE model file.

2.2. L_{drawn} and W_{drawn} Scaling

The increase of T_{ox} while maintaining the same drawn channel length may cause the gate terminal to lose control of the conduction state of the channel due to *drain induced barrier lowering* (DIBL). Hence, when T_{ox} changes, the drawn channel length must be scaled appropriately by ensuring the following relationship is satisfied [10]:

$$\frac{L_{eff}(new)}{L_{eff}(old)} = \frac{T_{ox}(new)}{T_{ox}(old)}$$

In order to maintain memory cell stability, the widths of the transistors in the memory cell need to be adjusted proportionately with the change in the drawn channel lengths. Thus the impact of T_{ox} scaling on the cell area has to be taken into account as the cell will grow in both horizontal and vertical dimensions. Figure 1 shows a stick-level diagram of a 6T-cell representing two cross-coupled inverter pairs and two access transistors when T_{ox} is 10Å and 14Å, respectively. [Note that the drawings are not drawn to scale] As can be seen, the additional layout overhead in the memory cell because of an increase of 4 Å in gate oxide thickness is approximately 7%.

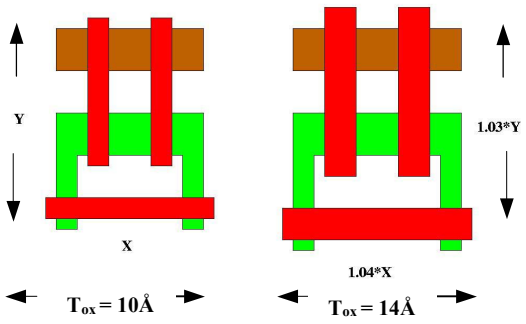


Figure 1. Stick Diagrams of 6T-cell at Two Different T_{ox}'s

2.3. Delay and Leakage Power Models

First we have designed the cache netlists to target for the 65nm technology node. We assume that internally, the cache consists of four components: 1) memory cell array and sense amplifier; 2) decoder: 3) address bus drivers; and 4) data bus drivers. Second, to systematically model the impact of T_{ox} and V_{th} on delay and leakage, we derive a set of analytical equations by applying curve-fitting techniques to the HSPICE simulation results of the cache netlists. It is observed that the total leakage current of a 16KB memory cell array is exponentially dependent on T_{ox} and V_{th}. Hence, we approximate the total leakage power as follows:

$$P_{total}(V_{th}, T_{ox}) = A_0 + A_1 * e^{a_1 * V_{th}} + A_2 * e^{a_2 * T_{ox}}$$

However, the delay of the array is shown to be linear with T_{ox} and over the range of our interest its dependence on V_{th} can be approximated to an exponential growth function with very small exponents as follows:

$$T_d(V_{th}, T_{ox}) = k_0 + k_1 * e^{(k_3 * V_{th})} + k_2 * T_{ox}$$

Although these total leakage and delay models are developed for the memory cell array, the same trends also hold for the rest of cache memory components — decoders and address/data bus drivers. Therefore, we can model the total leakage and delay of each component in the same way as we do for the memory cell array assuming that both total leakage and delay of each component is independent from one another. Then, we can approximate both the total leakage and the delay of a cache system by summing up the leakage and delay of each cache component.

3. Single Cache Leakage Optimization

To examine the dependence of leakage power on V_{th} and T_{ox} assignment, we study three different V_{th}/T_{ox} assignment schemes:

- Scheme I: assign independent V_{th}'s and T_{ox}'s to each cache component.
- Scheme II: assign a V_{th}/T_{ox} pair to the memory cell array and another pair to the remaining three cache components.
- Scheme III: assign the same V_{th}/T_{ox} pair to all four cache components.

We formulate the problem of minimizing the leakage power given the delay constraint as the following optimization problem [11]:

$$\textbf{\textit{Minimize}}\ LeakagePower(V_{th1}, T_{ox1,...} V_{th4}, T_{ox4})$$
$$= A_0 + A_1 * e^{a_1 * V_{th1}} + A_2 * e^{a_2 * T_{ox1}} + \ldots + A_7 * e^{a_7 * V_{th4}} + A_8 * e^{a_8 * T_{ox4}}$$
$$\textbf{\textit{Subject to}}\ T_d(V_{th1}, T_{ox1,...,} V_{th4}, T_{ox4}) =$$
$$B_0 + B_1 * e^{(b_1 * V_{th1})} + B_2 * T_{ox1} + \ldots + B_7 * e^{(b_4 * V_{th4})} + B_8 * T_{ox4};$$
$$10\text{Å} \le T_{ox1}, T_{ox2}, T_{ox3}, T_{ox4} \le 14\text{Å};$$
$$0.2\text{V} \le V_{th1}, V_{th2}, V_{th3}, V_{th4} \le 0.5\text{V};$$

We have restricted the ranges of V_{th} to be between 0.2V and 0.5V, and those of T_{ox} between 10Å and 14Å. This will have implications on the optimization results as will be seen in Section 5. In addition, in order to make our optimization procedure more tractable, we have chosen V_{th} and T_{ox} to take on discrete values with small step size. Figure 2 shows the results of the optimization performed on a 16KB cache. As expected, scheme III is the worst performer, and scheme I is the best. However, scheme II is only slightly behind scheme I for the same delay constraint, but from a process standpoint, scheme I is more costly than scheme II. Therefore, it is the preferred scheme, as it is not only economically feasible but also achieves close to optimal leakage. In addition, the results show that memory cell arrays always consume the most leakage regardless of the assignment scheme, followed by the word address decoders, with address and data buffers consuming the least amount of leakage. This is to be expected given the fact that the cell arrays contain the largest number of transistors in the cache memory and most of them are in their "off" states. In Table 1, we show the optimized V_{th}/T_{ox} assignments for scheme I, II and III, respectively. It is worth noting that in scheme I and II, high values of V_{th} and thick T_{ox}'s are always assigned to the memory cell arrays, and V_{th}/T_{ox} in the peripheral components have been set sufficiently low to help meet the delay target.

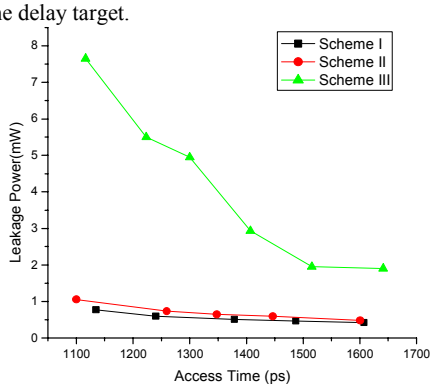


Figure 2. Leakage vs. Access Time for 16KB cache

Table 1. Optimized V_{th}/T_{ox} Assignments for Scheme I, II, III

Scheme I				Scheme II		Scheme III
Memory Cell Array	Address Decoder	Address Bus Driver	Data Bus Driver	Memory Cell Array	Peripheral Components	All Components
0.4/14	0.3/12	0.2/10	0.2/10	0.4/14	0.25/10	0.25/12
0.45/14	0.3/13	0.2/10	0.2/10	0.45/14	0.25/11	0.25/13
0.5/14	0.35/13	0.25/10	0.25/10	0.45/14	0.25/12	0.3/13
0.5/14	0.35/13	0.25/10	0.25/10	0.5/14	0.25/12	0.3/14
0.5/14	0.35/14	0.25/12	0.25/11	0.5/14	0.3/12	0.35/13

To gain further insight into the selection of the decision variables during the optimization process, we perform an experiment in which for a 16KB cache we hold either V_{th} or T_{ox} constant, and at the same time observe how leakage power is impacted by the other decision variable independently. In Figure 3, we show four curves, two of which are constructed by fixing T_{ox} at 10Å and 14Å, respectively, and the other two are created by fixing V_{th} at 0.2V and 0.4V, respectively. It is evident that the leakage is more sensitive to T_{ox} than V_{th}, and the delay doesn't show as wide a range when V_{th} is fixed as when T_{ox} is fixed. Hence, to achieve minimum overall leakage, it is best to set T_{ox} conservatively at a high value and let V_{th} be the knob designers can vary to meet a delay constraint.

4. Two-Level Cache Leakage Optimization

Having characterized a single cache, we now turn our attention to optimizing the leakage-performance tradeoff in two-level caches. For a two-level cache system, Average Memory Access Time (AMAT) is defined as follows [12]:

$$AMAT = Hit\ Time_{L1} + Miss\ Rate_{L1} \times (Hit\ Time_{L2} + Miss\ Rate_{L2} \times Main\ Memory\ Access\ Time)$$

To estimate AMAT, we need cache access statistics for each L1 and L2 cache size combination, which can only be derived from architectural simulations. The architectural simulator used in this study to obtain the cache access statistics are measured using the SimpleScalar/Alpha 3.0 tool set, a suite of functional and timing simulation tools for the Alpha AXP ISA; see Table 2 for the processor simulation parameters. The processor micro-architectural parameters model a high-end microprocessor similar to an Alpha 21264.

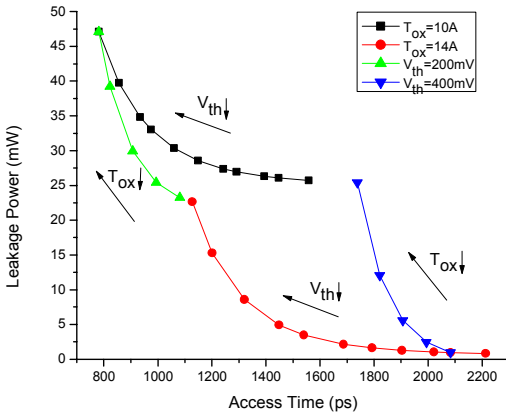


Figure 3. Fixed V_{th} vs. Fixed T_{ox}

To perform our evaluation we collected results from various benchmark suites such as SPEC2000, SPECWEB, TPC/C, etc.

Table 2. Processor simulation parameters

(The numbers in parentheses are the latencies of the respective units.)

Out of Order Execution	4-wide fetch / decode / issue commit, 64 RUU, 32 LSQ, and speculative scheduling
Functional Unit (latencies)	4 integer ALUs (1), 2 floating point ALUs (2), 1 integer MULT/DIV (3/20), 1 floating point MULT/DIV/SQRT (4/12/24), and 2 general memory port
Branch Prediction	Combined bimodal (1K-entry) / gshare (1K-entry) w/ selector (1K-entry), 32-entry RAS, 512-entry 4-way BTB, and 11-cycle mis-prediction recovery
Memory System (latencies)	16~64KB 2-way 32-byte block L1 inst (1) and data (1) caches, 128~1024KB 4-way 64-byte block unified L2 caches, 128-entry fully associative inst and data TLB (28/28), and main memory (82/8)

L2 Cache Leakage Power Optimization. Due to its size, L2 caches naturally consume much more leakage than L1 caches. Therefore, it is useful to investigate methods that aim at reducing leakage power in L2. In the first of our experiments, we fix the size of an L1 cache and assign the default V_{th} and T_{ox} to the L1 cache, and then proceed to see which L2 organization would yield better leakage whilst still meeting the same average memory access time (AMAT) constraint of the two-level cache system. For example, we can adjust both V_{th} and T_{ox} knobs to make two different size caches that have the same AMAT: note that the AMAT is a function of both the cache miss rate and access (hit) time. In the first part of our analysis, we assign a V_{th}/T_{ox} pair to the entire L2 cache memory. As can been seen from the plot, generally the bigger L2 consumes less leakage power than smaller ones under the same delay constraint. This agrees with the trends presented in [8] that focused only on the V_{th} assignment to optimize L2 cache leakage power. This may seem counterintuitive as larger L2 cache incurs more leakage. But since the L2 miss penalty, which is the same as main memory access time, is enormous compared to the L1 miss penalty, so any slight decrease in miss rate in L2 will result in significant improvement in the overall AMAT. A larger L2 cache results in a smaller miss rate and faster AMAT, therefore V_{th} and T_{ox} of L2 can be set more conservatively for a larger L2 than for a smaller one. Nevertheless, having the largest available L2 does not always yield the best leakage as seen in Figure 4(a). For example, for an AMAT of 1555ps, an L2 cache of 512KB outperforms one of 1024KB. This is because we reach a point where the leakage of a very large L2 outweighs the benefit of the improvement in the L2 miss rate.

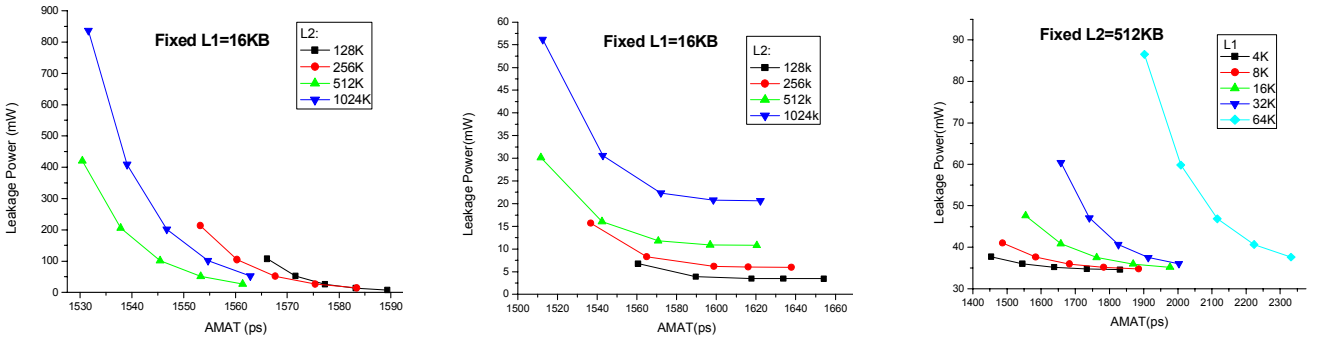
In the second part of our analysis, we assign a V_{th}/T_{ox} pair to the core array cells in an L2 cache and another pair to its peripheral circuitry. In this scenario, there are two ways to improve AMAT. One is through reducing the L2 miss rate by employing a large L2 cache as was done in the first part of our analysis; the other is by setting the V_{th}/T_{ox} assignments more aggressively in the peripheral circuitry. We found that the latter approach works better in the cases we have investigated. Figure 4(b) shows the results from this study and the smallest L2 cache consumes the least amount of leakage. Looking at the optimized V_{th}/T_{ox} assignments for L2, we see that V_{th} and T_{ox} in the core cell arrays are always set much more conservatively than those in the peripheral circuitry. This allows the leakiest component in the L2, which is the core cell array, to take on high values for V_{th} and T_{ox}, thus saving leakage. At the same time, we can still meet the target delay because V_{th} and T_{ox} in the peripheral circuitry can be set sufficiently low.

L1 Cache Leakage Power Optimization. Local L1 cache miss rates are already very low and they do not vary much amongst the L1 caches ranging from 4K to 64K as illustrated in [8]. Hence given a fixed L2, the key to minimizing total leakage power is to reduce the leakage power consumed by L1. A smaller L1 will consume less leakage and at the same time a smaller L1 will be faster. Therefore, a small L1 will probably be the optimal solution. Figure 4(c) supports this view.

Entire Processor Memory System Energy Optimization. In this experiment, we set out to find the minimum energy for a system comprising of L1, L2 and main memory. L1 is chosen to be 16KB and L2 is 512KB. We assume the main memory latency is 20ns [8], and dynamic energy dissipation per access is 3.57nJ [13] We study both the energy arising from leakage plus dynamic energy using the memory system access statistics used in the previous analysis. The core cell arrays of both L1 and L2 are to have the same V_{th}/T_{ox} assignments, and the peripherals of the L1 and L2 to have a separate pair of V_{th}/T_{ox} assignments. In Figure 5, each data point is annotated with the leakage energy and dynamic energy. In addition, V_{th}/T_{ox} assignments after optimization are also given, with the first pair denoting the V_{th}/T_{ox} assignments to the memory cell arrays of both L1 and L2, and the second pair representing V_{th}/T_{ox} assignments to the peripheral circuitry of the L1 and L2 caches. We note that T_{ox} in the core cell array is always set to the highest value, and V_{th}/T_{ox} in the peripheral circuitry are varied to minimize total energy under a given AMAT constraint. In the second part, we determine the optimal number of V_{th} and T_{ox} values needed to achieve the optimal solution. Figure 6 shows that the best scheme has 2 T_{ox}'s and 3 V_{th}'s. However, the difference between a system with dual T_{ox}, dual V_{th} and that with dual T_{ox}, triple V_{th} is very small. So in general a process with dual T_{ox} and dual V_{th} is sufficient to achieve near optimal total energy. It is also worth noting that a single T_{ox} and dual V_{th} process outperforms that with a single V_{th} and dual T_{ox} for the same reason that we discovered in Section 3: V_{th} is generally a more effective design knob than T_{ox}.

5. Temperature Effect

The studies above were done for an ambient temperature of 75°C. In this section we study the effect of different ambient temperatures on the optimization results. We pick another temperature, 25°C, and re-optimize to minimize total energy. Then we use the optimized V_{th}/T_{ox} assignments obtained at 25°C and apply them at 75°C, and vice versa. We generate total energy vs. AMAT curves for the four different scenarios, as shown in Figure 7. This figure shows that the optimization



(a) **(b)** **(c)**

Figure 4. Leakage Optimization with Various Assignment Schemes of V_{th} and T_{ox} in L2/L1 given fixed L1/L2, respectively

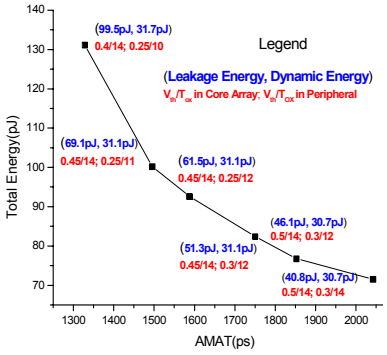


Figure 5. Total Energy Joint Optimization

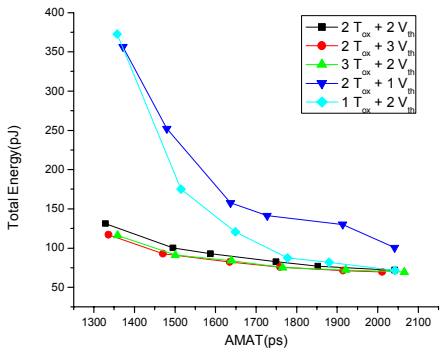


Figure 6. (T_{ox}, V_{th}) Tuple Problem

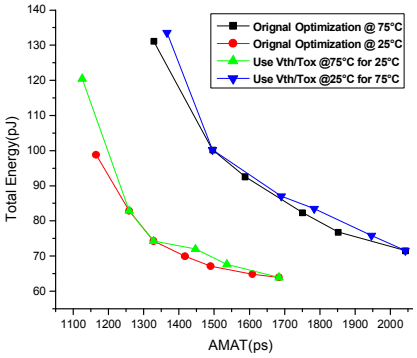


Figure 7. Temperature Sensitivity Study

results are not very sensitive to temperature. This is because after optimization, the total leakage is dominated by gate leakage. Unlike subthreshold leakage, gate leakage is only weakly dependent on the temperature. The reason that gate leakage dominates over subthreshold leakage is primarily a result of the range we have imposed on one of the two decision variables, namely, T_{ox}. From a pure optimization point of view, if T_{ox} were allowed to vary over a wider range of values, say from 10Å to 18Å, in principle the same optimal solution could be achieved where subthreshold leakage could very well dominate the total leakage. However given the targeted process technology is 65nm, we believe our choice for the range of T_{ox}, from 10Å to 14Å, is reasonable.

6. Conclusion and Future Work

In this paper, we studied the power performance trade-offs for single and two-level caches when multiple V_{th}'s and T_{ox}'s are available. For single-level cache leakage optimization, we found that it is sufficient to assign a pair of V_{th}/T_{ox} to the memory cell arrays, and another to the peripheral components. For two-level cache leakage optimization, we investigated two different scenarios. First, if L1 size is fixed and we only allow one pair of V_{th}/T_{ox} to L2, it was found that in general a bigger L2 is better, although there exists a point beyond which this trend breaks down. Second, if we allow the memory cells and the peripherals to have their own V_{th}'s and T_{ox}'s, we found that a two-level cache system with smaller L2's will yield less total leakage. However, when L2 size is fixed, smaller L1 is generally better suited for minimizing leakage. We then determined the optimal number of V_{th}'s and T_{ox}'s needed and found that a system with two V_{th}'s and two T_{ox}'s is sufficient. We also discovered that V_{th} is generally a better design knob than T_{ox} for leakage optimization. Finally we show the optimization results are temperature-insensitive because the post-optimization total leakage is dominated by the gate leakage.

In the future, the framework we have developed could be used to investigate new memory architectures and underlying device topologies like MRAM (magnetic), FeRAM (ferroelectric), etc.

References

[1] International Technology Roadmap for Semiconductors, 2003.

[2] K. Nii, *et al.*, "A Low Power SRAM Using Auto-Backgate-Controlled MTCMOS," *Proc. ISLPED*, pp 293~298, 1998.

[3] M. Powell, *et al.*, "Gated-VDD: A Circuit Technique to Reduce Leakage in Deep-Submicron Cache Memories," *Proc. ISLPED*, pp. 90~95, 2000.

[4] F. Hamzaoglu, *et al.*, "Analysis of Dual-VT SRAM Cells with Full-Swing Single-Ended Bit Line Sensing for On-Chip Cache," *IEEE Transaction on VLSI Systems*, Vol 10, pp 91~95, Apr. 2002.

[5] N. Azizi, *et al.*, "Low-Leakage Asymmetric-Cell SRAM," *Proc. ISLPED*, pp. 48~51, 2002.

[6] A. Agarwal, *et al.*, "A Single-Vt Low-Leakage Gated-Ground Cache for Deep Submicron," *IEEE JSCC*, Vol 38, pp. 319~328, 2003.

[7] C. H. Kim, *et al.*, "A Forward Body-Biased Low-Leakage SRAM Cache: Device and Architecture Considerations," *Proc. ISLPED*, pp. 6~9, 2003.

[8] N. Kim, *et al.,* "Leakage Power Optimization Techniques for Ultra Deep Sub-Micron Multi-Level Caches," *Proc. ICCAD,* pp. 627~632, 2003.

[9] Berkeley Predictive Technology Model, 2002.

[10] A. Sultania, *et al.,* "Tradeoffs between Gate Oxide Leakage and Delay for Dual T_{ox} Circuits," *Proc. DAC,* pp. 761~766, 2004.

[11] D. Bertsekas, "Nonlinear Programming," Athena Scientific, 1995.

[12] J. Hennessy and D. Patterson, "Computer Architecture A Quantitative Approach," Morgan Kaufmann Publisher, 2nd edition, 1996.

[13] V. Delaluz, *et al.*, "Compiler-Directed Array Interleaving for Reducing Energy in Multi-Bank Memories," *Proc. ASPDAC,* pp. 288~293, 2002.