Static Energy Reduction Techniques for Microprocessor Caches

Heather Hanson, *Student Member, IEEE,* M. S. Hrishikesh, *Student Member, IEEE,* Vikas Agarwal,
Stephen W. Keckler, *Member, IEEE,* and Doug Burger, *Member, IEEE*

***Abstract*— Microprocessor performance has been improved by increasing the capacity of on-chip caches. However, the performance gain comes at the price of static energy consumption due to subthreshold leakage current in cache memory arrays. This paper compares three techniques for reducing static energy consumption in on-chip level-1 and level-2 caches. One technique employs low-leakage transistors in the memory cell. Another technique, power supply switching, can be used to turn off memory cells and discard their contents. A third alternative is dynamic threshold modulation, which places memory cells in a standby state that preserves cell contents. In our experiments, we explore the energy and performance trade-offs of these techniques. We also investigate the sensitivity of microprocessor performance and energy consumption to additional cache latency caused by leakage-reduction techniques.**

***Index Terms*— static energy, leakage current, dual-V$_t$, gated-V$_{dd}$, MTCMOS, low-power design, power-consumption-model**

I. Introduction

CONTINUED improvements in integrated circuit fabrication technology have enabled the number of transistors in microprocessors to more than double each generation. A vast majority of transistors in modern microprocessors are used for on-chip storage, including level-1 and level-2 caches, and meta-state such as renaming registers, numerous predictor structures, and trace caches. As leakage current increases with each process technology generation, the energy consumption of memory structures will increase dramatically. In this paper, we explore the energy/performance trade-offs of three leakage-reduction techniques for on-chip level-1 and level-2 caches.

One method, *dual-V_t*, employs slower transistors with a higher threshold voltage, and hence lower leakage, in SRAM arrays. Transistors in the remainder of the cache circuit have a lower threshold voltage for faster switching speed. This dual-V$_t$ method decreases subthreshold leakage currents but increases the cell access time compared with an SRAM composed of fast, leaky transistors [1],[2]. Another method dynamically adjusts the effective size of the array by employing a circuit technique dubbed *gated-V_{dd}*. In this scheme, a low-leakage transistor is used to selectively shut off the power supply to a subset of SRAM cells [3]. Thus, the capacity of the array adjusts dynamically as the amount of active information in the cache changes throughout the duration of the program.

A third technique, *MTCMOS*, dynamically changes the threshold voltage by modulating the backgate bias voltage [4],[5]. With this technique, memory cells can be placed into a low-leakage “sleep” mode yet still retain their state. Cells in the active mode are accessed at full speed while accesses to cells in the sleep mode must wait until the cell has been awakened by adjusting the bias voltage. While the MTCMOS technique has been implemented for an entire SRAM [5], we examine this idea using fine-grain control of each cache line.

The fundamental circuits for leakage reduction have been introduced by other researchers; our contributions in this paper are to examine the energy/performance tradeoffs of these techniques applied to the memory hierarchy of a modern microprocessor. This paper is an extension of our prior work in [6] and is organized as follows. Section II introduces leakage current and its effects on cache energy. Section III describes the three methods for reducing leakage current in memory cells; Section IV explains our experimental methodology. Results of the experiments and a comparison of these techniques are presented in Section V. Section VI highlights relevant related work, and is followed by concluding remarks in Section VII.

II. Leakage Current

Power consumption in a digital integrated circuit is governed by the equation:

$$P = \alpha C V^2 f + I_{off} V \qquad (1)$$

where α is the average switching activity factor of the transistors, C is capacitance, V is the power supply voltage, f is the clock frequency, and I_{off} is the leakage current. The first term of the equation is dynamic power and the second term is static power. Smaller feature sizes in each generation of silicon process technologies have been accompanied by reduced power supply voltages that have helped mitigate the impact of increased transistor counts and higher clock frequencies on dynamic power. However, as the power supply voltage decreases, threshold voltages of the transistors must also decrease to achieve fast switching speeds and sufficient

S. Keckler and D. Burger direct the Computer Architecture and Technology (CART) Laboratory of the Computer Science Department, University of Texas, Austin, TX, USA.

H. Hanson, M. S. Hrishikesh, and V. Agarwal are students in the Electrical and Computer Engineering Department, University of Texas, Austin, TX, USA.

email: cart@cs.utexas.edu

This work was supported by Intel and IBM in the form of equipment grants and fellowships for Heather Hanson and Vikas Agarwal, by the NSF CADRE program, grant no. EIA-9975286, and by a grant from the Intel Research Council.

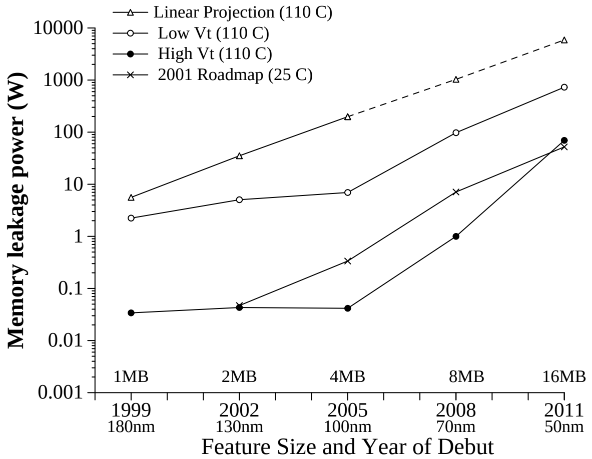


Fig. 1. Projected Leakage Power of Level-2 Caches Through Technology Generations.

noise margins. Subthreshold leakage current I_{off} is dependent on temperature T and transistor threshold voltage V_t, illustrated by the following relation:

$$I_{off} \propto e^{(\frac{-V_t}{T})} \tag{2}$$

Thus, lower threshold voltages lead to increased subthreshold leakage current and increased static power [7]. Most previous efforts at power reduction have focused on dynamic power sources because static power due to leakage current has been a small fraction of the total power dissipated by a chip. However, as transistor threshold voltages are reduced, subthreshold leakage current increases dramatically. Figure 1 shows estimated static power consumption due to leakage current in large secondary caches through five technology generations. In this chart, cache capacities are scaled from 1MB to 16MB, reflecting high-performance microprocessor cache sizes projected by [8]. Four leakage-current scaling models are charted: a linear projection from [9] for 180nm through 100nm that is extended to the 50nm node, two experimental leakage models based on our SPICE models for high V_t (low leakage) and low V_t (high performance) devices, and a projection based on the static power estimates for high-performance transistors from [10]. In these models, supply voltages are scaled from 1.6V down to 0.6V across the technology generations. The high-performance roadmap projection is charted for 25°C, while the other projections reflect a circuit temperature of 110°C. Note that due to the exponential dependence on temperature, leakage current from the roadmap model would be higher if it were also plotted for 110°C. While estimates of leakage current vary due to different scaling assumptions, each projection shows that if left unchecked, leakage current and static power will increase as feature sizes and threshold voltages decrease.

III. Leakage Reduction Techniques

This section describes our implementation of each leakage reduction strategy and our experimental methodology to simulate each technique applied to the level-1 instruction cache (IL1), level-1 data cache (DL1), and level-2 cache (L2). Table I summarizes the primary advantages and disadvantages of the three techniques for reducing leakage energy.

A. Static Threshold Selection: Dual-V_t

The dual-V_t technique employs transistors with higher threshold voltages in memory cells and faster, leakier transistors elsewhere within the SRAM. This technique requires no additional control circuitry and can substantially reduce the leakage current when compared to low-V_t devices. The amount of leakage current is engineered at design time, rather than controlled dynamically during operation. No data are discarded and no additional cache misses are incurred. However, high-V_t transistors have slower switching speeds and lower current drive. In our experiments, we consider an additional cycle of access time for SRAMs composed of these high-threshold devices.

B. Power Supply Switching: Gated-V$_{dd}$

The gated-V$_{dd}$ technique interposes a high-threshold transistor between the circuit and one of the power supply rails. This study uses an NFET as the control mechanism to take advantage of the greater current reduction from the stacking effect of the NFETs in the SRAM cell and bitline pass gates [3]. The left circuit in Figure 2 shows the schematic of a gated-V$_{dd}$ SRAM cell with an NFET selectively connecting the cell to the ground rail. When the `active` signal is asserted, the SRAM cell operates normally, but when `active` is deasserted, the cell is disconnected from ground and the state contained within the cell is lost. The activation transistor and the control mechanism for `active` can be shared by all cells within a cache line to minimize the extra area needed by the control transistor. We assume that this power supply gating transistor is sized so that the increase in memory array access time is negligible.

C. Dynamic Threshold Modulation: MTCMOS

Leakage current may also be reduced by dynamically raising the transistor threshold voltage, typically by modulating the back-gate bias voltage. A technique amenable to fine-grain control is Auto-Backgate-Controlled Multi-threshold-CMOS (which we will refer to as MTCMOS), as shown in the right circuit of Figure 2 [4],[5]. During normal operation, when `sleep` is deasserted, the SRAM is connected to V$_{dd}$ and ground and back-gate voltages are set to the appropriate power rails. When `sleep` is activated, the PFET wells are biased using an alternative power supply voltage, V$_{dd+}$, at a higher voltage level than the source terminals. Increasing the negative source-substrate voltage potential increases the effective threshold voltage for the PFETs. Diodes allow the voltage levels of source terminals of the NFETs to increase by two diode drop voltages while the NFET well remains at Gnd, increasing the source-substrate voltage potential and raising the effective V$_t$ for the NFETs. Thus all transistors experience higher threshold voltages and a corresponding drop in leakage current. As with gated-V_{dd}, we assume that any increase in

TABLE I

SUMMARY OF LEAKAGE REDUCTION TECHNIQUES

Technique	Benefit	Detriment
Dual-V_t	no additional circuitry	each read access is slower
Gated-V_{dd}	simple circuit	additional cache misses
MTCMOS	no additional cache misses	complex circuitry with diodes

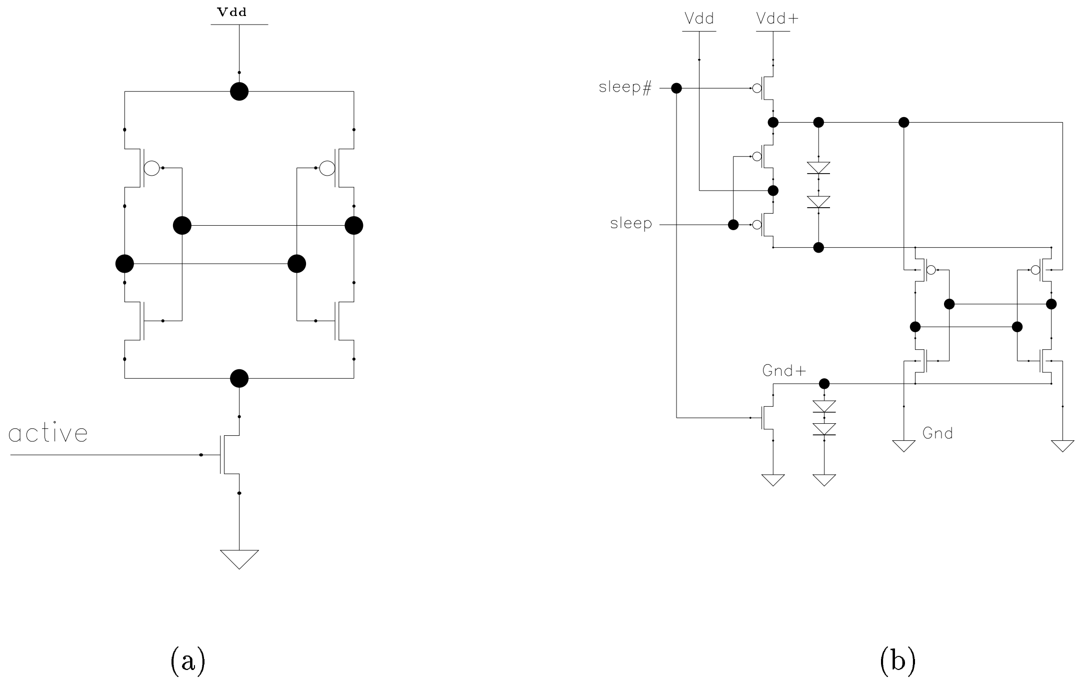


Fig. 2. Gated-V_{dd} and MTCMOS SRAM cell schematics

memory array access time is negligible while `sleep` is not asserted.

The advantage of adjusting the threshold voltage dynamically, rather than gating the power supply, is that memory cell values are preserved during sleep mode, so there are no additional cache misses caused by accessing a line in the low-power mode. This technique provides an opportunity to reduce static power consumption without incurring the cost in time and energy to retrieve data from another level of the memory hierarchy. The disadvantages of MTCMOS include an additional power supply voltage that must be distributed throughout the array, larger electric fields placed across the transistor gates during sleep mode that may adversely affect reliability, and a latency penalty to awaken a line that is in the sleep mode before the data can be accessed.

D. Decay Intervals

Energy-saving techniques such as gated-V_{dd} and MTCMOS that disable cache line rely on two properties of the data stored in caches. First, only a small fraction of the information in the cache is *live*, meaning that it will be referenced again before being replaced or over-written. In our experiments, we found that only 1–30% of a 2MB level-2 cache holds live data, depending on the application. Even in level-1 caches, less than half of the cache contains useful data across our benchmark suite. Second, most lines that will be reused are accessed within a relatively short time interval.

Cache lines containing information that is either not useful or will not be accessed for a long time can be put into an idle, low-leakage mode to save energy without a significant effect on processor performance. We determine which lines to place in an idle mode in the gated-V_{dd} and MTCMOS methods by measuring inter-access times, similar to Kaxiras *et al.* [11], [12] who proposed low-frequency counters to measure the time since last reference for every cache line. A read or write to a cache line resets its counter; when the counter reaches its maximum value after a duration named the *decay interval*, the line is deactivated.

IV. EXPERIMENTAL METHODOLOGY

To evaluate the effectiveness of the leakage-reduction techniques, we modified a version of the SimpleScalar simulator [13]. We added the capability to discard cache lines or put

TABLE II
EXPERIMENTAL PARAMETERS FOR ENERGY CALCULATIONS

	100nm Technology		Per-Bit Leakage Current (110 C)		Per-Bit Trans. Energy	Dynamic Energy Per Cache Access			
Technique	Clock Rate (GHz)	V_{dd} (Volts)	I_{max} (nA)	I_{min} (nA)	E_{switch} (fJ)	E_{IL1} (nJ)	E_{DL1} (nJ)	E_{L2} (nJ)	E_{pins} (nJ)
Baseline	2.5	0.75	1941	-	-	0.07	0.07	4.5	0.9
Dual-V_t	2.5	0.75	-	26	-	0.07	0.07	4.5	0.9
Gated-V_{DD}	2.5	0.75	1939	9.7	0.35	0.07	0.07	4.5	0.9
MTCMOS	2.5	0.75	1941	12	50	0.07	0.07	4.5	0.9

them to sleep after a specified decay interval had passed since the last access to the cache line.

A. Simulation Methodology

Our benchmark suite for this study consists of five SPEC2000 benchmarks that represent a range of cache usage characteristics: *gcc*, *eon*, *equake*, *mcf*, and *vpr*. The benchmarks are compiled for the Alpha instruction set. The simulation execution core is configured as a 4-wide superscalar pipeline organization roughly comparable to the Compaq Alpha 21264. The memory hierarchy consists of a 64KB, 2-way set associative level-1 instruction cache with a single-cycle hit latency, a 64KB, 2-way set associative level-1 data cache with a 3-cycle hit latency, and a unified 2MB 4-way level-2 cache with a 12-cycle hit latency. The level-1 caches have cache line sizes of 64 bytes, and the level-2 cache line size is 128 bytes. In the gated-V_{dd} and MTCMOS techniques, data bits may be placed into an idle mode and cache tags are kept in the active state to provide fast lookup times.

In each experiment, we applied a leakage reduction technique to one cache and simulated benchmark execution with our modified SimpleScalar simulator. The simulations executed 1 billion instructions after fast-forwarding through the first 500 million instructions. We measured instructions per cycle (IPC), active and inactive durations for each cache line, the number of hits and misses in each level of the hierarchy, and the number of times any cache line is enabled or disabled. For gated-V_{dd}, disabling a cache line is equivalent to switching off the power supply, while for MTCMOS, it is equivalent to placing the cache line into sleep mode. We calculated the total energy by multiplying these measured quantities by the relevant static and dynamic energy parameters described below and summing the energy consumed by individual components of the system.

B. Energy Parameters

Leakage currents and energy values were measured with the HSPICE circuit simulator. Physical parameters used in this study originally targeted a 70nm process and corresponding clock rate of 16 fanout-of-four inverter delays [14]. With information now available in [10], the process parameters used in this study are more closely aligned with 100nm technology parameters. We retained the original data, and have renamed the technology generation to reflect industrial trends.

Table II summarizes the experimental parameters used in this study. In this table, I_{max} and I_{min} are leakage currents when SRAM cells are active and disabled, respectively. The SRAM cell circuit and Level 3 HSPICE transistor models are adapted from the cache tool CACTI 2.0 [15], with parameters scaled for the 100nm technology generation. In each experiment, V_t = 0.4V for high threshold voltage transistors and V_t = 0.2V for low threshold voltage transistors. E_{switch} approximates the energy required to switch the cell between active and inactive modes. E_{IL1}, E_{DL1}, and E_{L2} represent the energy to read data from the level-1 instruction, level-1 data, and level-2 caches, respectively, based on a modified version of CACTI 2.0 [15] and our projected process parameters. We estimate the energy to drive the I/O pins with a simple model based on the following equation [16]:

$$E_{pin} = 1.3 C_{pin} V_{pin}^2. \tag{3}$$

We set C_{pin} = 10pF, according to the multi-chip module estimates in [16] and use a value for the pin supply voltage of V_{pin}=1.5V [17]. With a 32-bit address bus, this results in an energy cost of 0.9nJ per off-chip access. We account only for the pin energy that is expended in driving the address to the pins of the CPU, and not energy expended to receive data.

The total dynamic energy is calculated as the number of cache accesses multiplied by the appropriate energy per access parameter, plus the number of transitions into and out of idle mode multiplied by the energy per transition (for MTCMOS and gated-V_{dd} techniques). To compute the dynamic energy expended in cache accesses, we make the following approximations:

- Level-1 cache miss energy is equal to two cache hit accesses, one to detect the miss and one to load new data.
- Level-2 cache miss energy is equal to two cache hit accesses plus the energy to drive an address to 32 address pins for off-chip memory.
- Power consumed outside the CPU chip is not included in this study.

Static energy is computed as the product of static power per cycle and the number of cycles of program execution. In this paper, we focus only on the leakage in the cache memory arrays; this approximation neglects the leakage current due to the small fraction of transistors in the peripheral circuitry. The total energy is the sum of dynamic and static energy calculations.

Energy consumption and performance of the leakage-reduction techniques are compared to a baseline case to evaluate the experimental techniques' effectiveness in static energy reduction and performance. Implementation details specific to this baseline and the experimental techniques are outlined below.

TABLE III

SUMMARY OF EXPERIMENTAL RESULTS: HARMONIC MEAN ACROSS BENCHMARK SUITE

Level-1 Instruction Cache						
Technique	Decay Interval	IPC	Total Energy(J)	Dynamic Energy (J)	Leakage Energy (J)	Energy-Delay (E/IPC)
Baseline	-	1.645	4.688	4.539	0.141	2.663
Dual-V_t	-	0.680	4.525	4.520	0.005	6.181
Gated-V_{dd}	64K	1.641	4.584	4.539	0.039	2.613
MTCMOS	8K	1.644	4.580	4.539	0.035	2.607

Level-1 Data Cache						
Technique	Decay Interval	IPC	Total Energy (J)	Dynamic Energy (J)	Leakage Energy (J)	Energy-Delay (E/IPC)
Baseline	-	1.645	1.679	1.530	0.141	0.942
Dual-V_t	-	1.540	1.520	1.518	0.002	0.898
Gated-V_{dd}	64K	1.643	1.571	1.531	0.030	0.885
MTCMOS	1K	1.639	1.547	1.530	0.017	0.874

Level-2 Unified Cache						
Technique	Decay Interval	IPC	Total Energy (J)	Dynamic Energy (J)	Leakage Energy (J)	Energy-Delay (E/IPC)
Baseline	-	1.645	4.540	0.004	4.513	2.424
Dual-V_t	-	1.625	0.084	0.004	0.061	0.042
Gated-V_{dd}	64K	1.386	0.239	0.005	0.225	0.112
MTCMOS	0	1.626	0.140	0.004	0.115	0.072

Baseline: The baseline for comparison in this study is a high-performance cache without leakage current control. Each transistor in the SRAM cell has a threshold voltage of 0.2V, with a high leakage current of I_{max} at all times. The baseline case has the maximum performance and maximum energy consumption for the set of experiments.

Dual-V_t: Though the dual-V_t technique has low-leakage transistors in memory cells and high-leakage transistors elsewhere, we account for static energy only in the memory array, and thus only use the reduced-leakage current, I_{min}. The dual-V_t technique does not transition between idle and active states and thus does not incur extra cache misses or additional time to access sleeping cells.

Gated-V_{dd}: For the gated-V_{dd} technique, I_{max} is the leakage current when the memory cell is in the active state, and I_{min} is the leakage current when the memory cell is disconnected from the power supplies. The gating transistor has a high threshold voltage of 0.4V, and the other SRAM cell transistors' threshold voltages are the low-V_t value of 0.2V. The value of E_{switch} is based on the gate capacitance of the activation transistor and the wire capacitance to reach all of the cells in the cache line. Only "clean" lines that do not require a write back to the memory hierarchy are disabled; "dirty" lines that are not accessed before the decay interval expires are kept in the active state.

MTCMOS: The circuit design for the MTCMOS technique is adapted from [4]. In our example, the leakage current for MTCMOS SRAM arrays is controlled on the granularity of a cache line rather than the full cache. The transistors in our SRAM cells have a V_t of 0.2V, and the total voltage drop across the diodes is 3.2 volts. The second power supply, V$_{dd}$+, is 3.3 volts. I_{max} is the leakage current when the memory cell is awake, and I_{min} is the leakage current when the cells have transitioned into sleep mode. E_{switch} is the energy required to charge the cache line's well plus the energy consumed to discharge the source terminals of the NFETs. The time and energy to enter and exit sleep mode depend directly on the effective capacitance of the well that contains the PFETs in the SRAM cell; in this study, we vary the delay to awaken a sleeping cache line from 1 to 10 cycles to examine the sensitivity to wakeup latency.

V. RESULTS

This section presents our experimental results and compares trade-offs between performance and energy reduction for three leakage-reduction techniques. We analyze each technique's energy-saving potential and impact on performance using the combined energy-delay metric. Then, we explore the effects of additional cache access latency due to each leakage reduction technique.

A. Energy-Delay

We use a metric of the energy-delay product to balance the benefits of lower leakage with the potential penalty of reduced performance. We calculate the energy-delay product as the total energy divided by IPC, which is equivalent to the product of energy and a measure of time (cycles per instruction, with a fixed number of instructions).

To evaluate the gated-V_{dd} and MTCMOS strategies, we observed each technique's performance throughout a range of decay intervals, and chose intervals that resulted in the minimum energy-delay product. The best-case decay interval depends upon program cache access patterns and circuit parameters unique to each leakage-reduction technique [18]. In our study, the best decay interval for the gated-V_{dd} technique was found to be 64K cycles for each cache. For the MTCMOS technique, the best decay interval is 8K cycles for the level-1 instruction cache, 1K cycles for the level-1 data cache, and immediate sleep mode (zero-cycle decay interval) for the level-2 cache. Table III summarizes the experimental results, reported as the harmonic mean of IPC, energy, and energy-delay product for simulated program execution across the benchmark suite.

Figure 3 shows the total energy required for program execution for each leakage-reduction technique applied independently to one cache. The charts present data from the best decay interval in the gated-V_{dd} and MTCMOS techniques. In the figures of the left column, stacked bar charts illustrate the contribution of static and dynamic energy for each benchmark.

Note that in the level-1 caches, the majority of energy consumption is due to dynamic energy, whereas in level-2 caches, static energy dominates the total energy. Charts in the right column of Figure 3 show the energy-delay product for each benchmark and highlight the variation between techniques. Each of the three leakage-reduction methods in this study achieves lower leakage energy compared to the baseline case with high-performance SRAM cells but sacrifices performance to do so, whether by slowing cache accesses or causing delays to re-fetch data.

Dual-V_t: The dual-V_t cache is effective at reducing leakage; however, with an extra cycle of delay, the technique has a negative effect on performance for level-1 caches. The dual-V_t technique reduces the static energy consumed by the IL1 cache by 96%, at the expense of reducing the IPC by over half. The energy-delay product of the dual-V_t technique is more than twice that of the IL1 baseline case. Although the leakage current and therefore static energy is reduced, the performance penalty may be unacceptable for a dual-V_t method applied to an instruction cache or other structures that rely on fast access times. The dual-V_t DL1 cache reduces static energy by 98%, with an energy-delay product that is 4% better than the baseline case. In the level-2 dual-V_t cache experiment, static energy decreases by 98% with negligible performance degradation and the energy-delay product improves by over a factor of 50.

Gated-V_{dd}: With gated-V_{dd}, static energy savings are offset by the dynamic energy required to service additional misses to prematurely disabled cache lines. The total energy of the frequently accessed primary caches is dominated by dynamic energy of read accesses, and despite substantial static energy savings, the energy-delay product is only slightly better than the baseline case. The gated-V_{dd} technique applied to an IL1 with a 64K decay interval produces a 72% static energy savings, with a 2% improvement in energy-delay compared with the baseline. In the DL1 cache, the technique had similar results: 79% reduction in static energy, with a 6% improvement in the energy-delay product. In the level-2 cache, the penalty for additional execution time creates a noticeable drop in IPC. However, the energy savings with the gated-V_{dd} technique is 95%, for an overall effect of improving the energy-delay by a factor of 20.

MTCMOS: The MTCMOS IL1 cache with an 8K decay interval reduces static energy by 75%, an improvement in energy-delay of 2%. In the DL1 cache, the MTCMOS technique and a 1K decay interval decreases static energy by 88%, while improving the energy-delay product by 7%. For the level-2 cache and an aggressive sleep policy, leakage current is dramatically reduced at the expense of a slightly lower IPC. The level-2 cache with MTCMOS circuitry and an immediate sleep mode reduces static energy by 97% and improves the energy-delay product by a factor of approximately 34.

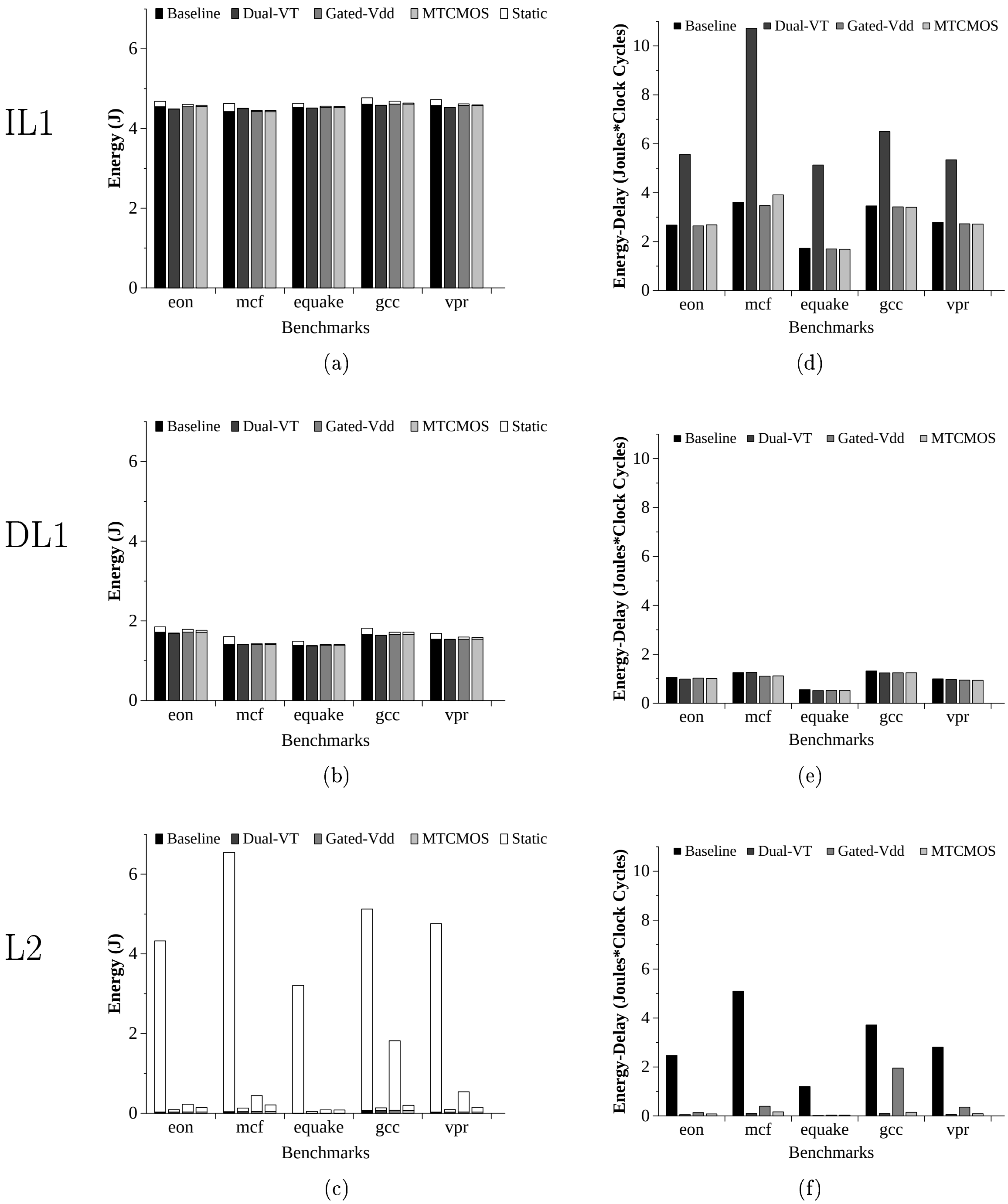
B. Sensitivity to Delay

Although leakage reduction techniques attempt to reduce static energy consumption, the performance penalties they can impose act in opposition to such savings and can reduce the techniques' effectiveness. In particular, if a program takes more time to complete with leakage reduction techniques enabled, then all remaining leaky components of the chip will leak for a longer period of time. In this section, we investigate the effects of additional latency on processor performance and static energy consumption. In dual-V$_t$ and gated-V$_{dd}$, delays are manifested in cache access time overhead, while the most interesting variable for MTCMOS is the time to wake a sleeping line.

Dual-V$_t$: Cache access time for dual-V$_t$ can increase if the speed reduction of the higher threshold devices in the cache is significant. Likewise, the high-V$_t$ cut-off transistor implemented in a gated-V$_{dd}$ strategy could also increase overall cache access time. The increase in access latency can extend the execution time of the program and degrade performance. Graphs in the left column of Figure 4 show the performance degradation for processors accessing dual-V$_t$ caches as the access latency is increased by one and two cycles. The IPC values are calculated as the harmonic mean of measured IPC results from all five benchmarks. Figure 4a shows the IPC for the level-1 instruction cache drops from 1.65 to 0.41, a substantial 74% reduction in performance as the latency increases by 2 cycles. The processor is less sensitive to additional delays in the level-1 data cache, as illustrated in Figure 4b. The mean IPC values dip from 1.64 to 1.50, an average performance reduction of 4% when the DL1 cache latency increases by two cycles. Figure 4c shows that additional latency in the level-2 cache causes the least impact on performance, with an average of 2% decrease in IPC for two extra cycles of latency.

The right column of Figure 4 indicates how longer access times translate into increased static energy for individual program execution. In addition, the harmonic mean over the full benchmark suite is reported in this discussion on sensitivity trends. In the level-1 instruction cache, the mean static energy increases by 157% for one additional cycle and 387% for two additional cycles of IL1 cache latency. Figure 4d shows how each extra cycle of latency adds to static energy consumption for each program in the benchmark suite. The short bars in Figure 4e indicate that static energy of the level-1 data cache is not as strongly affected by additional access latency. In the DL1, the static energy increases for one and two additional cycles of latency are 5% and 9%, respectively. The unified level-2 cache shows an overall 1% increase in static energy for each additional cycle of latency. Figure 4d illustrates that the static energy consumption depends upon program behavior; the increase is more pronounced in the benchmarks mcf and gcc than in equake.

MTCMOS: While MTCMOS does not suffer from additional latency to access cache lines in an awake state, its effectiveness does depend on the speed at which cache lines can be re-awakened. Additional clock cycles used to awaken sleeping cache lines can extend the program execution time, with the effect of reducing processor performance and increasing the static energy expended. The wakeup transition time is determined by the circuit configuration and physical parameters; this section explores the sensitivity of the MTCMOS technique applied to primary and secondary caches



(a) (d)

(b) (e)

(c) (f)

Fig. 3. Energy and Energy-Delay Product for L1 and L2 Caches

as the experimental wakeup penalty is varied from 1 to 10 cycles. Results are reported as the harmonic mean of IPC and the harmonic mean of the static energy for program execution of all benchmarks in the suite.

Graphs in the left column of Figure 5 show the combined effect of decay interval and wakeup latency on processor performance. In figures 5a, 5b, and 5c, the processor's performance is plotted as a function of the wakeup latency for four cache decay intervals: immediate sleep, 1K, 8K, and 64K cycles. Graphs in the right column of Figure 5 show the static energy consumption expended by the processor as a function of the wakeup latency for four cache decay intervals: immediate sleep, 1K, 8K, and 64K processor cycles. Unlike the dual-V_t scenario in which extra latency affects each cache access, MTCMOS caches incur extra latency only for accesses to sleeping cache lines.

An MTCMOS level-1 instruction cache causes the largest performance degradation in IPC when short decay intervals with long wakeup latencies are employed, as illustrated in Figure 5a. For an IL1 cache with an MTCMOS immediate sleep policy, the measured IPC drops by 93% when the wakeup penalty is ten cycles compared to a wakeup penalty of 1 cycle.

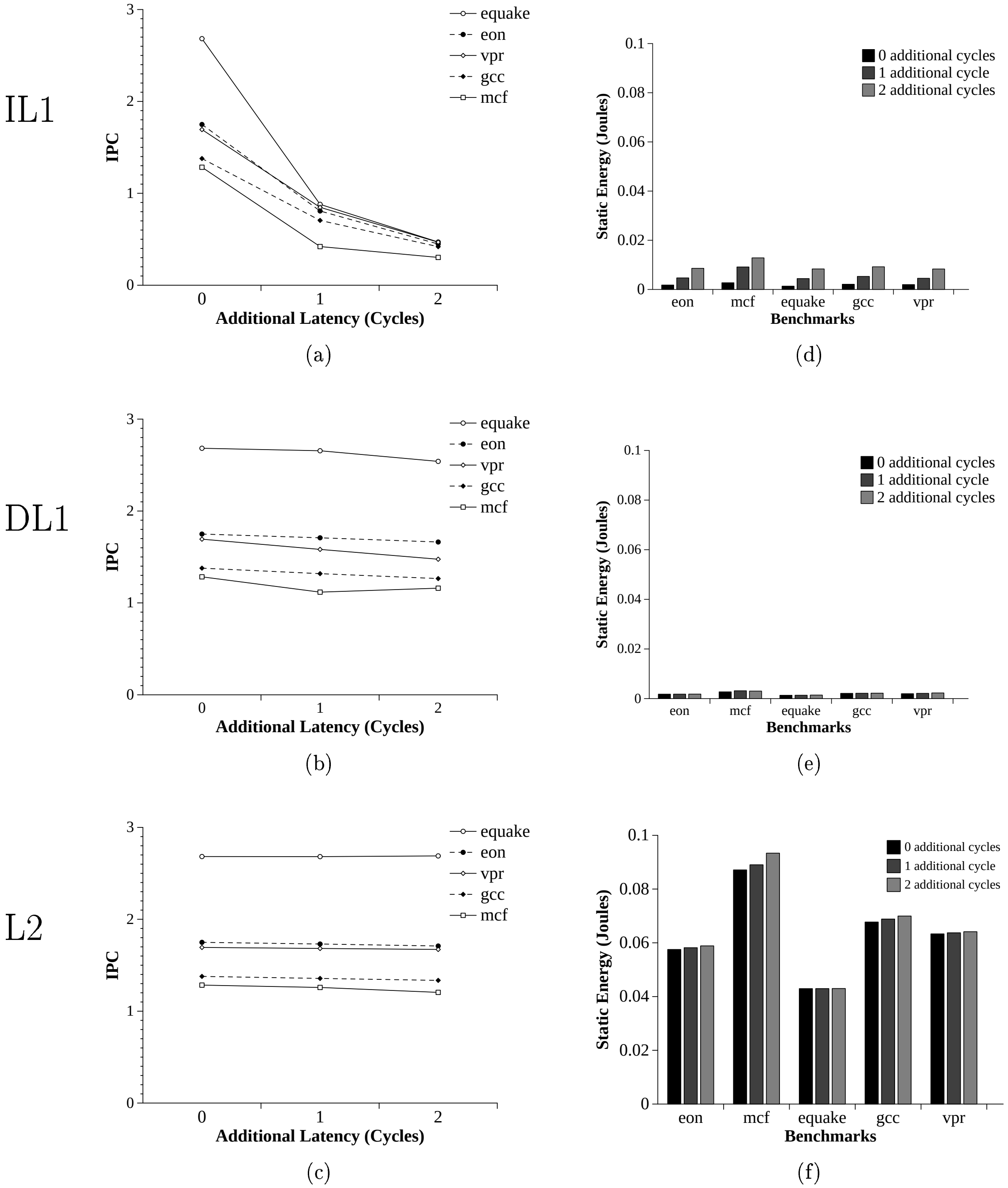
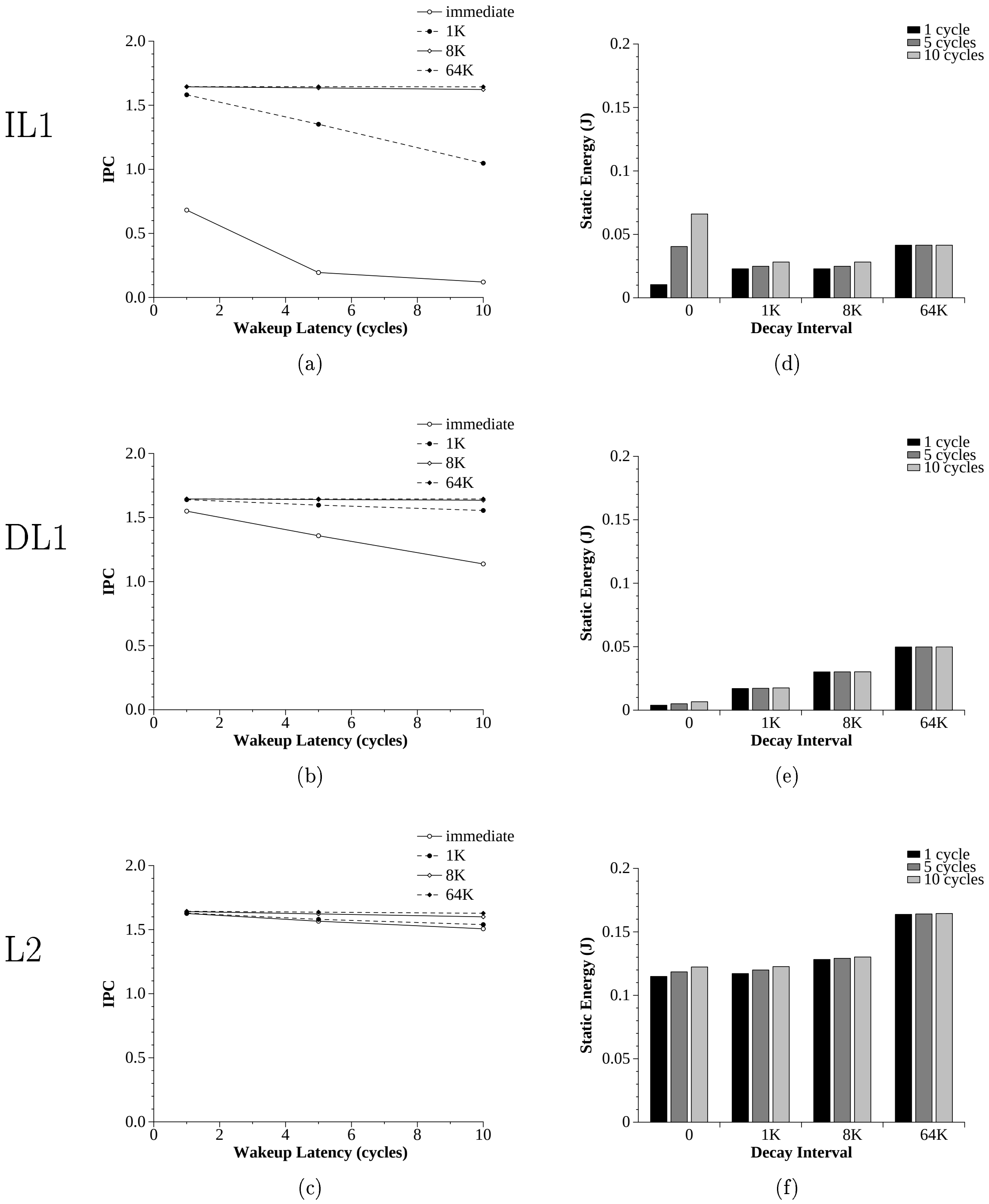


Fig. 4. IPC and Energy Sensitivity to Access Delay for L1 and L2 Dual-V_t Caches.

For a larger decay interval of 64K cycles, when most useful cache lines are kept awake, the IPC is reduced by less than 1% when the wakeup penalty is increased from 1 to 10 cycles. With a decay interval of 8K, the best-case interval in this study for MTCMOS IL1 caches, the IPC is 1.35% lower for a ten-cycle wakeup time. Figure 5d shows that an MTCMOS IL1 cache with an immediate sleep mode uses 18 times more static energy with a wakeup penalty of 10 cycles than with a 1 cycle penalty. However, since dynamic energy dominates the total energy for the primary caches, the total IL1 cache energy consumption increases by only 3%. With a decay interval of 64K, the program execution time is not noticeably affected, and the static energy is essentially unchanged.

The MTCMOS DL1 cache also causes performance degradation with short decay intervals. As Figure 5b illustrates, an MTCMOS DL1 cache with an immediate sleep policy causes an IPC drop of 31% from 1-cycle to 10-cycle wakeup penalties. The extra execution time for this case leads to an additional 3mJ of static energy, an 86% increase. Longer decay intervals, however, show only a slight decrease in performance, and the static energy shows more sensitivity to the decay interval than to extra latency, as seen in Figure 5e.

Since L2 accesses are relatively infrequent, program execution time is only mildly extended due to waiting for sleeping



(a) (d)

(b) (e)

(c) (f)

Fig. 5. IPC and Energy Sensitivity to Access Delay for L1 and L2 MTCMOS Caches.

L2 cache lines to transition to the active mode. A zero-cycle decay interval leads to the largest IPC drop of 8%. With most lines in a low-leakage mode, additional processor cycles contribute only a small amount of extra leakage current. The largest static energy increase was 7% for the immediate-sleep policy. Figure 5e shows that as the decay interval increases, the effect of additional latency decreases. Since static energy is the largest component of the total energy in the level-2 cache, the effect of increased static energy is an overall energy increase of 5% for the immediate-sleep configuration.

VI. Related Work

Leakage-reducing circuit techniques can be incorporated into architectural solutions that rely on programs' use of system resources to reduce static energy. One example employs a gated-V_{dd} circuit to selectively disable cache lines based on miss rates, dynamically resizing the instruction cache (DRI I-cache) to a size appropriate for the currently executing program. Yang *et al*. found that a 64K DRI I-cache reduced the energy-delay product by 62% with a 4% increase in execution time with SPEC95 benchmarks, compared to a standard cache

[19].

Kaxiras *et al.* have developed improvements to the gated-V_{dd} technique with an adaptive control on the gating transistor, and have shown that their technique can reduce leakage energy in level-1 caches by a factor of five [12]. Zhou *et al.* have proposed a low-leakage cache design named Adaptive Mode Control that dynamically adjusts the number of cache lines turned off by the gated-V_{dd} method throughout program execution to keep the number of extra cache misses caused by disabling cache lines proportional to the number of misses that would be incurred with a standard cache [20]. With adaptive mode control, a level-1 instruction cache with an average of 74% of the cache lines disabled and a level-1 data cache with an average of 50% disabled cache lines results in an IPC drop of less than 1.6%.

Recently, Flautner, *et al.* introduced a technique that in principle is similar to the cache-line level control we introduce for MTCMOS [21]. Instead of modulating the back-gate bias, their drowsy caches modulate the power supply voltage to the cache's memory cells to reduce the voltage, and thus the leakage current, when a cache line has not been accessed for a while. The advantages to this technique are that the circuit to control leakage is simpler and is likely to enable faster transitions into and out of the sleep mode. However, according to our estimates, MTCMOS can provide an additional order of magnitude reduction in leakage current. Thus the technique of Flautner *et al.* is better suited for latency-critical caches while MTCMOS is better suited to leakage-critical caches.

VII. Conclusion

In this paper we have explored energy and performance trade-offs associated with three techniques for reducing static energy consumption in on-chip caches: high-V$_t$ transistors in memory arrays, power supply switching, and dynamic transistor threshold modulation.

Each of the techniques is effective in reducing energy consumption in primary and secondary caches. We found that with careful selection of decay intervals, the MTCMOS and gated-V_{dd} techniques yielded better energy-delay products than the dual-V_t technique in the primary caches, due to their overall lower access time. With our assumptions, both the gated-V_{dd} and MTCMOS techniques improve the energy-delay product by 2% in the IL1 cache, and yield an improvement of 6% and 7%, respectively, in the DL1 cache compared to the experimental baseline. The dual-V_t technique improves the energy-delay product of the DL1 by 4%, and degrades energy-delay product in the IL1. For the secondary cache, the dual-V$_t$ technique has the best energy-delay characteristics, with a 50-fold improvement compared to the baseline case. The gated-V_{dd} and MTCMOS techniques were also effective at improving the energy-delay of L2 caches, with overall reductions of factors of 20 and 34, respectively.

However, additional latency and energy penalties contributed by the leakage reduction strategy [18], can extend program execution time and increase static energy consumption, especially when applied to the primary instruction cache. Increasing the dual-V_t IL1 cache access by two extra cycles results in performance degradation of 74%, and a 387% increase in static energy expenditure. For an MTCMOS IL1 with a zero-cycle decay interval, performance drops by 93% and static energy increases by a factor of 18 when the wakeup latency is ten cycles rather than one. In the level-1 data cache, the effect of additional access time was less detrimental. A dual-V_t DL1 with two additional cycles of access time reduces performance by 4% and increases static energy by 9%. An MTCMOS DL1 with a ten-cycle wakeup latency causes performance to drop by 31% with the shortest decay interval; longer decay intervals do not suffer such performance degradation. The unified level-2 cache is the least sensitive to additional delays, with a 2% dip in IPC for the dual-V_t L2 cache accompanied by a 2% increase in static energy; an MTCMOS L2 cache with the worst-case of immediate sleep policy caused 8% reduction in IPC and 7% increase in static energy consumed.

This paper has emphasized static energy reduction in cache memories while considering the effect on processor performance and total energy. The same principles may be applied to other hardware structures, as well. For example, the static energy required to maintain the state of branch predictor table entries may be balanced against the dynamic energy required to execute with fewer correct predictions. Future work will include static energy analysis of other microarchitectural features and their impact on microprocessor performance and total energy.

References

[1] T. McPherson, R. Averill, D. Balazich, K. Barkley, S. Carey, Y. Chan, Y.H. Chan, R. Crea, A. Dansky, R. Dwyer, A. Haen, D. Hoffman, A. Jatkowski, M. Mayo, D. Merrill, T. McNamara, G. Northrop, J. Rawlins, L. Sigal, T. Slegel, and D. Webber, "760 MHz G6 S/390 microprocessor exploiting multiple V_t and copper interconnects," in *International Solid-State Circuits Conference*, 2000, pp. 96–97.

[2] K. Roy, "Leakage power reduction in low-voltage CMOS designs," in *International Conference on Electronics, Circuits and Systems*, 1998, pp. 167–73.

[3] M. Powell, S.H. Yang, B Falsafi, K. Roy, and T.N. Vijaykumar, "Gated-V_{dd}: A circuit technique to reduce leakage in deep-submicron cache memories," in *International Symposium on Low Power Electronics and Design*, 2000, pp. 90–95.

[4] H. Makino, Y. Tujihashi, K. Nii, C. Morishima, Y. Hayakawa, T. Shimizu, and T. Arakawa, "An auto-backgate-controlled MT-CMOS circuit," in *Symposium on VLSI Circuits*, 1998, pp. 42–43.

[5] K. Nii, H. Makino, Y. Tujihashi, C. Morishima, Y. Hayakawa, H. Nunogami, T. Arakawa, and H. Hamano, "A low power SRAM using auto-backgate-controlled MT-CMOS," in *International Symposium on Low Power Electronics and Design*, 1998, pp. 293–298.

[6] H. Hanson, M. Hrishikesh, V. Agarwal, S. Keckler, and D. Burger, "Static energy reduction for microprocessor caches," in *International Conference on Computer Design*, 2001.

[7] J. A. Butts and G. Sohi, "A static power model for architects," in *Proceedings of 33rd Annual International Symposium on Microarchitecture*, December 2000, pp. 191–201.

[8] "International technology roadmap for semiconductors, 2000 update, overall technology roadmap characteristics," 2000, http://public.itrs.net/Files/2000UpdateFinal/ORTC2000final.pdf.

[9] S. Borkar, "Design challenges of technology scaling," *IEEE Micro*, vol. 19, no. 4, pp. 23–29, July-August 1999.

[10] International Technology Roadmap for Semiconductors, "International technology roadmap for semiconductors, 2001 edition," 2001, http://public.itrs.net/Files/2001ITRS/Home.htm.

[11] S. Kaxiras, Z. Hu, G. Narlikar, and R. McLellan, "Cache-line decay: A mechanism to reduce cache leakage power," in *Workshop on Power Aware Computer Systems*, 2000.

[12] S. Kaxiras, Z. Hu, and M. Martonosi, "Cache-line decay: Exploiting generational behavior to reduce leakage power," in *The 28th Annual International Symposium on Computer Architecture*, July 2001, pp. 240–251.

[13] D. Burger and T. Austin, "The simplescalar tool set version 2.0," Tech. Rep. 1342, Computer Sciences Department, University of Wisconsin, June 1997.

[14] Mark Horowitz, Ron Ho, and Ken Mai, "The future of wires," in *Semiconductor Research Corporation Workshop on Interconnects for Systems on a Chip*, May 1999.

[15] G. Reinman and N. Jouppi, "An integrated cache timing and power model," 1999, Unpublished document.

[16] D. Liu and C. Svensson, "Power consumption estimation in CMOS VLSI chips," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 6, pp. 663–660, June 1994.

[17] "Pentium III processor for the sc242 at 450MHz to 1.13GHz," Intel Corporation, June 2000, Order Number 244452-008.

[18] Heather Hanson, "Comparison of leakage energy reduction techniques," Tech. Rep. TR-01-18, Computer Sciences Department, University of Texas at Austin, June 2001.

[19] S.H. Yang, M. Powell, B. Falsafi, K. Roy, and T.N. Vijaykumar, "An integrated circuit/architecture approach to reducing leakage in deep-submicron high-performance caches," in *International Symposium on High-Performance Computer Architecture*, 2001, pp. 147–157.

[20] H. Zhou, M. Toburen, E. Rotenberg, and T. Conte, "Adaptive mode-control: A static-power-efficient cache design," in *International Conference on Parallel Architectures and Compilation Techniques*, 2001, pp. 61–72.

[21] K. Flautner, N.S. Kim, S. Martin, D. Blaauw, and T. Mudge, "Drowsy caches: Simple techniques for reducing leakage power," in *The 29th Annual International Symposium on Computer Architecture*, May 2002, pp. 148–157.

Heather Hanson received the B.S. degree in Electrical and Computer Engineering and the B.A. degree in Liberal Arts in 1994, and the M.S in Electrical and Computer Engineering in 2001 from The University of Texas at Austin. She has worked at Logical Silicon Solutions as a circuit designer and Intel Corporation as a logic designer. She is currently a doctoral candidate at the University of Texas at Austin, researching power and energy-efficient microprocessors.

M. S. Hrishikesh received his B.E. degree in Electrical Engineering from the University of Madras in 1997 and his M.S. degree from the University of Texas at Austin in 1999. He is currently a doctoral candidate at the University of Texas at Austin. His research focuses on the scalability of processor to very small feature sizes for high performance computing. He is currently investigating clustering mechanisms for very wide issue processors.

Vikas Agarwal received the B.Tech. degree in Electrical Engineering from the Indian Institute of Technology, Bombay in 1996 and the M.S. degree in Electrical and Computer Engineering from the University of Texas at Austin in 1998. He is currently a doctoral candidate at the University of Texas at Austin. His research focus is modeling the effect of semiconductor technology scaling on microprocessor microarchitectural structures. He is currently investigating reliability issues of large on-chip cache structures.

Stephen W. Keckler is an Assistant Professor of both Computer Sciences and Electrical and Computer Engineering at the University of Texas at Austin, as well as an Alfred P. Sloan Research Fellow. His research interests include computer architecture, parallel and embedded processors, VLSI design, adaptive computing, and the relationship between technology and computer system development. As co-director the Computer Architecture and Technology (CART) Laboratory, Dr. Keckler's research is currently supported by a National Science Foundation CAREER award, an IBM University Partnership award, and grants from the National Science Foundation, Intel, and IBM, and DARPA. He received a BS in electrical engineering from Stanford University and an MS and a PhD in computer science from the Massachusetts Institute of Technology. Dr. Keckler is a member of the IEEE, Sigma Xi, and Phi Beta Kappa.

Doug Burger has been an Assistant Professor of Computer Sciences and Electrical and Computer Engineering at the University of Texas at Austin since 1999. He received his Ph.D. in Computer Sciences from the University of Wisconsin-Madison, and his B.S. from Yale University in 1991. His main research area is computer architecture, and his interest span compilers, operating systems, and emerging technologies. He is co-leader of the TRIPS project at UT-Austin, which is building the microprocessors for a new levels performance and flexibility across many application classes. He is a 2000 NSF CAREER Award recipient, an IBM Center for Advanced Studies Fellow, and a Sloan Foundation Research Fellow.