System Level Leakage Reduction Considering the Interdependence of Temperature and Leakage *

Lei He, Weiping Liao and Mircea R. Stan†

EE Department, University of California, Los Angeles 90095
† ECE Department, University of Virginia, Charlottesville 22904
{lhe, wliao}@ee.ucla.edu, mircea@virginia.edu

ABSTRACT

The high leakage devices in nanometer technologies as well as the low activity rates in system-on-a-chip (SOC) contribute to the growing significance of leakage power at the system level. We first present system-level leakage-power modeling and characteristics and discuss ways to reduce leakage for caches. Considering the interdependence between leakage power and temperature, we then discuss thermal runaway and dynamic power and thermal management (DPTM) to reduce power and prevent thermal violations. We show that a thermal-independent leakage model may hide actual failures of DPTM. Finally, we present voltage scaling considering DPTM for different packaging options. We show that the optimal V_{dd} for the best throughput may be smaller than the largest V_{dd} allowed by the given packaging platform, and that advanced cooling techniques can improve throughput significantly.

Categories and Subject Descriptors: B.7.1 [Integrated Circuits]: Types and Design Styles
General Terms: Design
Keywords: Microarchitecture, Leakage power, Temperature

1. INTRODUCTION

The leakage current in nanometer devices has increased drastically due to reduction in threshold voltage, channel length and gate oxide thickness [1]. In addition, an increasing number of modules in a highly integrated system are idle at any given time. The high leakage devices and low activity rates both contribute to the growing significance of leakage power at the system level. The Intel Pentium IV processors running at 3GHz already have an almost equal amount of leakage and dynamic power [3]. Furthermore, since leakage power has an exponential dependence on temperature [2], power and thermal modeling is hardly accurate without considering the interdependence between leakage and temperature.

*This work is partially supported by NSF contracts CCR-0306682 and CCR-0306682, SRC grant 1116, UC MICRO grants by Fujitsu, Intel and Mindspeed, an IBM Faculty Partner Award, and a University of Virginia FEST award. Address comments to lhe@ee.ucla.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
DAC'04, June 7–11, 2004, San Diego, California, USA.
Copyright 2004 ACM 1-58113-828-8/04/0006 ...$5.00.

An important contribution of this paper is the exploration of various microarchitecture-level leakage power and thermal models as well as coupled power and thermal simulation and management considering the interdependence between leakage power and temperature. This "closes the loop" in temperature-aware simulation as shown in Figure 1. We also discuss related issues such as microarchitecture-level leakage power characteristics, runtime leakage reduction on caches, thermal runaway and dynamic power and thermal management (DPTM). Furthermore, we present a case study on system assessment by DPTM with appropriate delay model considering voltage and temperature scaling.

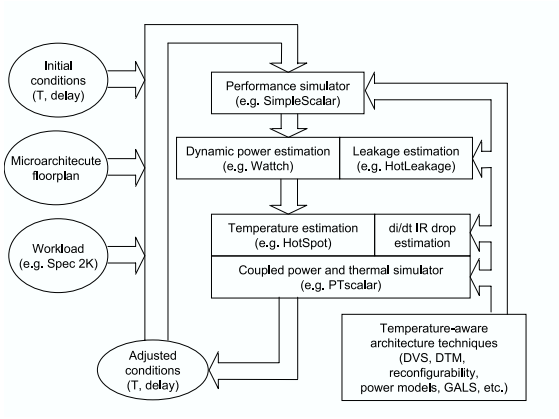

Figure 1: Closing the loop in temperature-aware design: from initial specifications, power and performance models, to resulting temperature profiles in space and time, to updated performance and power models.

The rest of the paper is organized as follows. Section 2 discusses microarchitecture-level leakage power modeling and reduction. Section 3 discusses microarchitecture-level coupled power and thermal modeling. Section 4 presents a case study on system assessment with DPTM. Finally Section 5 concludes the paper. The detailed references and settings for all experiments are included in [4].

2. MICROARCHITECTURE-LEVEL LEAKAGE MODELING AND REDUCTION

Three power states can be defined at the system level: (i) *active mode*, where a circuit performs an operation and dissipates both dynamic power (P_d) and leakage power (P_s). The active power

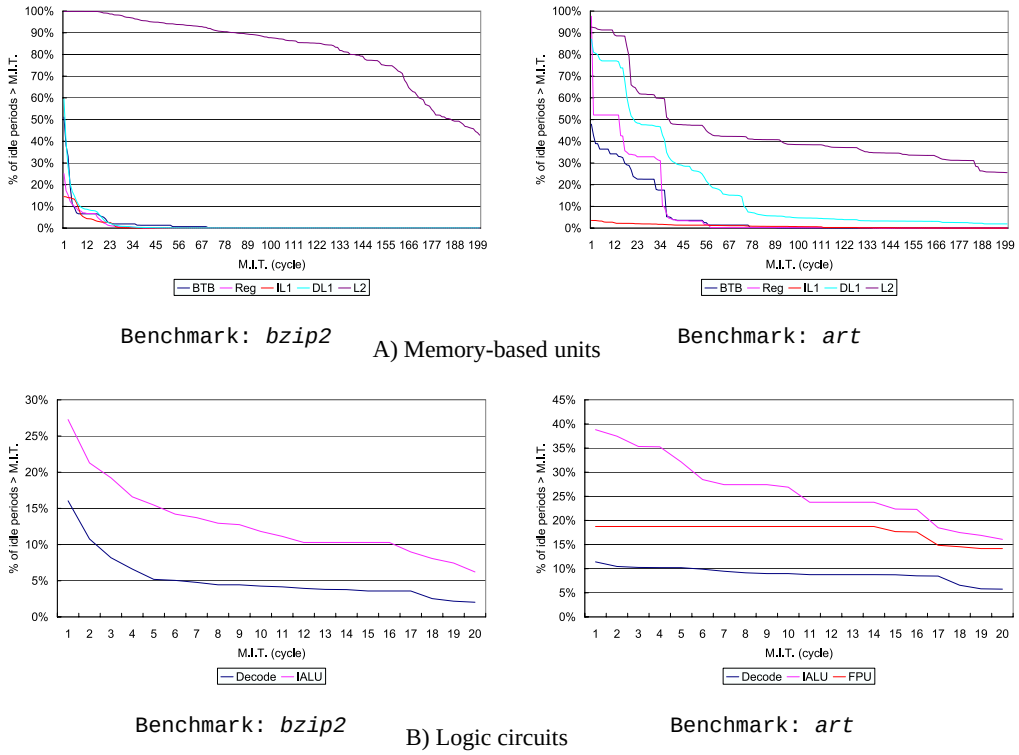

Figure 2: The percentage of idle periods larger than the Minimum Idle Time (M.I.T.) for: A) memory components, B) logic circuits.

P_a is the sum of P_d and P_s. (ii) *standby mode*, where a circuit is idle but ready to execute an operation and dissipates only leakage power(P_s). (iii) *inactive mode*, where a circuit is deactivated by power gating [5] or other leakage reduction techniques, and dissipates a reduced leakage power defined as inactive power ($P_i < P_s$).

2.1 Leakage Modeling

Early microarchitecture-level leakage power modeling studies [6] propose a simple leakage power model:

$$P_s = V_{dd} \cdot N_{FET} \cdot k_{design} \cdot I_{leakage} \quad (1)$$

where V_{dd} is the supply voltage, N_{FET} is the number of transistors, k_{design} is the design dependent parameter for different circuits, geometries and stack effect; and $I_{leakage}$ is a technology dependent parameter.

In [8] the authors study microarchitecture-level leakage power modeling with consideration of two types of circuits: (1) memory-based units such as register files and caches; and (2) logic circuits such as functional units. Two different leakage power reduction techniques: VRC [9] and MTCMOS [5] are applied to memory-based units and logic circuits, respectively. Empirical formulas are developed for both types of circuits with coefficients obtained from circuit level SPICE simulation. In [10] the authors present the following leakage power model similar to (1):

$$P_s = N_{gate} * I_{avg} * V_{dd} \quad (2)$$

where N_{gate} is the total number of gates in the circuit and a method to calculate I_{avg} is developed considering the stacking effect and input vector.

2.2 Leakage Power Characteristics

Due to the extra energy overhead during the standby-to-inactive and inactive-to-standby transitions for leakage power reduction techniques, circuits have to remain inactive for a certain amount of time, called the minimum idle time (M.I.T.), in order to get power savings [8]. For microarchitecture components, the cycle time between two consecutive accesses is defined as the *idle period* of the component and leakage power reduction techniques for components are beneficial only when the idle period is longer than the M.I.T.. Fig. 2A and 2B present the percentage of idle periods longer than that minimum idle time for different M.I.T. values. [1] The microarchitecture is similar to Intel Itanium 2 processors. Ideal power gating implemented by trace analysis [8] can be used to decide the upper bound of leakage power reduction, where a component can be power gated for any idle period longer than the M.I.T. and always wakes up in time to avoid performance loss. VRC and MTCMOS are applied to memory-based units and logic circuits with M.I.T. 160 cycles and 3 cycles for 100nm technology, respectively. With ideal power gating, Fig. 2A indicates that among memory-based units, the L2 cache has about 20% to 50% chance of leakage power reduction. Fig. 2B shows that for logic circuits, there is a 20% to 40% opportunity for leakage power reduction. Detail studies show that the upper bound of leakage reduction can be up to 30% of total power in high-performance VLIW processors [8].

2.3 Runtime Leakage Reduction for Caches

Due to the large SRAM array structures, caches dissipate large amount of leakage power and become the focus of leakage energy reduction research in the literature. Different techniques, such as DRI Caches [11], selective cache ways [12], and AMC cache [14], have been proposed to reduced L1 cache leakage power by dynam-

[1] Note that FPU is omitted for the integer benchmark *bzip2*.

ically turning off partial cache array structure. Feedback control is further applied [15] to AMC cache for better adaptive control with bounded performance loss. Similar to VRC cache [8], drowsy cache [13] addresses the data retention problem by dynamically reducing supply voltage for leakage reduction. As large on-chip L2 caches become common in high-performance microprocessors, the leakage power of L2 cache dominates the leakage component of power. Due to the much larger miss penalty, approaches good for L1 caches can not be simply re-applied to L2 caches [8]. Focusing on L2 caches, studies in [8] choose VRC as the leakage power reduction technique for data retention, and propose two time-out based control mechanisms, where the data portion of the L2 cache is shutdown when the cache has not been accessed for a period longer than the time-out threshold. The time-out threshold can be either fixed or dynamically adjusted by ad-hoc methods. Furthermore, feedback control timeout (FCTO) scheme can also be implemented to adjust the time-out threshold with the proportional-integral (PI) feed-back controller. A PI controller has two preset parameters: the *gain* and the cache-miss threshold to adjust the time-out threshold (*setpoint*). The input to the PI controller is the L2 cache miss-rate during a fixed time window and the output of the PI controller is used to adjust the time-out threshold. Table 1A and 1B show the comparison of power reduction and performance loss among DRI cache, selective way cache and FTCO. Note that DRI cache and selective way cache originally designed for L1 caches do not perform well due to the larger miss penalty of L2 caches. Furthermore, only FCTO can guarantee bounded performance loss.

	DRI	SWAY	FCTO
go	56.79%	57.55%	63.80%
li	26.56%	26.64%	27.87%
equake	45.71%	46.40%	48.61%
art	2.18%	2.17%	2.20%

A) Power reduction comparison

	DRI	SWAY	FCTO
go	7.39%	9.95%	1.10%
li	7.71%	7.28%	1.07%
equake	10.58%	9.73%	1.01%
art	3.14%	3.18%	0.92

B) Performance penalty comparison

Table 1: Comparison between *FCTO* and two existing cache power reduction schemes, DRI cache and the selective way (*SWAY*) cache: A) Power reduction, B) Performance penalty.

3. COUPLED POWER AND THERMAL MODELING

3.1 Thermal Modeling and Calculation

HotSpot [16] provides a microarchitecture-level thermal model. HotSpot tracks temperatures at the granularity of individual pipeline units, is independent of boundary and initial conditions, and is parameterized, in the sense that a new compact model is automatically generated for different microarchitectures. It is a simple, portable library that provides an interface for specifying some basic information about the package and for specifying any floorplan that corresponds to the architectural blocks layout. If power dissipations are known over any chosen time step, HotSpot computes temperatures at the center of each block and also at the center of each shared edge between two blocks. The model is also fast, adding less than 0.1% overhead to cycle-accurate power/performance simulations with SimpleScalar/Wattch.

HotSpot's equivalent RC circuit tracks 3D heat flow using two major components, a vertical model to capture 1D heat flow into and within the thermal package, and a lateral model to capture 2D lateral heat flow among pipeline units. The locations and adjacencies of the units are determined by a prior floorplanning step. Fig. 3A shows a simple floorplan that consists of only four blocks. The HotSpot 2D original lateral model creates a thermal RC circuit having a node corresponding to each block and to each adjacent edge between two blocks. This RC model has been calibrated using an independent model of the same system created in Floworks, a commercial, finite-element simulator of 3D fluid and heat flow for arbitrary geometries, materials, and boundary conditions [16]. More recently we have also validated HotSpot against measurements on a commercial thermal test chip [17]. During the validation we have also realized that one of HotSpots main features, its simplicity at the microarchitecture level, can be a weakness for very heterogeneous floorplans for which more detail is needed about the thermal landscape of large units. For this purpose we have recently developed a new thermal model that divides the floorplan into a regular grid as in Fig. 3B, and, similar to the original Hotspot, assigns a node to each grid cell. The advantage now is that the granularity of the thermal model is no longer tied to that of the microarchitecture, but in general the resulting circuit needs to have many more nodes. This does not affect the efficiency of the algorithm though since there are faster methods to solve a regular circuit than for an irregular one [17]. Ideally we would prefer a hybrid scheme that combines the advantages of the original per-functional unit model with the new grid model as in Fig. 3C, but this is part of future work.

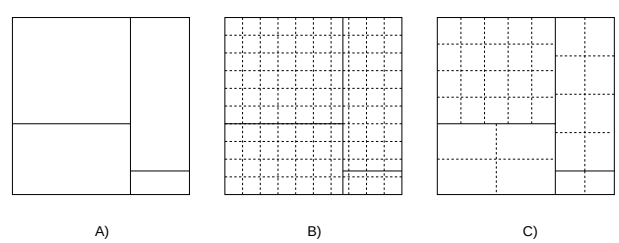

Figure 3: Simple floorplan and use in HotSpot: A) original per functional unit block model, B) new regular grid model, C) hybrid per unit grid model.

3.2 Leakage Model with Temperature Scaling

The leakage power model in [6] does not consider temperature scaling explicitly. In [10] the authors propose temperature scaling for both P_s of memory-based units and I_{avg} in logic circuits. Such model is further improved in [18] by considering temperature scaling according to the BSIM 3v3 subthreshold leakage current model. For 100nm technology, the formulas for P_s of memory-based units and I_{avg} for logic circuits become:

$$P_s = P_{ckts} + P_{cells} \quad (3)$$

$$P_{ckts}(T, V_{dd}) = V_{dd} * T^2 * e^{\left(-\frac{-614.98*V_{dd}+3528.43}{T}\right)} * (5.3 * 10^{-10} * words + 1.7 * 10^{-10} * word_size) \quad (4)$$

$$P_{cells}(T, V_{dd}) = V_{dd} * T^2 * e^{\left(-\frac{-711.92*V_{dd}+3725.53}{T}\right)} * (5.29 * 10^{-10} * words * word_size) \quad (5)$$

$$I_{avg}(T, V_{dd}) = I_s(T_0, V_0) * T^2 * e^{\left(-\frac{-614.98*V_{dd}+3528.43}{T}\right)} \quad (6)$$

where P_{cells} is the leakage power dissipated by SRAM memory cells, P_{ckts} is the power generated by companion logic circuits such as wordline drivers, precharge transistors, etc, and I_s is a constant value for a given reference temperature T_0 and voltage V_0. The coefficients in (3) - (6) are different for different technologies and different leakage reduction techniques [18]. Only subthreshold leakage power is considered in both [10] and [18] as gate leakage is not sensitive to temperature [19] and can be easily integrated into the leakage power model as a constant.

Similar temperature scaling has been proposed in other leakage power models. In [7] the dependence of leakage power on temperature is modeled by an exponential distribution for the ratio of leakage power to dynamic power as a function of temperature T. In [20], a second-order polynomial approximation is applied to describe the temperature and V_{dd} dependencies of leakage power.

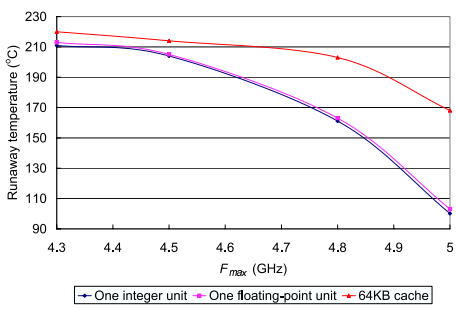

Figure 4: Thermal Runaway temperatures.

3.3 Thermal Runaway

The thermal runaway problem in MOSFETs due to the positive feedback loop between on-resistance, temperature and power is well known [21]. In [18] another thermal runaway problem is presented as the result of the interaction between leakage power and temperature. As component temperature increases, the leakage power increases exponentially. The increase of power consumption can further increase the temperature until the component is in thermal equilibrium with the package's heat removal ability. But if the heat removal ability is not adequate, and the temperature and leakage power interact in a positive feedback loop, both can keep increasing (theoretically to infinity), leading to thermal runaway and catastrophic thermal failure. By assuming no throttling[2] and constant power consumption, [18] defines two criteria as sufficient and necessary conditions for thermal runaway and theoretically proves that the criteria are equivalent to $\frac{d^2T}{dt^2} > 0$, where T is temperature and t is time. The lowest temperature to meet thermal runaway criteria are defined as the *runaway temperature*. As long as the transient temperature reaches the runaway temperature, thermal runaway can not be avoided and the transient temperature will increase until failure if no appropriate thermal management is applied. With the thermal model in Section 3.1 and power model in [18], we plot the runaway temperatures in Fig. 4. It is easy to see that the runaway temperature decreases as clock increases, and for clock frequencies faster than 5.0GHz, the runaway temperatures for functional units can be lower than the maximum temperature constraint 110^oC widely supported by current packaging techniques. Therefore, thermal runaway may become a severe problem in the near future as clock rates continue increasing. Special thermal management schemes are required to combat this problem.

[2]Any mechanism that slows down the processor execution can be categorized as throttling.

3.4 Coupled Power and Thermal Simulation

Coupled power and thermal simulation has been studied [10,18]. In this subsection three important issues are discussed: (1) simulation speedup by time stepping; (2) the effect of clock gate on leakage energy; and (3) system level leakage power variation at different operating temperatures.

In the thermal model, the thermal time constants for components are usually in the order of millisecond and millions of simulation cycle. Therefore, it is not necessary to update temperature and power every cycle. [10] shows that negligible running time overhead for coupled thermal and power simulation is introduced by updating temperature and power value for every time step $t_s >$ 1000 cycles, and negligible temperature and power calculation error is introduced for $t_s <$ 0.5% of thermal time constant. Similarly, t_s = 10000 cycles is chosen in [16] with negligible error on temperature calculation.

Due to its exponential dependence on temperature, leakage energy can be greatly affected by mechanisms which significantly reduce system power and temperature. Clock gating [22] reduces dynamic power by turning off the clock signal for idle components. It is shown in [10] that clock gating actually can indirectly affect leakage energy consumption by changing the temperatures of system components and reduces total leakage energy by up to 48%.

[18] presents the system-level total leakage power consumption at different operating temperatures and shows that by changing the temperature from 85^oC to 110^oC, the total leakage energy can change by a factor of $2X$. Clearly, any study regarding leakage power is not accurate if the temperature dependence of leakage power is not considered. Furthermore, since leakage is a non-trivial component of total power for common temperatures, by extension, the temperature dependence of total power must also be considered.

3.5 Dynamic Power and Thermal Management

Dynamic power and thermal management (DPTM) for microprocessors is implemented by considering the interdependence between power and temperature in dynamic thermal management mechanisms, which dynamically throttle the processor to keep the temperature below the predefined maximum temperature constraint. A thermal violation happens if the maximum on-chip temperature exceeds the maximum temperature constraint. In this section, we first discuss a number of dynamic thermal management mechanisms, and then present an example to show the importance of considering temperature dependence of leakage power in dynamic thermal management.

3.5.1 Dynamic Thermal Management

Dynamic thermal management mechanisms are triggered to control processor temperature whenever the maximum processor temperature exceeds a predefined threshold (lower than the maximum temperature constraint). Fetch toggling [16,23,24] toggles the fetch

engine such as I-cache, I-TLB, branch prediction and decode units to regulate temperature. Toggling can be performed at a certain duty cycle of x/y, which means that the fetch engine operates at full capacity for x cycles and then stalls for $y - x$ cycles [16]. Dynamic frequency scaling (DFS) and dynamic voltage scaling (DVS) [16, 23, 25] control temperature by adjusting the clock frequency and supply voltage V_{dd}. For each change in clock frequency or V_{dd}, the whole processor must stall for 10-50 μs to accommodate resynchronization of the clock's phase-locked loop (PLL) [16]. The long stall time leads to large performance penalty and becomes the disadvantage of DVS and DFS. Activity migration [16, 26] introduces extra copy of components (e.g. register files) and migrates computation activities from one copy with high temperature to the other with low temperature. Another example of activity migration is the dual-pipeline scheme proposed in [27] where a low-power secondary pipeline is implemented as an alternate pipeline when the primary processor pipeline is overheated. Activity migration can also be considered a limited form of multi-clustered architecture [16]. However, none of the above work considers the interdependence between leakage and temperature.

3.5.2 *Need of Temperature Dependent Leakage Model*

In this section the dynamic thermal management is studied by forms of fetch toggling with the proportional-integral (PI) feedback controller. The input of the PI controller is the highest on-chip temperature and the output of the PI controller is used to adjust instruction fetch rate by throttling L1 instruction cache, ITLB, branch predictor and decode units with clock gating. Two leakage power models are implemented: (1) the *accurate* model where leakage power has temperature dependence, and (2) the *simple* model where leakage power is fixed and calculated at nominal temperature 85°C. With the PI controller designed by the simple model, Fig. 5 plots the transient temperature curves simulated for both the simple and accurate models. For the simple model, it appears that the feedback thermal control effectively limits the maximum on-chip temperature below the maximum temperature constraint. However, this appearance is erroneous due to underestimated leakage power. Simulating with accurate leakage model, we can see that the PI controller actually can not prevent thermal constraints violation. If we design the PI controller according to the simple model, the controller may fail to prevent the temperature from exceeding the maximum temperature constraint. Clearly, we must consider accurate temperature dependent leakage modeling in the study of DPTM.

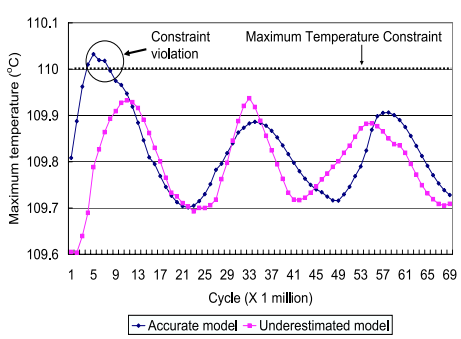

Figure 5: Transient temperature curves obtained by the accurate model and the simple model that underestimates leakage. The benchmark is *gcc*.

4. VOLTAGE SCALING WITH TEMPERATURE DEPENDENT LEAKAGE

In this section we study the following problem: given different packaging and cooling techniques, we consider voltage scaling with dynamic power and thermal management (DPTM) such that system performance is maximum. The system performance is defined as throughput BIPS (Billion Instruction Per Second) in (7):

$$Throughput = \frac{IPC \times clock_frequency}{10^9} \tag{7}$$

where $clock_frequency$ is the processor clock frequency. In order to select appropriate clock frequency under different supply voltage V_{dd} and operating temperature, we develop the delay model with voltage and temperature scaling. We conduct experiments using PTscalar toolset considering the interdependence between leakage, delay and temperature [28].

Similar problem is studied in [29]. Our approaches differs from those in [29] by: (1) using mean throughput over a given workload instead of clock frequency as the performance metric, and (2) consideration of dynamic power and thermal management.

4.1 Delay Model with Voltage and Temperature Scaling

For VLSI circuits, the relationship between circuit delay and supply voltage V_{dd} is $delay \propto V_{dd}/(V_{dd} - V_t)^{\alpha}$, where V_t is the threshold voltage and α is a fitting coefficient. Temperature also affects circuit delay by affecting carrier mobility and threshold voltage [30]. The delay model with temperature and voltage scaling is shown in (8):

$$delay \propto \frac{V_{dd}T^{\beta}}{(V_{dd} - V_t)^{\alpha}} \tag{8}$$

where $\alpha = 1.2$ and $\beta = 1.19$ are coefficients for 100nm technology decided by SPICE simulation and curve fitting empirically.

By assuming the maximum clock frequency $f_{max} = 1/delay$, the appropriate supply voltage to achieve f_{max} can be decided by (9):

$$f_{max} \propto \frac{(V_{dd} - V_t)^{1.2}}{V_{dd}T^{1.19}} \tag{9}$$

4.2 System Performance with Air Cooling

In this subsection we assume the air cooling techniques with heatsink thermal resistance 0.8^{o}C/W. Same as Section 3.5.2, we choose the PI controller and fetch toggling mechanism for DPTM. We design PI controller by choosing setpoint as 5°C lower than the maximum temperature constraints and fix gain as 1.0. Table 2 shows that the maximum throughput with DPTM is 11% higher compared to the maximum throughput without DPTM. Figure 6 further presents the performance impact of DPTM under V_{dd} and temperature scaling. It has been assumed in literature that higher V_{dd} always leads to faster system clock and higher throughput. However, higher V_{dd} leads to larger power consumption and higher temperature, which result in more throttling and larger IPC loss in DPTM. Therefore, higher V_{dd} does not always guarantee better throughput. Figure 6 shows that by increasing V_{dd} from 1.2V to 1.3V, throughput can actually be reduced by up to 35%. Clearly, optimal Vdd for the best throughput may not be the largest Vdd with the presence of DPTM. Voltage scheduling schemes may have to consider the thermal impact on performance, in order to decide the optimal V_{dd} for maximum throughput.

	Without DPTM	With DPTM	With DPTM improved
Performance (BIPS)	4.56	5.04	11%

Table 2: Performance comparison between cases without DPTM and with DPTM. Results are the the average over six SPEC 2000 benchmarks: *art, bzip2, equake, gcc, gzip* and *mesa*.

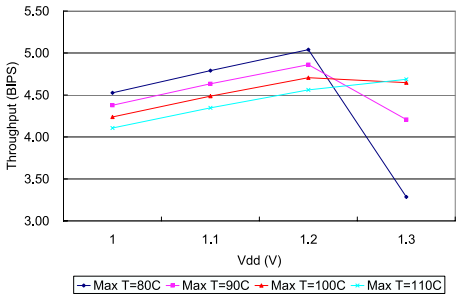

Figure 6: Average throughput with DPTM under different V_{dd} and maximum temperature constraints for six SPEC 2000 benchmarks: *art, bzip2, equake, gcc, gzip* and *mesa*.

4.3 Impact of Advanced Cooling Techniques

Better cooling techniques can help to reduce system thermal resistance, dissipate heat more quickly, and enable faster clocks. Novel cooling techniques include cooling studs, microbellows cooling, microchannel cooling [31] and direct water spray-cooling on electronic devices [32]. In this subsection, we consider two representative heatsink thermal resistances: (1) $R_t = 0.8^o$C/W for the conventional cooling, and (ii) $R_t = 0.067^o$C/W for water spray-cooling in [32], which we name as *active cooling*, and study the impact of active cooling.

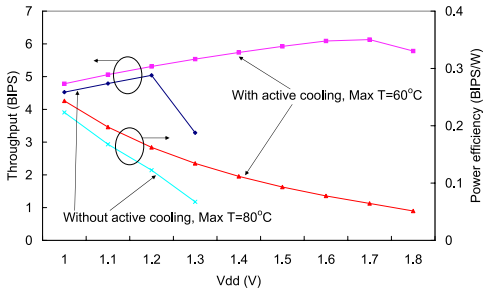

Figure 7: Average throughput and power efficiency under different V_{dd}, maximum temperature constraints and different cooling conditions for six SPEC 2000 benchmarks: *art, bzip2, equake, gcc, gzip* and *mesa*.

With active cooling, the maximum on-chip temperature is greatly reduced. As consequences, we can (1) reduce the maximum temperature constraint; and (2) increase V_{dd}, both of which enable faster clock frequency and larger solution space for better throughput. Figure 7 compares the performance and power efficiency (power/throughput) between cases with and without active cooling. It shows that active cooling not only increases maximum throughput by 30%, but also slows down the decay of power efficiency as V_{dd} increases and improves maximum power efficiency by 9%. Traditionally the research of cooling techniques are only limited to mainframe computers. Our results in Figure 7 clearly indicate that they can also be effective and may become necessary for microprocessors.

5. CONCLUSIONS AND DISCUSSIONS

This paper presents system-level leakage-power modeling and reduction considering the interdependence of temperature and leakage power. Related issues such as microarchitecture-level leakage power characteristics, runtime leakage reduction on caches, thermal runaway and dynamic power and thermal management (DPTM) are also discussed. Finally, a case study is present on voltage scaling considering DPTM for different packaging options.

Acknowledgment

The authors would like to thank their colleagues and students at UCLA and UVa for discussions that have helped shaped this work.

6. REFERENCES

[1] A. Agarwal et al., DAC, 2004.
[2] A. Chandrakasan et al. *Design of High-Performance Microprocessor Circuits*. IEEE Press, 2001.
[3] A. S. Grove. International Electron Devices Meeting, Dec 2002.
[4] L. He, W. Liao and M. Stan, *System Level Leakage Reduction Considering Leakage and Thermal Interdependency*. Technical Report, UCLA, http://eda.ee.ucla.edu/publications.html, 2004.
[5] S. Mutoh et al. IEEE J. of Solid-State Circuits, 1995.
[6] J. Butts and G. Sohi, MICRO'33, 2000.
[7] K. Skadron et al University of Virginia, Department of Computer Science Technical Report, 2003.
[8] W. Liao et al. ICCAD, 2002.
[9] K. Kumagai et al. Symposium on VLSI Circuits, 1998.
[10] W. Liao et al. ISLPED, 2003.
[11] S.-H. Yang et al. HPCA, 2001.
[12] D.Albonesi. MICRO'32, 1999.
[13] K. Flautner et al. ISCA, 2002.
[14] H. Zhou et al. IEEE PACT, 2001.
[15] S. Velusamy et al. Workshop on Memory Performance Issues, in conjunction with ISCA-29, 2002.
[16] K. Skadron et al. ISCA, 2003
[17] W. Huang et al. DAC, 2004.
[18] W. Liao and L. He, The 3rd Workshop on Power-Aware Computer Systems, 2003.
[19] "well-tempered" bulk-si NMOSFET device home page, in *http://www-mtl.mit.edu/Well/*.
[20] H. Su et al. ISLPED, 2003.
[21] R. Severns, Siliconix applications note AN83-10, 1983.
[22] V. Tiwari et al. DAC, 1998.
[23] D. Brooks and M. Martonosi, HPCA, 2001.
[24] K. Skadron et al. HPCA, 2002.
[25] M. Huang et al. MICRO'33, 2000
[26] S. Heo et al. ISLPED, 2003.
[27] C.-H. Lim et al. ISQED, 2002.
[28] http://eda.ee.ucla.edu/PTscalar/, 2004.
[29] A. Vassighi et al. DAC, 2004.
[30] R. Cobbold, Electronic Letters, 1966.
[31] H. B. Bakoglu, Circuits, Interconnections, and Packaging for VLSI. Addison-Wesley, 1990.
[32] M. Shaw et al. 8th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems, 2002.