COUPLED POWER AND THERMAL SIMULATION AND ITS APPLICATION

Weiping Liao and Lei He

Electrical Engineering Department
University of California, Los Angeles, CA 90095
{wliao, lhe}@ee.ucla.edu

Abstract

Power is rapidly becoming the primary design constrain for systems ranging from server computers to handhelds. In this paper we study microarchitecture-level power modeling and management with temperature and voltage scaling. We develop an accurate temperature-dependent leakage power model and efficient temperature calculation, and show that leakage energy and total energy can be different by up to $10X$ and 30% for temperatures between 35^{o}C and 110^{o}C, respectively. Given the growing significance of leakage power and its sensitive dependence on temperature, no power modeling at microarchitecture is accurate without considering dynamic temperature calculation. Furthermore, we discuss a new thermal runaway phenomenon induced by the temperature dependent leakage power and show that in the near future thermal runaway could be a severe problem. We also study the microarchitecture level coupled power and thermal management by clock gating and novel active cooling techniques. We show that with thermal constraints, clock gating can increase maximum system clock by up to $1.5X$ and reduce leakage energy by up to 68.5% compared to the cases without clock gating, and active cooling techniques providing smaller thermal resistance can further increase the maximum clock by a factor of $2.44X$.

1. INTRODUCTION

Power is rapidly becoming the primary design constraint for systems ranging from sever computers to handhelds. As semiconductor technology scales to smaller feature sizes, leakage power increases exponentially because transistor threshold voltages are reduced in concert with supply voltage to maintain transistor performance. For current high-performance design methodologies, the contribution of leakage power increases at each technology generation [1]. The Intel Pentium IV processors running at 3GHz already have an almost equal amount of leakage and dynamic power [2]. Therefore, considering leakage is of paramount importance in future designs.

Considering supply voltage is critical in power and energy modeling. Voltage scaling [3] is widely applied to reduce system energy consumption. Because both dynamic and leakage energy exhibit strong dependence on supply voltage, quadratic and exponential dependence, respectively [1], any study focused on power and energy must consider voltage scaling for accuracy.

In addition to power and energy, thermal constraints are another important issue in microprocessor designs. Thermal stress caused by high on-chip temperature and large temperature differentials between functional units may lead to malfunction of logic circuits, p-n junction breakdown, and clock skew [4] or ultimate physical failure of the microprocessor chip. As leakage power becomes more significant, these thermal problems are exacerbated since leakage power also has an exponential dependence on temperature [1].

Given the above discussion, obviously any study considering power and thermal management for future technology generations must consider both dynamic and leakage power, and in addition, the dependence of each on voltage and temperature, in order to provide an accurate characterization of system behavior.

1.1. Prior Related Work

Current microarchitecture-level power simulators [5, 6, 7] calculate leakage power by assuming a ratio between dynamic and leakage power. For example, [7] determines the ratio for logic circuits by SPICE simulations on typical circuits and also presents formula-based models for leakage power in memory based units. Such models [5, 6, 7] are efficient for microarchitectural simulations because dynamic power calculations within the model can therefore automatically compute the contribution of leakage power. However, as discussed previously, this assumption is not accurate as dynamic power and leakage power scale differently as a function of supply voltage and temperature. [8] presents the following formula to estimate leakage power P_s:

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

where V_{dd} is the supply voltage, N_{FET} is the number of transistors, k_{design} is a design dependent parameter, and $I_{leakage}$ is a technology dependent parameter. However, no well-defined method to decide k_{design} and $I_{leakage}$ is described. More importantly, [8] does not consider temperature scaling and the exponential supply voltage dependence of $I_{leakage}$.

[9] proposes a leakage power model with temperature scaling for 100nm technology. Different formulas for logic circuits and memory circuits are proposed in [9]. For logic circuits, the leakage power is calculated as the product of gate count (N_{gate}) and the average leakage current per gate (I_{avg}), as shown in (2):

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

Memory-based units are modeled as SRAM arrays. [9] provides temperature dependent formulas for both I_{avg} for logic circuits

This paper is partially supported by NSF CAREER award CCR-0093273, SRC grant 1008, a UC MICRO grant sponsored by Analog Devices, Fujitsu Laboratories of America, Intel and LSI Logic, and a Faculty Partner Award by IBM. We used computers donated by Intel and SUN Microsystems. Address comments to lhe@ee.ucla.edu.

and P_{so} for SRAM arrays. However, the temperature dependence is characterized by a purely empirical exponential term $exp(\frac{-a}{T-b})$ without providing a theoretical model where a and b are coefficients and T is the temperature which can be extended to future technology generations. Voltage scaling is not consider for either dynamic or leakage power in [9]. [9] considers thermal calculation based on the whole chip and individual modules, but the thermal resistance for all modules only have relative value and refer to the thermal resistance of one integer unit decided by an empirical assumption. [9] studies the on-chip temperature up to 130^{o}C, which is too high for current packaging techniques to support. Furthermore, clock gating in [9] is assumed to reduced 100% dynamic power and therefore too ideal. An earlier work [10] proposes chip-level thermal calculation similar to the universal mode in [9]. However, [10] does not consider temperature dependence for leakage power either.

HotSpot [11] provides a thermal model based on an equivalent circuit of thermal resistances and capacitances that correspond to microarchitectural units and also the chip-level thermal package. The thermal calculation in HotSpot considers horizontal heat transfer and detailed modeling of heat spreader and heatsink. However, both temperature modeling and dynamic thermal management in HotSpot fail to consider the temperature and voltage dependence of leakage power. Note that it is straightforward to integrate our temperature and voltage scalable leakage model with any thermal calculation considering horizontal heat transfer. In fact, we have developed a coupled power and thermal simulator PTscalar, which considers the leakage model developed in this paper and thermal modeling similar to that in HotSpot, but uses a more efficient thermal calculation to consider horizontal heat transfer. This tool will be made available at http://eda.ee.ucla.edu.

1.2. Our Contributions in This Work

In this paper, we present power models with clock, voltage, and temperature scaling based on the BSIM2 subthreshold leakage current model, and discuss its applicability to gate leakage. We develop a coupled thermal and power microarchitecture simulator considering interdependence between leakage and temperature. With this simulator, we are able to accurately simulate the inter-dependence between power and temperature and evaluate microarchitectural power and thermal management techniques. We show the dramatic dependence of leakage power on temperature at the microarchitecture level based on the reasonable thermal resistance and chip area of Intel Itanium 2 processors within the temperature range between 35^oC and 110^{o}C. We also theoretically discuss leakage-induced thermal runaway. These studies underscore the need for coupled power and thermal management.

We further study the impact of clock gating and active cooling techniques on coupled power and thermal management. We study both ideal clock gating reducing 100% dynamic power and realistic clock gating reducing only 75% dynamic power due to implementation overhead. We show that compared to the cases with no throttling and current packaging and cooling techniques, clock gating can increase maximum system clock by up to $1.5X$ and reduce leakage energy by up to 68.5% compared to the cases without clock gating. We further study active cooling techniques providing smaller thermal resistance and show that they can improve performance with the same thermal constraints by a factor of $2.44X$.

The rest of this paper is organized as follows. In Section 2, we develop dynamic and leakage power models with both voltage and temperature scaling. In Section 3, we introduce both transient and stable-state temperature calculation and the experiments on thermal-sensitive energy simulations at microarchitecture-level. In Section 4, we present the the impact of coupled power and thermal management considering clock gating and novel active cooling techniques. We conclude and discuss future work in Section 5.

2. POWER MODEL WITH TEMPERATURE AND VOLTAGE SCALING

We define three power states: (i) $active$ *mode*, where a circuit performs an operation and dissipates both dynamic power (P_d) and leakage power (P_s). The sum of P_d and P_s is active power (P_a). (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 [12] or other leakage reduction techniques, and dissipates a reduced leakage power defined as inactive power (P_i). A circuit in the inactive mode requires a non-negligible amount of time to wake up and then perform an useful operation [7].

In cycle accurate simulations, power is defined as the energy per clock cycle. Therefore, P_d is equal to $\frac{1}{2}f_sCV^2$ where C is the switching capacitance, V is the supply voltage and f_s is the switching factor per clock cycle. In essence, P_d is the energy to finish a fixed number of operations during one cycle. Consistently, P_s is defined as $P_{so} * t$ where P_{so} is leakage power per second and t is the clock period. Same as P_s, $P_i = P_{io} * t$ is proportional to the clock period with P_{io} being reduced leakage power in the inactive mode.

2.1. Dynamic Energy with Voltage Scaling

Dynamic energy is consumed by charging and discharging capacitances. It is independent of temperature, but has a quadratic dependence with supply voltage. For VLSI circuits, the relationship between circuit delay and supply voltage V_{dd} is $delay \propto V_{dd}/(V_{dd} - V_T)^2$, where V_t is the threshold voltage. By assuming the maximum clock $f_{max} = 1/delay$, the appropriate supply voltage to achieve f_{max} can be decided by (3):

$$f_{max} \propto (V_{dd} - V_T)^2/V_{dd} \qquad (3)$$

Therefore, the dynamic energy for each cycle varies to achieve different f_{max}.

2.2. Leakage Estimation with Voltage and Temperature Scaling

2.2.1. Leakage Model with Temperature and Voltage Scaling

We improve the temperature dependent leakage model in [9] for 100nm technology with voltage scaling and more realistic temperature scaling according to the BSIM2 subthreshold current model [1] as shown in (4):

$$I_{sub} = Ae^{\frac{(V_{GS}-V_T-\gamma V_{SB}+\eta V_{DS})}{nV_{TH}}}\left(1-e^{-\frac{V_{DS}}{V_{TH}}}\right) \qquad (4)$$

$$A = \mu_0 C_{ox}\frac{W}{L_{eff}}{V_{TH}}^2 e^{1.8} \qquad (5)$$

	Logic circuits				Memory based units		
	X	Y	α	β	Z	γ	δ
With power gating	3.5931e-12	1.2080e-11	-1986.1263	4396.0880	8.7286e-11	-443.2760	3886.2712
Without power gating	5.2972e-10	1.7165e-9	-614.9807	3528.4329	5.2946e-10	-711.9226	3725.5342

Table 1: Coefficients in (6) - (9) for 100nm technology, where MTCMOS and VRC are the power gating techniques for logic and SRAM arrays, respectively.

Circuit	Temperature (^{o}C)	V_{dd}	I_{avg} or P_{so}		abs. err. %
			formula	SPICE	
adder	100	1.3	0.0230	0.0238	3.74
	50	1.3	0.00554	0.00551	0.71
multiplier	100	1.3	0.0209	0.0217	3.83
	50	1.3	0.00493	0.00506	2.63
shifter	100	1.3	0.0245	0.0255	3.92
	50	1.3	0.00592	0.00585	1.32
SRAM 128x32	50	1.3	54.1	56.8	4.81
	50	1.0	21.62	22.31	3.07
SRAM 512x32	50	1.3	211.7	227.2	6.85
	50	1.0	84.41	88.83	4.98

Table 2: Comparison between our formula and SPICE simulation. I_{avg} and P_{so} are for logic circuits and SRAM arrays, respectively. The SRAM arrays are represented as “row number” x “column number”. The units for I_{avg} and P_{so} are uA and uW, respectively.

where V_{GS}, V_{DS} and V_{SB} are the gate-source, drain-source and source-bulk voltages, respectively; V_T is the zero-bias threshold voltage, V_{TH} is the thermal voltage $\frac{kT}{q}$, γ is the linearized body-effect coefficient, η is the Drain Induced Barrier Lowering (DIBL) coefficient, μ_0 is the carrier mobility, C_{ox} is gate capacitance per area, W is the width and L_{eff} is the effective gate length.

From (4) we can see the temperature scaling for leakage current is $T^2 e^{-\frac{1}{T}}$, where T is the temperature, and the voltage scaling for leakage current is $e^{-(\alpha V_{dd}+\beta)}$, where α and β are parameters to be decided. Based on these observation, we propose the following formula for I_{avg} considering the temperature and voltage scaling:

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

where I_s is a constant value for the reference temperature T_0 and voltage V_0. The coefficients α and β are decided by circuit designs. Values for α and β as well as validation of (6) will be presented in Section 2.2.2.

We also improve the formula in [9] with better temperature and voltage scaling as shown in (7) - (9):

$$\begin{aligned} P_{so} &= P_{circuits} + P_{cells} && (7) \\ P_{circuits}(T, V_{dd}) &= (X * words + Y * word\_size) && (8) \\ & \quad * V_{dd} * T^2 * e^{\left(-\frac{\alpha * V_{dd} + \beta}{T}\right)} && \\ P_{cells}(T, V_{dd}) &= (Z * words * word\_size) && (9) \\ & \quad * V_{dd} * T^2 * e^{\left(-\frac{\gamma * V_{dd} + \delta}{T}\right)} && \end{aligned}$$

where P_{cells} is the leakage power dissipated by SRAM memory cells and $P_{circuits}$ is the power generated by the circuits such as wordline drivers, precharge transistors, and etc. $P_{circuits}$ essentially has the same format as (2) as $X * words + Y * word\_size$ in (8) can be viewed as N_{gate}, and the scaling in $P_{circuits}$ is same as (6). P_{cells} is proportional to the number of SRAM memory cells. X, Y, Z, γ and δ in (8) and (9) are coefficients decided by circuit designs. Values for X, Y, Z, γ and δ as well as validation of (8) and (9) will be presented in Section 2.2.2.

2.2.2. *Leakage Model Validation*

We collect the power consumption for different types of circuits at a few temperature levels by SPICE simulations. We then obtain the coefficients in (6) - (9) by curve fitting. Table 1 summarizes the coefficients for ITRS 100nm technology we used. Table 2 compares our high-level leakage power estimation for logic circuits and SRAM arrays with SPICE simulations in ITRS 100nm technology. We use different circuits and temperature during curve fitting and verification. The overall difference between our formulas and SPICE simulation is less than 7%.

3. COUPLED POWER AND THERMAL SIMULATION

3.1. Temperature Calculation

We develop the thermal model based on conventional heat transfer theory [13]. The stable temperature at infinite time can be calculated according to (10):

$$T \quad = \quad T_a + R_t * P \qquad (10)$$

where T is the temperature, T_a is the ambient temperature, P is the power consumption, and R_t is the thermal resistance, which is inversely proportional to area and indicates the ability to remove heat to the ambient under the steady-state condition. According

Component	Configuration
Decode	6-issue width
BTB	512 entries 4-way associative, Two-level predictor
Register file	128 integer and 128 floating-point registers with 64-bit data width
Memory	page size 4096 bytes, latency 30 cycles
Memory Bus	8 bytes/cycle

ALU	Number	Latency
Integer (IALU)	4	1 cycle for add, 2 cycles for multiply and 15 cycles for division
Floating-point (FPU)	2	2 cycles for add/multiply, 15 cycles for division

Cache	Size	Block size	Associativity	Policy
L1 Instruction	64 KB	32 bytes	4	LRU
L1 Data	64 KB	32 bytes	4	LRU
L2	2MB	64 bytes	8	LRU

Table 3: System configuration for experiments.

to (10), the heat loss to ambient can be modeled as $P_o = (T - T_a)/R_t$.

The unbalance between total power consumption P and heat loss to ambient P_o leads to the transient temperature T characterized by (11):

$$P - P_o = C_t \dot{T} \tag{11}$$

where C_t is the thermal capacitance. By substituting P_o in (11) with $(T - T_a)/R_t$ we can get the differential equation (12):

$$R_t C_t \dot{T} + (T - T_a) = R_t P \tag{12}$$

where $\dot{T} = \Delta T/\Delta t$ and ΔT is the temperature change after a short time period Δt. By manipulating (12) we can get (13) for the temperature change ΔT:

$$\Delta T = \frac{PR_t - (T - T_a)}{\tau} \Delta t \tag{13}$$

where $\tau = R_t C_t$ is the thermal time constant. By solving (13) we can obtain an exponential form for temperature T in terms of time t and power, as shown in (14):

$$T = PR_t + T_a - (PR_t + T_a - T_0) \times e^{-\frac{t - t_0}{\tau}} \tag{14}$$

where T and T_0 are temperatures at two different time points t and t_0. This exponential form clearly shows that the power has a *delayed* impact on the temperature. Note that our cycle-accurate simulation uses (13) directly to avoid the time-consuming exponential calculation.

Same as [9], in our thermal model, we have two different modes with different granularities to calculate the temperature: (i) *individual mode*. We assume that there is no horizontal heat transfer between components, and calculate a temperature for each individual component. In general, the horizontal heat reduces the temperature gaps between components. So the individual mode essentially gives the upper bound of the highest on-chip temperature and temperature gap. (ii) *universal mode*, which is similar to the thermal model in TEM^2P^2EST [10]. We assume the whole processor as a single component with a uniform thermal characteristic and temperature. The universal mode gives the lower bound of the highest on-chip temperature.

3.2. Experiment Parameter Settings

Although our power and thermal models are applicable to any architecture, we study VLIW architecture in this paper. We integrate our thermal and power model into the PowerImpact [7] toolset. The microarchitecture components in our VLIW processor include BTB, L1 instruction cache, L1 data cache, unified L2 cache, integer register file, floating-point register file, decoder units, integer units (IALUs) and floating-point units (FPUs). Among them, BTB, caches and register files are memory-based units, while the others are logic circuits. When calculating the power of memory-based units, we first partition the component into pieces of SRAM arrays with CACTI 3.0 toolset [14], then apply our formulas for power consumption of each SRAM array. The total component power consumption is the sum of power of all SRAM arrays. For IALUs and FPUs, we take the area and gate count in the design of DEC alpha 21264 processor [15], and scale from 350nm technology down to 100nm technology. For decode unit, we simply assume one decode unit has the same area and power consumption as one integer unit.

To obtain a set of reasonable thermal resistances for components, we set the reference as the thermal resistance 0.8 ^{o}C/W for a chip with die size 374 mm^2 similar to Intel Itanium 2 [16]. Based on this reference, for each component, we calculate its thermal resistance as it is inversely proportional to its area. The whole chip thermal resistance is calculated in the same manner. Table 2.2.2 presents the micro-architecture configuration of the VLIW processors we study. Table 4 summarizes the power consumption, the thermal resistances and the areas for all components in our system. According to the thermal time constant for microarchitecture components without consider heatsink in [17], we set the thermal time constants as $\tau = 100us$, which is independent of component area.

To consider appropriate supply voltage scaling for varying clock, we assume that V_t is 20% of V_{dd} and $V_{dd} = 1V$ obtains 3GHz clock as specified by the ITRS. According to Equation (3) the corresponding V_{dd} for a range of clocks in our experiments is shown in Table 5.

3.3. Chip Temperature

In our experiments, we update temperatures after each time step t_s. We then update the power value with respect to new temperature for each t_s. Smaller t_s gives a more accurate transient temperature analysis, e.g., $t_s = 1$ cycle represents the cycle accurate tempera-

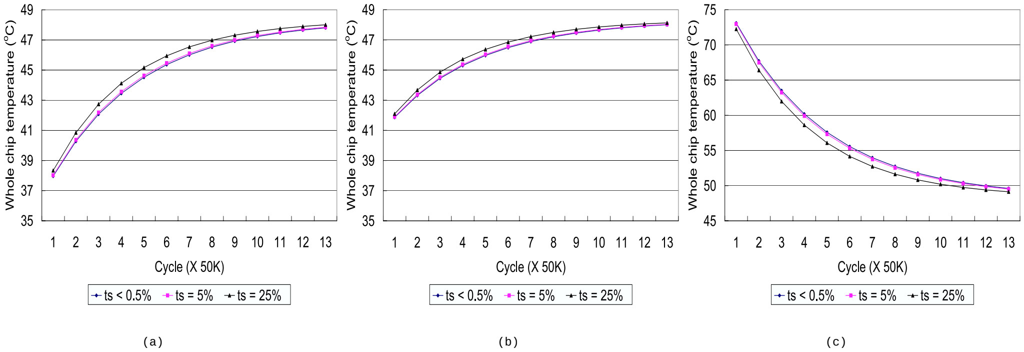


Figure 1: Whole chip temperature curve obtained by the universal mode for different time step t_s. The clock frequency is 2GHz. Three different starting temperatures are chosen: (a) 35^{o}C; (b) 40^{o}C; and (c) 80^{o}C. No throttling is applied. Therefore, the results are independent of benchmarks.

Component	P_a	P_s	P_i	R_t (K/W)	$Area$ (mm^2)
BTB	119	1.23	0.0504	64.4	1.63
L1 Instruction Cache	535	1.145	0.0458	22.129	4.74
L1 Data Cache	460	1.145	0.0458	20.967	4.99
Unified L2 Cache	1858	34.2	1.37	1.401	59.8
Integer Register File	59.6	0.027	0.0011	24.692	4.24
FP Register File	35.8	0.0275	0.0011	84.844	1.24
One Decode Unit	79.2	0.68	0.0068	236.355	0.44
One IALU	79.2	0.68	0.0068	236.355	0.44
One FPU	158	0.68	0.0068	125.599	0.83

Table 4: Power consumption (in pJ/cycle), thermal resistance R_t and areas for all components. For 100nm technology, we choose 1V supply voltage and 3GHz clock rate as specified by the ITRS. The decode, integer ALU and FPU are only one unit among total six, four and two units. The temperature is 35^oC. Note the P_s is relative small due to the low temperature.

Clock (GHz)	2	3	4	5
V_{dd} (V)	0.667	1.0	1.33	1.667

Table 5: V_{dd} after appropriate voltage scaling for different clocks

ture calculation. Figure 1 plots the transient temperature for whole chip [1] calculated under different t_s shown as the percentages of the thermal time constant, where 0.5% of the thermal time constant is equal to 1000 clock cycles for a 2GHz clock. When $t_s \leq$ 0.5% of the thermal time constant, the temperatures are identical to those with $t_s = 1$ cycle. Observable difference appears when t_s is increased to 5% of the thermal constants and significant error is induced when $t_s = 25$% of the thermal constant. Clearly, it is not necessary to update temperatures for each cycle. Since 0.5% of thermal constants always lead to negligible error on temperature calculation compared with the cycle accurate temperature calculation, we only update temperatures and power values after every period of 0.5% of the thermal time constants in the rest of the paper.

Note in Figure 1, we also present transient temperature with different starting temperatures. Clearly, different starting temperatures lead to the virtually same stable temperature without considering the thermal runaway problem which will be discussed in Section 3.5.

3.4. Temperature Dependent Leakage Power and Maximum Clock

Figure 2 shows the experimental results for total leakage energy consumption at 2.5GHz clock. We assume there is no throttling, i.e., P_a is dissipated in every cycle. We study two cases: one assumes a fixed temperatures, and another considers energy consumption with temperature dependence in both individual mode and universal mode. From Figure 2 we can see that by changing the temperature from 35^oC to 110^oC, the total leakage energy can be changed by a factor of $10X$. Figure 2 clearly shows that any study regarding leakage energy is not accurate if the thermal issue is not considered. To consider temperature in methods in [7, 18], the designers need to assume a fixed temperature appropriate for the processor and the environment, and then use leakage values at this temperature. How to decide the appropriate temperature is of paramount importance for accurate energy estimation, and it is an open problem in the literature. Our work actually presents an approach to select the appropriate temperature.

Faster system clock is always desired in the high-performance processor designs. However, as clock increases, the total energy and system temperature both increase as well. The maximum temperature and maximum temperature gap constraints prevent us from increasing the clock rate indefinitely. In the following experiments, we assume the maximum allowable temperature is 110^oC which is the maximum temperature supported by current semiconductor packaging techniques, and the maximum temperature gap among components is 40^oC. We use the individual mode to calculate the maximum temperature and the maximum temperature gap, where the maximum temperature is set as the largest temperature among all components [2]. Table 6 shows the maximum system temperature and the maximum temperature gap without any throttling. We can see that the maximum clock with thermal constraints is about 1.5GHz when there is no throttling.

[1] Memory-based units and functional units behavior similarly

[2] The universal mode gives us a lower bound of the maximum temperature.

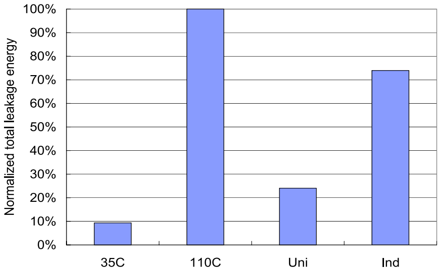


Figure 2: Total Leakage energy consumption without any throttling. We study fixed temperatures of 35^oC and 110^oC, as well as the case with dynamically updated temperature. The cases of "ind" and "uni" stand for the individual mode and universal mode, respectively. The clock is 2.5GHz. Note the results are independent of benchmarks in the no-throttling cases.

Clock (GHz)	0.5	1	1.5	2	2.5
Max Temperature	35.2 - 36.016	36.7 - 41.5	40.7 - 56.7	48.4 - 87.3	61.4 - 157.2
Max Temperature Gap	0.922	3.969	19.187	46.230	110.437

Table 6: Maximum temperatures (*Max T*) and temperature gaps (*Max Gap*) among components for different clocks without any throttling. The unit for temperatures is ^{o}C. The ambient temperature is 35^{o}C. Note the results are independent of benchmarks in the no-throttling cases.

3.5. Thermal Runaway

The MOSFET thermal runaway problem is widely known as due to the positive feedback loop between the on-resistance, temperature and power of MOSFET [19]. In this section we will present another thermal runaway problem due to the interaction between leakage power and temperature. As the component temperature increases, its leakage power increases exponentially. The increase of power consumption further increases the temperature until the component is in thermal equilibrium with the package's heat removal ability. If the heat removal is not adequate, thermal runaway occurs as the temperature and leakage power interact in a positive feedback loop and both increase to infinity. For transient temperature T_0 and T_1 at consecutive time t_0 and t_1 and corresponding power $P(T_0)$ and $P(T_1)$, we define the following two criterion as necessary conditions for the thermal runaway to occur:

1. $T_1 > T_0$, i.e., the temperature should be increasing.
2. the increment of power is larger than the increment of package's heat removal ability. The package's heat removal ability is defined as $P_o(T) = \frac{T-T_a}{R_t}$ where T_a and R_t are ambient temperature and thermal resistance, respectively.

The second criterion can be mathematically formulated as (15) with relationship between T_0 and T_1 defined by (16):

$$P(T_1) - P(T_0) > \frac{T_1 - T_0}{R_t} \quad (15)$$

$$T_1 - T_0 = \frac{P(T_0)R_t - (T_0 - T_a)}{\tau}(t_1 - t_0) \quad (16)$$

where (16) is derived from (13).

In addition to temperatures, (15) and (16) require knowledge of runtime power, R_t, τ and T_a. We can simplify the second criterion with Theorem 1.

Theorem 1 *Criterion (2) is equivalent to* $\frac{d^2T}{dt^2} > 0$, *where T is temperature and t is time.*

Proof: suppose three different temperatures T_0, T_1 and T_2 are measured at consecutive time t_0, t_1 and t_2, where $t_1 - t_0 = t_2 - t_1 = \Delta t$ and Δt is a small time period, then $\frac{d^2T}{dt^2} > 0$ is equivalent to (17):

$$\frac{\frac{T_2 - T_1}{\Delta t} - \frac{T_1 - T_0}{\Delta t}}{\Delta t} > 0 \quad (17)$$

From (16) we can derive (18) and (19):

$$\frac{R_tP(T_1) - (T_1 - T_a)}{\tau} = \frac{T_2 - T_1}{\Delta t} \quad (18)$$

$$\frac{R_tP(T_0) - (T_0 - T_a)}{\tau} = \frac{T_1 - T_0}{\Delta t} \quad (19)$$

On the other hand, from (15) we can get (20):

$$R_tP(T_1) - (T_1 - T_a) > R_tP(T_0) - (T_0 - T_a) \quad (20)$$

Combining (18) - (20) we can get (21):

$$\frac{T_2 - T_1}{\Delta t} > \frac{T_1 - T_0}{\Delta t} \quad (21)$$

From (21) we can prove (17) and then Theorem 1.

On the other hand, by assuming $\frac{d^2T}{dt^2} > 0$ and (16) we can prove (15) following similar derivation. □

Compared to (15) and (16), Theorem 1 provides a simpler mechanism with reduced complexity to detect thermal runaway.

We define the lowest temperature to meet the criterion 1 and 2 as *runaway temperature*. As long as the transient temperature reach the runaway temperature, thermal runaway happens and the transient temperature will increase to infinity if no appropriate thermal management is applied. Figure 3 and plot transient temperature curves with thermal runaway. [3] It clearly shows that as long as the transient temperature reach the runaway temperature, thermal runaway occurs. Note two starting temperatures, 35^oC and 55^oC, are chosen in Figure 3. It is easy to see the starting temperature is independent of transient temperature behavior and thermal runaway is independent of the starting temperature because runaway temperature is decided by the power and the package's heat removal ability.

We calculate the runaway temperature according to criteria 1 and 2 for different clocks. Figure 4 shows the runaway temperatures for clock from 4.5GHz to 6.5GHz. As clock increases, the runaway temperature decreases since the difference between

[3] Memory units such as caches present similar curves and therefore are not shown.

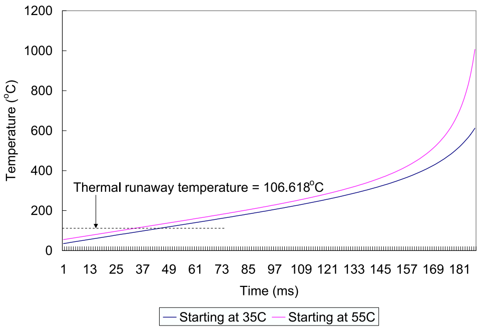


Figure 3: Transient temperature curves one IALU with 5.5GHz clock. By reaching the runaway temperature, the thermal runaway happens and the transient temperature finally increases to infinity. The thermal runaway temperature is labeled. No throttling is applied.

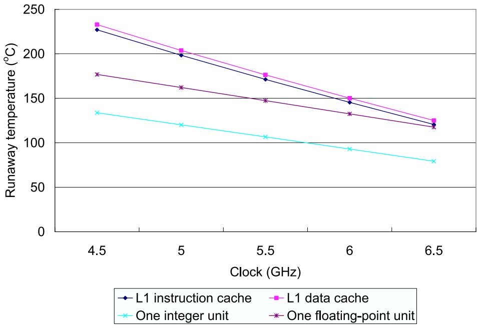


Figure 4: Runaway temperatures for different clocks and different components.

power $P(T_1)$ and $P(T_0)$ increases. For clocks faster than 5.5GHz, the runaway temperatures of integer units are below our maximum temperature constrain 110^oC. In other words, we can not eliminate the thermal runaway by simply limiting the operating temperature to be no more than maximum junction temperature supported by current packaging techniques. We anticipate that thermal runaway could be a severe problem in the near future as the clock keeps increasing. Special thermal management schemes are expected to encounter this problem.

4. POWER AND THERMAL MANAGEMENT

4.1. Clock Gating

Due to the exponential dependence on temperature, leakage energy can be greatly affected by mechanisms which can significantly reduce system power and temperature. In this section we study the impact of clock gating on system temperature and leakage energy consumption.

Clock gating [20] is effective to reduce dynamic power by turning off the clock signal for idle components. As pointed out in [9], clock gating actually can indirectly affect the leakage energy consumption by affecting the temperatures of system components. In this section, we consider two types of clock gating: one is ideal clock gating, i.e. all dynamic power can be eliminated by clock gating. The other is realistic clock gating which can reduce dynamic power by 75%. We present a quantitative study on the impact of clock gating.

Similar to [9], we propose to evenly distribute instructions to functional units and eliminate the temperature gaps between integer units.

4.1.1. *Maximum Clock*

We consider the maximum clock with clock gating under the same maximum on-chip temperature constraint and maximum temperature gap constraint as in Section 3. Table 7 presents the maximum temperatures and maximum temperature gaps with clock gating for different clocks. We do not observe thermal runaway for clocks up to 3GHz. Based on the three thermal constraints we consider, we show in Table 8 the maximum clocks for different benchmarks with and without clock gating. From Table 8 we can see, compared to no throttling cases, by reducing temperature, maximum system clock can be increased under the same thermal constraints from 1.5GHz to 2.2GHz and 2.25GHz for realistic and ideal clock gating, respectively. Such increase corresponds to $1.47X$ and $1.5X$ maximum clock boost for ideal and realistic clock gating, respectively. Different from intuition, realistic clock gating achieves faster maximum clock than ideal clock gating does. There are a few reasons for such a result: first, since realistic clock gating reduces less power than ideal clock gating, the temperature for the less-frequent-accessed components such as L2 cache is higher under realistic clock gating than under ideal clock gating; second, with clock gating, our experiments show that L1 instruction cache has highest temperature. Its temperature varies little with realistic or ideal clock gating because it is busy for the most of the time. As a result, the maximum temperature gap with realistic clock gating is smaller than that with ideal clock gating. Because the maximum temperature gap constrain is the first violation we observed when we increase the clock, realistic clock gating obtains a faster maximum clock.

	Benchmark	Clock (GHz)	0.5	1	1.5	2	2.5	3
Ideal clock gating	go	Max temperature	35.1-35.4	35.4-38.4	36.3-46.4	38.0-61.9	40.9-87.7	45.3-127.9
		Max temperature gap	0.429	3.38	11.4	26.9	52.7	92.9
	$equake$	Max temperature	35.1-35.4	35.5-38.8	36.6-47.7	38.8-65.2	42.4-94.4	47.8-141
		Max temperature gap	0.462	3.72	12.6	29.9	58.9	105
Realistic clock gating	go	Max temperature	35.1-35.4	35.7-38.4	37.3-46.4	40.3-61.9	45.5-87.7	53.2-128
		Max temperature gap	0.394	2.69	10.5	25.1	49.2	86.7
	$equake$	Max temperature	35.1-35.5	35.8-38.8	37.6-47.7	41.0-65.2	46.7-94.4	55.5-141
		Max temperature gap	0.437	3.52	11.9	28.3	55.8	99.3

Table 7: Maximum temperatures and temperature gaps among components for different clocks under different clock gating conditions. The percentages of clock gating represent the dynamic power reduction rates for each clock gating condition. The maximum temperatures are shown as a range where the lower bound and the upper bound are given by the universal mode and individual mode, respectively. The unit is in ^oC.

Benchmark	No throttling	Ideal clock gating	Realistic clock gating
equake	1.5GHz	2.2GHz	2.25GHz
go	1.5GHz	2.25GHz	2.3GHz

Table 8: Maximum clock under thermal constraints.

If we further reduce the power reduction rate in clock gating, we could expect increase of the maximum clock. However, at some points temperature of other components such as IALUs and FPUs can become the highest. Since temperatures of these components increases greatly as power reduction rate of clock gating decreases, temperature gaps will increase again and we can not increase maximum clock any more. Our coupled power and thermal simulation enables the designers to accurately considers three types of thermal constraints.

4.1.2. Leakage Energy Reduction and Total Energy Variation

Traditionally, clock gating is considered as a technique solely for dynamic power reduction. However, As pointed out in [9], clock gating can indirectly reduce the leakage energy by reducing component temperatures. Figure 5 presents the total leakage energy with respect to different clocks, with and without clock gating. From Figure 5 we can see for clocks ranging from 500MHz to 2.25GHz, clock gating can effectively reduce the total leakage energy by up to 68.5%. Furthermore, the faster the clock is, the more leakage energy is reduced by clock gating. The reason is that as clock increases, processors operate at higher temperature and temperature reduction by clock gating becomes more effective in leakage energy reduction due to its temperature dependence. Because system clock will inevitably keep increasing, this conclusion indicates in the future clock gating is more effective to reduce leakage energy.

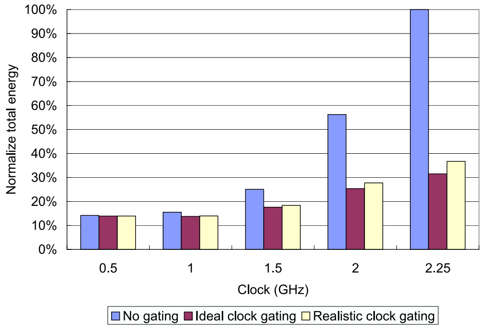


Figure 5: Total leakage energy consumption for different clock and gating scheme. The benchmark is *go*.

Figure 6 and 7 shows the total energy consumption with clock gating for different clocks. As clock gating is applied and leakage energy gains importance, not only the leakage energy, but also the total energy changes dramatically with respect to temperature due to the variation of leakage energy. When the temperature changes from 35^{o}C to 110^{o}C, the total energy increased by up to 30% due to the dramatic change in leakage energy. These results show that studies related to total energy, not only leakage energy, should consider the temperature dependence of the leakage power for accuracy.

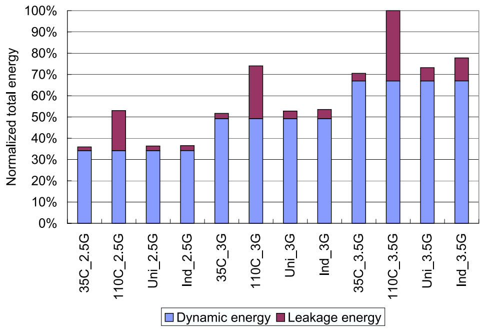


Figure 6: Energy consumption with ideal clock gating. We study fixed temperatures of 35^{o}C and 110^{o}C, as well as the case with dynamically updated temperature. The prefix of “ind” and “uni” stand for the individual mode and universal mode, respectively. The benchmark is *go*.

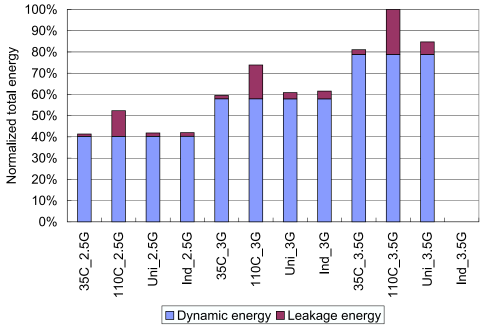
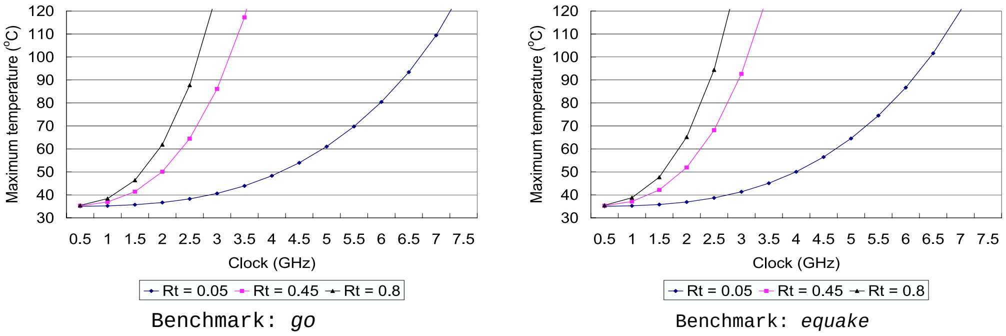


Figure 7: Energy consumption with realistic clock gating. The conditions are the same as those in Figure 2. The energy for individual mode in 3.5GHz clock is not shown because it is infinity due to thermal runaway.

4.1.3. Thermal Runaway

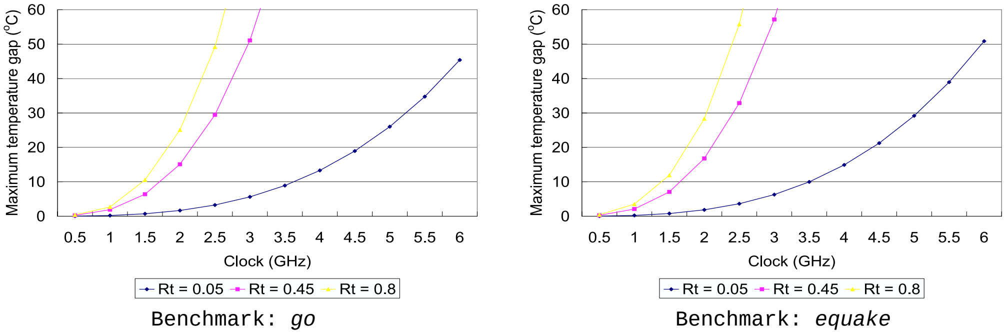
With clock gating, since the power consumption with clock gating in every cycle depends on program behaviors, thermal runaway criteria (1) and (2) discussed in Section 3.5 are no long sufficient conditions for thermal runaway to happen. By simulating 100 million instructions for benchmark *equake* we observe the power consumption and temperature satisfy criteria (1) and (2) for more than 10 times. However, when the program activity becomes low due to dynamic throttling, a processor might be able to escape from thermal runaway.



Benchmark: *go*

Benchmark: *equake*

Figure 8: Maximum temperature under individual mode with different thermal resistance. Realistic clock gating is assumed.



Benchmark: *go*

Benchmark: *equake*

Figure 9: Maximum temperature gap under individual mode with different thermal resistance. Realistic clock gating is assumed.

4.2. Active Cooling

As we can see from previous discussion, the designer's desire to increase system clock can be severely limited by thermal constraints. Better packaging and active cooling techniques can help to remove the thermal resistance, dissipate heat more quickly, and enable faster clocks. [4] discusses a few active cooling techniques such as cooling studs, microbellows cooling and microchannel cooling. [21] introduces a novel active cooling technique by direct water spray-cooling on electronic devices. In this section, we assume realistic clock gating, individual mode and consider three thermal resistance value: (i) $R_t = 0.8^o$C/W for the conventional cooling, (ii) $R_t = 0.05^o$C/W for water spray cooling in [21], and (iii) $R_t = 0.45^o$C/W, a value in between the above two. We call both (ii) and (iii) as *active cooling* and study the impact of active cooling.

4.2.1. Maximum Clock

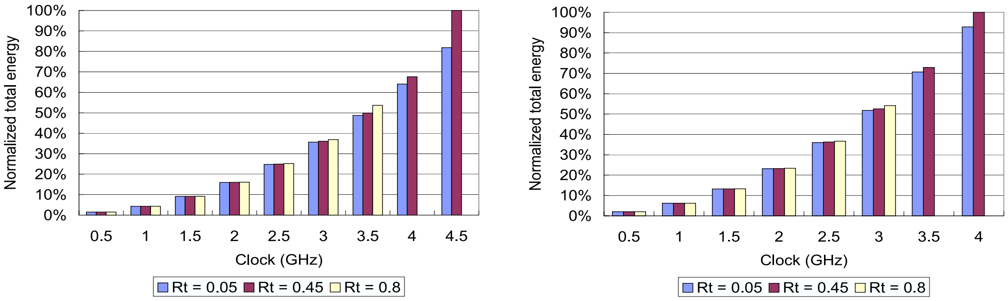
Figure 8 plots maximum temperatures for different clocks with different R_t. Obviously by applying active cooling techniques we can effectively increase the maximum clock while limiting the system temperature well below the thermal constraints. Figure 9 plots the maximum temperature gaps under different cooling techniques and clocks. By combining results in Figure 8 and 9 with the thermal constraints applied in Section 4.1.1, we can increase system clock to up to 5.5GHz by scaling V_{dd} up with $R_t = 0.05^o$C/W. Compared to the 2.25GHz maximum clock without active cooling, the active cooling technique can increase the maximum clock by the factor of $2.44X$.

4.2.2. Total Energy

Figure 10 show the total energy consumption with three different thermal resistances R_t. Clearly the cooling techniques substantially reduce the total energy at the same clock. Compared to R_t of 0.45, R_t of 0.05 reduces the total energy by up to 18%. From Figure 10 we can also see that the energy reduction with active cooling techniques increases as clock increases, which means active cooling techniques is more effective for faster clocks. Note that in Figure 10 a few bars for $R_t = 0.45$ and 0.8^oC/W are missing due to thermal runaway. Traditionally the active cooling techniques such as cooling stubs and microchannel cooling [4] are only applied to mainframes computers. Our result clearly indicates that they can also be effective and may become necessary for microprocessors.

5. CONCLUSIONS AND DISCUSSIONS

Considering cycle accurate simulation, we have presented dynamic and leakage power models with clock, supply voltage and temperature scaling, and developed the coupled thermal and power simulation at the microarchitecture level. With this simulator, we have shown that the leakage energy and total energy can be different by up to $10X$ and 30% for different temperatures, respectively. Hence, microarchitecture level power simulation is hardly accurate without considering temperature dependent leakage model. We



Benchmark: *go*

Benchmark: *equake*

Figure 10: Total energy consumption under individual mode with different thermal resistances. Ideal clock gating is assumed. Note a few bars for clock at 3.5GHz and 4GHz are missing due to thermal runaway.

have studied the system-level thermal runaway problem induced by temperature-dependent leakage power and show that it could be a severe problem in the near feature as the runaway temperature can be much lower than the maximum temperature packages can support. We have studied the microarchitecture level coupled power and thermal management by clock gating and novel active cooling techniques. We show that with thermal constraints, clock gating can increase maximum system clock by up to $1.5X$ and reduce leakage energy by up to 68.5% compared to the cases without clock gating, and active cooling techniques providing smaller thermal resistance can further increase the maximum clock by a factor of $2.44X$. We have developed a coupled power and thermal simulator PTscalar, which integrates temperature and voltage scalable leakage model with accurate thermal calculation considering three dimensional heat transfer. This tool will be available at http://eda.ee.ucla.edu.

Our future work will consider interconnect power and study microarchitecture power/thermal management with simultaneous interconnect power estimation and floorplanning optimization.

References

[1] A. Chandrakasan, W. J. Bowhill, and F. Fox, *Design of High-Performance Microprocessor Circuits*. IEEE Press, 2001.

[2] A. S. Grove, “Changing vectors of moore’s law,” in *Keynote speech, International Electron Devices Meeting*, Dec 2002.

[3] T. Burd, T. Pering, A. Stratakos, and R. Bordersen, “A dynamic voltage-scaled microprocessor system,” in *2000 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb 2000.

[4] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*. Addison-Wesley, 1990.

[5] W.Ye, N.Vijaykrishnan, M.Kandemir, and M.J.Irwin, “The design and use of simplepower: a cycle-accurate energy estimation tool,” in *DAC*, 2000.

[6] D.Brooks, V.Tiwari, and M.Martonosi, “Wattch: A framework for architectural-level power analysis optimization,” in *ISCA*, 2000.

[7] W. Liao, J. M. Basile, and L. He, “Leakage power modeling and reduction with data retention,” in *ICCAD 02*, Nov 2002.

[8] J. Butts and G. Sohi, “A static power model for architects,” in *Proc. of MICRO33*, December 2000.

[9] W. Liao, F. Li, and L. He, “Microarchitecture level power and thermal simulation considering temperature dependent leakage model,” in *ISLPED*, Aug 2003.

[10] A. Dhodapkar, C. Lim, and G. Cai, “Tem^{2}p^{2}est: A thermal enabled multi-model power/performance estimator,” in *Workshop on Power Aware Computer Systems*, Nov 2000.

[11] K. Skadron, M. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, “Temperature-aware microarchitecture,” in *Proceedings of the 30th International Symposium on Computer Architecture*, 2003.

[12] S. Mutoh and *et al*, “1-v power supply high-speed digital circuit technology with multithreshold-voltage cmos,” *IEEE Journal of Solid-state circuits*, vol. 30, pp. 847–854, Aug. 1995.

[13] J. V. D. Vegte., *Feedback Control System, 3rd Edition*. Prentice Hall, 1994.

[14] P. Shivakumar and N. P. Jouppi, “Cacti 3.0: An integrated cache timing, power, and area model,” in *WRL Research Report 2001/2*, 2001.

[15] B. A. Gieseke and et al, “A 600mhz superscalar risc microprocessor with out-of-order execution,” in *Proc. IEEE Int. Solid-State Circuits Conf.*, pp. 176–177, 1997.

[16] J. Stinson and S. Rusu, “A 1.5ghz third generation itanium 2 processor,” in *DAC*, June 2003.

[17] K. Skadron, T. Abdelzaher, and M. Stan, “Control-theoretic techniques and thermal-rc modeling for accurate and localized dynamic thermal management,” in *Proceedings of the Eighth International Symposium on High-Performance Computer Architercture*, 2002.

[18] H. Hanson, M. Hrishikesh, V. Agarwal, S. Keckler, and D. Burger, “Static energy reduction techniques for microprocessor caches,” in *Proceedings of the International Conference on Computer Design*, 2001.

[19] R. Severns, “Safe operating area and thermal design for mospower transistors,” in *Siliconix applications note AN83-10*, Nov 1983.

[20] V. Tiwari, D. Singh, S. Rajgopal, and G. Mehta, “Reducing power in high-performance microprocessors,” in *DAC*, 1998.

[21] M. Shaw, J. Waldrop, S. Chandrasekaran, B. Kagalwala, X. Jing, E. Brown, V. Dhir, and M. Fabbeo, “Enhanced thermal management by direct water spray of high-voltage, high power devices in a three-phase, 18-hp ac motor drive demonstration,” in *Eighth Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, 2002.