Technical Report UCLA Engr. 04-250
University of California at Los Angeles

Temperature-Aware Performance and Power Modeling

Weiping Liao, Lei He and Kevin Lepak†
Electrical Engineering Department
University of California, Los Angeles 90095
†Advanced Micro Devices, Inc., Austin, TX 78741
{wliao, lhe}@ee.ucla.edu, †kevin.lepak@amd.com

Abstract

Power has become the primary design constraint for systems ranging from server computers to handhelds. As semiconductor technology scales down, leakage power becomes significant. Leakage power is exponentially dependent on temperature. Therefore, future design studies call for temperature-aware power modeling and coupled power and thermal management due to the temperature dependence of leakage power. Additionally, temperature-aware performance modeling must be carried out with coupled power and thermal management because circuit delay also depends on temperature. In this paper we study microarchitecture-level temperature-aware power and performance modeling. We present a leakage power model with temperature and voltage scaling. We show that leakage energy and total energy vary 38% and 24% for temperatures between 65^{o}C and 110^{o}C, respectively. We study thermal runaway induced by the interdependence between temperature and leakage power. We also demonstrate that without temperature-aware modeling, underestimation of leakage power may lead to the failure of thermal controls, and overestimation of leakage power may result in excessive performance penalties of up to 5.24%. All of these studies underscore the necessity of temperature-aware power modeling. Furthermore, we study optimal voltage scaling for best performance with dynamic power and thermal management under different packaging options. We show that dynamic power and thermal management allows designs targeting at the common-case thermal scenario among benchmarks and improves performance by 6.59% compared to designs targeting at the worst-case thermal scenario without dynamic power and thermal management. Additionally, the optimal V_{dd} for the best performance may not be the largest V_{dd} allowed by the given packaging platform, and that advanced cooling techniques can improve throughput significantly.

I. Introduction

Power has become a primary design constraint for systems ranging from server computers to handhelds [1]. For VLSI circuits, power consumption includes dynamic power and leakage power. As semiconductor technology keeps scaling down, leakage power grows significantly at the system level because of (1) increase of device leakage current due to the reduction in threshold voltage, channel length, and gate oxide thickness [2], and (2) the increasing number of idle modules in a highly integrated system. For current high-performance design methodologies, the contribution of leakage power increases at each technology generation [3]. The Intel Pentium IV processors running at 3GHz already have an almost equal amount of leakage and dynamic power [4].

Existing microarchitecture-level power simulators [5]–[7] calculate leakage power by assuming a fixed ratio between dynamic and leakage power. This assumption is not accurate because dynamic power and leakage power scale differently as a function of V_{dd} and temperature. Furthermore, leakage power is sensitive to temperature while dynamic power is independent of temperature.

As leakage power becomes important, due to its dependence on temperature, temperature-aware leakage power modeling and dynamic coupled power/thermal management (DPTM) becomes necessary for accurate power estimation and appropriate power/thermal management. [8] presents a high-level leakage power model without temperature scaling. [9] proposes a leakage power model with temperature scaling for 100nm technology with an empirical temperature-dependent term $exp(\frac{-a}{T-b})$ where a and b are empirical constants and T is the temperature. Voltage scaling is not considered for either dynamic or leakage power in [9]. 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.

System performance, on one hand, is affected by V_{dd} scaling because circuit delay and the maximum system clock frequency depend on V_{dd} [11]. Existing performance simulators [12], [13] use instructions per cycle (IPC) to represent performance and do not consider changes in clock frequency possible with different V_{dd}. This approach is no longer valid with V_{dd} scaling, considering power/thermal envelopes. A temperature-dependent circuit delay model has been developed [11] which

This paper is partially supported by NSF CAREER award CCR-0401682, 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.

may improve this deficiency in existing microarchitecture simulators. However, there are no existing microarchitecture-level studies considering the impact of temperature-dependent circuit delay. Furthermore, the impact of leakage power on temperature is not considered during performance evaluation.

In this paper, we present leakage power models with V_{dd} and temperature scaling based on the BSIM4 model for subthreshold and gate leakage current. We develop a coupled thermal and power microarchitecture simulator *PTscalar* [14] which considers the interdependence between leakage and temperature. With PTscalar, we are able to explore various microarchitecture-level leakage power and thermal models as well as coupled power/thermal simulation and management considering the interdependence between leakage power and temperature. We show the dramatic dependence of leakage power on temperature at the microarchitecture level within the temperature range between 65^oC and 110^oC. We also discuss thermal runaway induced by the interdependence of leakage and temperature. We further demonstrate that for dynamic thermal management, underestimating the temperature dependence of leakage leads to violations of temperature constraints and overestimating the temperature dependence of leakage leads to up to 5.24% performance loss due to over-aggressive application of power reduction techniques. These studies underscore the need for temperature-aware power modeling and DPTM.

Furthermore, we present studies on optimal voltage scaling for best performance with DPTM considering voltage scaling. We show that DPTM can increase maximum system throughput by 6.59% compared to designs targeting worst case thermal scenarios without DPTM. Contrary to the widely-accepted belief that scaling to larger V_{dd} leads to improved performance (through gains in clock frequency), we show that the optimal V_{dd} for the best performance may not be the largest V_{dd} allowed by the given package platform. We also study the impact of active cooling techniques providing smaller thermal resistance and show that such techniques can improve maximum system throughput by 15.1% compared to conventional air cooling. All these studies indicate the necessity of temperature-aware performance modeling.

The rest of this paper is organized as follows. In Section II, we develop power and delay models with both voltage and temperature scaling. In Section III, we introduce our thermal model, study microarchitectural-level coupled power and thermal simulation, and discuss the thermal runaway induced by the interdependence between leakage and temperature. In Section IV, we study the importance of coupled power and thermal management. In Section V we study optimal voltage scaling for the best performance with dynamic power and thermal management under different packaging options. We conclude in Section VI.

II. Power and Delay Model with Temperature and Voltage Scaling

A. 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 [15] 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 a useful operation [7].

Dynamic energy is consumed by charging and discharging capacitances. It is independent of temperature, but has a quadratic dependence on supply voltage. In our experiment, dynamic energy in each clock cycle is calculated as CV^2.

In the rest of this subsection, we discuss our leakage power model with V_{dd} and temperature scaling. It has been shown in [16] that leakage power mainly consists of subthreshold and gate leakage power. Each type of leakage exhibits a different temperature and V_{dd} dependence. More importantly, the two manifest themselves at different conditions and the worse-case leakage power is not the simple sum of the worst-case subthreshold and gate leakage power.

1) Subthreshold Leakage Power Models: We study subthreshold leakage power modeling for two types of circuits: one is logic circuits such as functional units, the other is memory-based units such as caches and register files, modeled by SRAM arrays.

For logic circuits, we use the leakage power model proposed in [17]. As shown in (1), for a given circuit, the leakage power can be calculated as the product of the number of gates (N_{gate}) and the average subthreshold leakage current per gate (I_{avg}^{sub}):

$$P_{sub} = N_{gate} \cdot I_{avg}^{sub} \cdot V_{dd} \quad (1)$$

I_{avg}^{sub} can be calculated by computing the average leakage current per gate for the given n circuits using gate-level estimation. Because leakage current depends on different input vectors [18], we apply a genetic algorithm to obtain the input vectors for both maximum and minimum leakage currents, and then calculate I_{avg}^{sub} [17]. Figure 1 shows this I_{avg}^{sub} calculated with respect to the number of circuits. It is easy to see that after the number of circuits exceeds 20, the value of I_{avg}^{sub} becomes stable for both maximum and minimum leakage current when these circuits are designed using the same cell library. Also shown in Figure 1, the average difference between maximum and minimum I_{avg}^{sub} is about 60% of the minimum I_{avg}^{sub}.

A formula similar to (1) has been proposed in [19] which explicitly considers the statistical impacts of transistor stacking. However, no explicit method is proposed in [18], [19] to consider voltage and temperature scaling. We characterize the

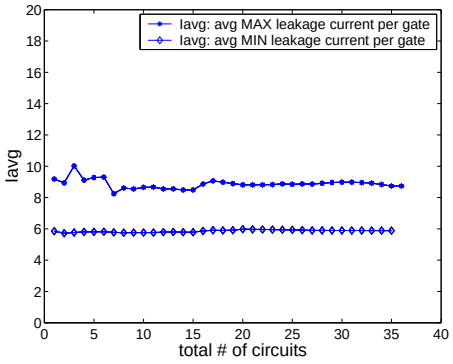


Fig. 1. I_{avg} of random logic.

temperature and voltage scaling of I_{avg}^{sub} based on the following BSIM4 subthreshold leakage current model [3]:

$$I_{sub} = Ae^{\frac{(V_{GS}-V_T-\gamma^{'}V_{SB}+\eta V_{DS})}{nV_{TH}}}\left(1-e^{-\frac{V_{DS}}{V_{TH}}}\right) \quad (2)$$

$$A = \mu_0 C_{ox}\frac{W}{L_{eff}}{V_{TH}}^2 e^{1.8} \quad (3)$$

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 $\frac{kT}{q}$, $\gamma^{'}$ is the linearized body-effect coeffi cient, η is the Drain Induced Barrier Lowering (DIBL) coeffi cient, 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 (2) we can see the temperature scaling for subthreshold leakage current is $T^2 e^{\frac{1}{T}}$, where T is the temperature, and the voltage scaling for leakage current is $e^{V_{dd}}$. Based on these observation, we propose the following formula for I_{avg}^{sub} considering temperature and voltage scaling:

$$I_{avg}^{sub}(T, V_{dd}) = I_s^{sub}(T_0, V_0)\cdot T^2 \cdot e^{\left(\frac{\alpha_{s1}\cdot V_{dd}+\beta_{s1}}{T}\right)} \quad (4)$$

where I_s^{sub} is a constant current at the reference temperature T_0 and voltage V_0. α_{s1} and β_{s1} in (4) are empirical constants decided by circuit designs.

Memory based units such as caches and register files are usually modeled by SRAM arrays. A formula-based subthreshold leakage power model without temperature and voltage scaling has been proposed in [7]. We use a similar model in this work:

$$P_{sub} = P_{ckts}^{sub} + P_{cells}^{sub} \quad (5)$$

$$P_{ckts}^{sub}(T, V_{dd}) = (X\cdot words\cdot word\_size + Y\cdot word\_size) \cdot V_{dd}\cdot T^2\cdot e^{\left(\frac{\alpha_{s2}\cdot V_{dd}+\beta_{s2}}{T}\right)} \quad (6)$$

$$P_{cells}^{sub}(T, V_{dd}) = (Z\cdot words\cdot word\_size) \cdot V_{dd}\cdot T^2\cdot e^{\left(\frac{\alpha_{s3}\cdot V_{dd}+\beta_{s3}}{T}\right)} \quad (7)$$

where P_{cells}^{sub} is the subthreshold leakage power dissipated by SRAM memory cells and proportional to the number of SRAM memory cells. P_{ckts}^{sub} is the power generated by accompanying circuits such as wordline drivers, precharge transistors, etc. P_{cells}^{sub} and P_{ckts}^{sub} essentially have the same format as (1) where $X\cdot words\cdot word\_size + Y\cdot word\_size$ in (6) and $Z\cdot words\cdot word\_size$ in (7) can be viewed as N_{gate}. X, Y, Z, α_{s2-s3} and β_{s2-s3} in (6) and (7) are empirical constants decided by circuit designs.

2) Gate Leakage: In the BSIM4 gate leakage model [20], gate leakage current is calculated as gate direct tunneling current– including tunneling current between gate and substrate (I_{gb}) and current between gate and channel (I_{gc}). The formulas for both I_{gb} and I_{gc} are:

$$I_{gb} = W_{eff}\cdot L_{eff}\cdot X_1\cdot (EXP_{acc} + EXP_{inv}) \quad (8)$$

$$I_{gc} = W_{eff}\cdot L_{eff}\cdot X_2\cdot e^{(-B_3\cdot T_{ox}\cdot(\alpha_3-\beta_3\cdot V_{oxdepinv})\cdot(1+\gamma_3\cdot V_{oxdepinv}))} \quad (9)$$

$$(10)$$

where

$$X_1 = A_1 \cdot T_{oxRatio} \cdot V_{gb} \cdot V_{uax} \quad (11)$$

$$EXP_{acc} = e^{(-B_1 \cdot T_{ox} \cdot (\alpha_1 - \beta_1 \cdot V_{oxacc}) \cdot (1 + \gamma_1 \cdot V_{oxacc}))} \quad (12)$$

$$EXP_{inv} = e^{(-B_2 \cdot T_{ox} \cdot (\alpha_2 - \beta_2 \cdot V_{oxdepinv}) \cdot (1 + \gamma_2 \cdot V_{oxdepinv}))} \quad (13)$$

$$X_2 = A_2 T_{oxRatio} V_{gse} V_{uax} \quad (14)$$

A_1, A_2, B_1, B_2, B_3, α_1, α_2, α_3, β_1, β_2, β_3, γ_1, γ_2 and γ_3 are all empirical constants given by BSIM4 gate leakage model, W_{eff} and L_{eff} are the channel width and length, respectively; $T_{oxRatio}$, V_{uax} are defined in BSIM4 gate leakage model.

From (8) - (9) we can see that in contrast to subthreshold leakage, gate leakage is insensitive to temperature. However, gate leakage is dependent on V_{dd} in the form of $e^{V_{dd}}$.

3) Total Leakage Power: Combining subthreshold leakage and gate leakage, we still keep the format of formulas in our subthreshold leakage power model as in (1) and (5) - (7), but take into account the different scaling feature for subthreshold leakage and gate leakage. With this framework in place, we consider both subthreshold and gate leakage power for logic circuits and memory-based units as shown in (15) - (19)

$$P_{s\_log} = N_{gate} \cdot I_{avg} \cdot V_{dd} \quad (15)$$

$$I_{avg}(T, V_{dd}) = I_s(T_0, V_0) \cdot f_{avg}(T, V_{dd}) \quad (16)$$

$$P_{s\_mem} = P_{ckts} + P_{cells} \quad (17)$$

$$P_{ckts}(T, V_{dd}) = V_{dd} \cdot (X \cdot words \cdot word\_size + Y \cdot word\_size) \cdot f_{ckts}(T, V_{dd}) \quad (18)$$

$$P_{cells}(T, V_{dd}) = V_{dd} \cdot (Z \cdot words \cdot word\_size) \cdot f_{cells}(T, V_{dd}) \quad (19)$$

where P_{slog} is the total leakage power for logic circuits, I_{avg} is the total leakage current per gate, I_s is the I_{avg} at given temperature T_0 and supply voltage V_0, P_{s_mem} is the total leakage power for memory-based units, P_{ckts} and P_{cells} are the total leakage power for SRAM cells and accompanying circuits, respectively, $f_{avg}(T, V_{dd})$, $f_{ckts}(T, V_{dd})$ and $f_{cells}(T, V_{dd})$ are scaling functions to characterize temperature and V_{dd} scaling considering both subthreshold and gate leakage. All three scaling functions f_{avg}, f_{ckts} and f_{cells} have the same format as (20):

$$f(T, V_{dd}) = A \cdot T^2 \cdot e^{\left(\frac{\alpha \cdot V_{dd} + \beta}{T}\right)} + B \cdot e^{(\gamma \cdot V_{dd} + \delta)} \quad (20)$$

where A, B, α, β, γ, and δ are empirical constants for different circuit types, technologies and designs. Notice there is one temperature dependent scaling term for subthreshold leakage current and one temperature independent scaling term for gate leakage current in (20). Each empirical constant is different for different scaling functions. The value of A, B, α, β, γ and δ as well as validation of our power model will be presented in Section II-A.5.

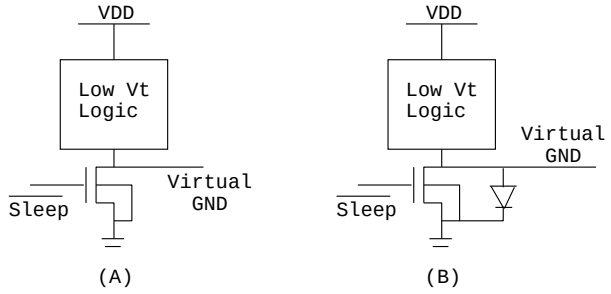


Fig. 2. Power gating techniques: (a) MTCMOS and (b) VRC.

4) Leakage Power Reduction: For both logic circuits and memory based units, we can easily extend our leakage power model for leakage reduction techniques. For example, power gating techniques such as Multi-threshold CMOS (MTCMOS) and Virtual power/ground Rails Clamp (VRC) are used in [7]. Figure 2 shows the circuit schematics for MTCMOS and VRC. In MTCMOS, a sleep transistor is inserted between the circuits and GND. When the sleep transistor is turned off, there is no power supply to the logic circuits. Leakage is reduced substantially but the circuits also cannot hold logic values when the circuit is asleep (sleep transistor is turned off). In VRC, a diode is inserted in parallel with the sleep transistor and maintains the voltage at a reduced level (decreasing leakage) high enough to maintain logic values in the circuits. Note in [7], two sets of diodes and sleep transistors are inserted for both V_{dd} and GND. In this paper, we use only one sleep transistor and diode at GND for leakage reduction and data retention. VRC should be used for memory based units such as caches and register files when data retention is required. In these cases, the leakage power formulas still have the same format as formulas (15) - (20), although the empirical constants may be different.

5) Leakage Model Validation: We obtain the constants in (4) - (7) and (20) empirically by determining the power consumption for different circuit types at multiple temperatures using SPICE simulations and then applying curve fitting. In our experiments we use the input vectors which maximize subthreshold leakage power for each type of circuit. For I_{avg}, we use the average leakage current for three types of circuits with different bit-width: adder (4-bit, 16-bit and 32-bit), shifter (8-bit, 16-bit and 32-bit), and multiplier (4-bit, 5-bit and 6-bit). For SRAM arrays, we use different combination of row and column. Different temperature are chosen during curve fitting and verification. Tables I and II summarize the empirical constants for the 65nm technology used. Table III compares our high-level leakage power estimation for logic circuits and SRAM arrays with SPICE simulations in 65nm technology. The overall difference between our formulas and SPICE simulation is less than 7%, indicating the formulas for high-level leakage power estimation achieve reasonable accuracy.

X	Y	Z
0.20306	-0.25289	1.0

TABLE I

EMPIRICAL CONSTANTS IN (4) - (7) FOR 65NM TECHNOLOGY. THESE CONSTANTS ARE THE SAME FOR CASES WITH AND WITHOUT POWER GATING.

		A	B	α	β	γ	δ
Without power gating	I_{avg}	1.1432e-12	1.0126e-14	466.4029	-1224.74083	6.28153	6.9094
	P_{ckts}	1.1432e-12	1.3906e-13	466.4029	-1224.74083	6.6943	4.46958
	P_{cells}	2e-12	2.0581e-13	930.1355	-1712.5319	6.6943	4.46958
With power gating	I_{avg}	1.1432e-14	1.0126e-16	466.4029	-1224.74083	6.28153	6.9094
	P_{ckts}	1.1432e-14	1.3906e-15	466.4029	-1224.74083	6.6943	4.46958
	P_{cells}	8e-14	8.2324e-15	930.1355	-1712.5319	6.6943	4.46958

TABLE II

COEFFICIENTS FOR THE SCALING FUNCTION IN (20) FOR DIFFERENT CIRCUITS IN 65NM TECHNOLOGY.

Circuit	Temperature (^{o}C)	V_{dd}	I_{avg} (uA) formula	I_{avg} (uA) SPICE	abs. err. %
logic circuits for adder, multiplier, and shifter	100	0.95	23.44	23.56	0.49
	100	1.05	29.56	29.63	0.23
	80	0.95	19.44	19.54	0.56
	80	1.05	25.14	25.21	0.27
	60	0.95	16.00	16.11	0.65
	60	1.05	21.33	21.39	0.31
Circuit	Temperature (^{o}C)	V_{dd}	P_{so} (uW) formula	P_{so} (uW) SPICE	abs. err. %
SRAM 128x32	100	0.95	181.91	188.18	3.54
	100	1.05	262.71	271.42	3.31
SRAM 512x32	100	0.95	729.11	753.38	3.33
	100	1.05	1052.8	1086.5	3.21

TABLE III

COMPARISON BETWEEN OUR FORMULA AND SPICE SIMULATION. I_{avg} IS FOR LOGIC CIRCUITS. P_{so} AND P_{io} ARE STANDBY POWER WITHOUT VRC AND INACTIVE POWER WITH VRC FOR SRAM POWER MODEL, RESPECTIVELY. THE SRAM ARRAYS ARE REPRESENTED AS "ROW NUMBER" X "COLUMN NUMBER". THE UNITS FOR I_{avg} AND SRAM POWER ARE UA AND UW, RESPECTIVELY.

B. Delay Model with Voltage and Temperature Scaling

For VLSI circuits, the relationship between circuit delay and supply voltage V_{dd} is $delay \propto V_{dd}/(V_{dd} - V_t)^{\xi}$, where V_t is the threshold voltage and α is an empirical constant. Temperature also affects circuit delay by affecting carrier mobility and threshold voltage [21]. The delay model with temperature and voltage scaling is shown in (21):

$$delay \propto \frac{V_{dd}T^{\mu}}{(V_{dd} - V_t)^{\xi}} \tag{21}$$

where μ and ξ are empirical constants for different technology. We obtain $\mu = 1.19$ and $\xi = 1.2$ for 65nm technology by SPICE simulation and curve fitting empirically. Table IV compares our delay model with SPICE simulation for circuit delay of an inverter with load of FO-4. The absolute error is within 8%.

T(°C)	V_{dd}	SPICE	Formula	Error (%)
60	0.9	31.17	33.53	7.57
60	1.1	28.42	30.08	5.85
80	0.9	38.31	35.94	6.17
80	1.1	30.65	32.24	5.19
100	0.9	40.27	38.38	4.71
100	1.1	32.94	34.42	4.51

TABLE IV

COMPARISON BETWEEN OUR FORMULA AND SPICE SIMULATION FOR CIRCUIT DELAY OF AN INVERTER WITH FO-4 LOAD.

By assuming the maximum clock frequency $f_{max} = 1/delay$, the appropriate supply voltage to achieve f_{max} can be decided by (22):

$$f_{max} \propto \frac{(V_{dd} - V_t)^{1.2}}{V_{dd}T^{1.19}} \tag{22}$$

III. COUPLED POWER AND THERMAL SIMULATION

A. Thermal Model

According to the well-know duality between heat transfer and electrical phenomena [22], temperature can be modeled by equivalent RC thermal circuits, where two parameters: thermal resistance R_t and thermal capacitance C_t are used to characterized thermal behavior. We develop our thermal calculation based on the equivalent RC thermal circuits presented in the HotSpot toolset [23]. As shown in Figure 3 from [23], the equivalent RC thermal circuit consists of three layers: heatsink, heat spreader and chip die. The chip die is partitioned into functional blocks according to microarchitecture functionality. The heat spreader is divided into five blocks: one for the area right under the die and four trapezoids for the periphery not covered by the die. Similar to heat spreader, the heat sink is divided into five blocks. For each block, there are two types of RC pairs to capture both vertical and horizontal heat transfer characteristics: The vertical RC pairs connect the center of each block down to the center of the next layer, to model the vertical heat transfer between layers. The lateral RC pairs connect the center of each block to the center of the cross-section between this block and adjacent blocks in the same layer. The lateral RC pairs characterize the horizontal heat transfer between blocks within each layer. For each RC pair, the thermal resistance R_t is proportional to the thickness of the block and inversely proportional to the cross-sectional area across which the heat is being transferred. In contrast, the thermal capacitance C_t is directly proportional to both thickness and area. Provided the average power within a time period, the transient temperature is calculated by solving the differential equations for the RC circuit with a fourth-order Runge-Kutta method [23].

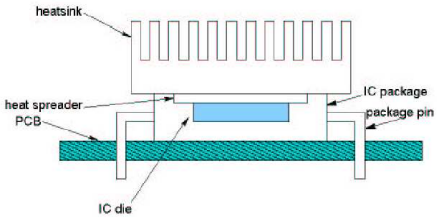
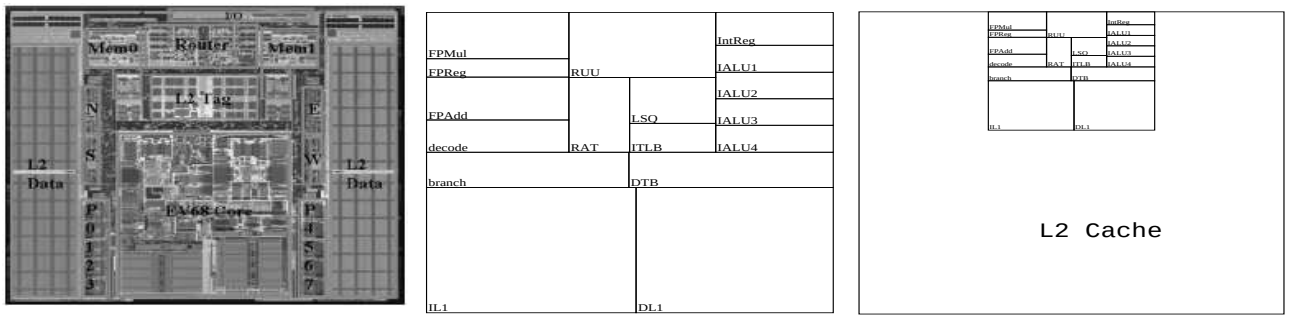


Fig. 3. Side view of IC package.

The thermal time constants ($\tau = R_t * C_t$) for blocks are usually in the order of milliseconds, and millions of times larger than the order of clock cycle. Therefore, it is not necessary to update temperature and power for every clock cycle. During simulation, we update temperature and power after each time step t_s. An appropriate value of t_s can greatly reduce simulation overhead while maintaining accurate temperature calculation. Details of selecting t_s are given in Subsection III-C.

B. Experiment Settings

We choose 65nm technology in our experiments. Although our power model is applicable to any instruction set architecture and microarchitecture, we study out of order superscalar architectures in this paper. We integrate our power model and temperature calculation into the SimpleScalar 3.0b toolset [12] with Alpha ISA and name the new coupled power and thermal simulator *PTscalar*. Table V presents the microarchitectural processor configuration. We partition the microprocessor for power/thermal modeling by major functional components. As shown in Table VI, there are two types of components: memory-based units and logic circuits. When calculating the power of memory-based units, we first partition the component into pieces of SRAM arrays with the CACTI 3.0 toolset [24], then apply our formulas for power consumption to each SRAM array. The



(a) DEC Alpha 21364 die photo (b) Floorplanning without L2 Cache (c) Full-Chip Floorplanning

Fig. 4. The die photo of the DEC Alpha 21364 and the floorplanning similar to the Alpha 21364 used in our experiments. Note our floorplanning is slightly different from that in [23] due to different microarchitectural functional block partitioning, for example, [23] distinguishes integer and floating issue queues, while we use the unified register update unit (RUU) as defined in SimpleScalar.

total component power consumption is the sum of power for all SRAM sub-arrays. Among logic circuits, for integer ALUs and FPUs, we take the area and gate count in the design of Alpha 21264 processor [25] and scale from 350nm technology down to 65nm technology. For all other logic circuits, we estimate gate count according to the designs in [26], and then apply formula (1) to calculate the leakage power for logic circuits. Table VII summarizes the power consumption for all components in our system. Similar to other Microarchitecture level power simulators [6], [23], we do not consider the control logic as one component.

We choose the fborplan similar to DEC Alpha 21364 [23] as shown in Figure 4. The thermal model extracts the thermal resistance R_t and thermal capacitance C_t according to this fborplan. To consider appropriate supply voltage scaling for varying clock frequencies, we assume that $V_{dd} = 0.9V$ obtains f_{max} = 5GHz as specified by the ITRS [27]. According to Equation (22), the f_{max} for different V_{dd} and maximum temperature T allowed for the circuits in our experiments are shown in Table VIII.

Parameter	Value
Processor Core	
RUU size	64 instructions
LSQ size	32 instructions
Fetch Queue size	8 instructions
Fetch width	4 instructions/cycle
Decode width	4 instructions/cycle
Issue width	4 instructions/cycle
Commit width	4 instructions/cycle
Functional Units	3 integer addition, 1 integer multiplication/division, 1 FP addition, 1 FP multiplication/division
Branch Predictor	Combined, Bimodal 4K table 2-Level 1K table, 10-bit history 4K chooser
BTB	512 entries, 4-way
Memory Hierarchy	
L1 instruction-cache	64KB, 4-way (LRU) 32B blocks, 1-cycle latency
L1 data-cache	64KB, 4-way (LRU) 32B blocks, 1-cycle latency
L2	Unified, 4MB, 8-way (LRU) 128B blocks, 12-cycle latency
TLB	128 entry, fully associative 30-cycle miss latency
Main memory	255-cycle latency

TABLE V
SIMULATED MICROPROCESSOR CONFIGURATION.

Component type	Microarchitecture structure
Memory-based units	Caches, register files, TLB, branch predictor, register update unit (RUU), load/store queue (LSQ), rename table (RAT)
Logic circuits	Integer and floating-point functional units

TABLE VI

COMPONENTS IN OUR EXPERIMENTS.

Component	P_a	P_s	P_i
BTB	639.41	87.39	18.93
L1 Instruction Cache	770.16	222.55	8.90
L1 Data Cache	732.09	222.60	8.90
Unified L2 Cache	20580.31	13123.87	524.95
Integer Register File	56.20	1.57	0.06
Floating-point Register File	56.20	1.57	0.06
RUU	66.49	3.48	0.15
LSQ	112.40	3.14	0.19
One Decode Unit	30.38	1.60	0.06
One Integer ALU	554.60	11.46	0.11
One Floating-point Unit	1122.45	21.57	0.22

TABLE VII

POWER IN MW FOR ALL COMPONENTS FOR 65NM TECHNOLOGY, THE SUPPLY VOLTAGE IS 0.9V AND THE CLOCK FREQUENCY IS 5GHZ. THE DECODE, INTEGER ALU AND FPU ARE ONLY ONE UNIT AMONG TOTAL FOUR, FOUR, AND TWO UNITS. THE TEMPERATURE IS 100^oC.

	V_{dd}			
T(^{o}C)	0.9	1.1	1.3	1.5
40	6.36	7.09	7.67	8.15
50	6.12	6.83	7.39	7.85
60	5.91	6.58	7.12	7.57
70	5.70	6.36	6.88	7.31
80	5.51	6.14	6.65	7.06
90	5.33	5.94	6.43	6.83
100	5.16	5.75	6.22	6.61
110	5.00	5.57	6.03	6.41

TABLE VIII

f_{max} IN GHZ AFTER APPROPRIATE VOLTAGE AND TEMPERATURE SCALING. V_{dd} IS THE SUPPLY VOLTAGE AND T IS THE MAXIMUM TEMPERATURE ALLOWED FOR THE CIRCUITS.

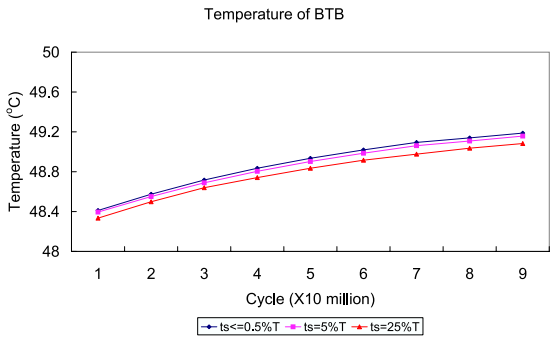


Fig. 5. Temperature curve of the BTB for different time step t_s. The time constant is 2ms. The clock frequency is 5GHz and V_{dd} is 0.9V. 0.5%, 5% and 25% of thermal time constant corresponds to 50 thousand, 500 thousand and 2.5 million cycles, respectively. The benchmark is *gcc*.

C. Speedup of Coupled Power and Thermal Simulation

We update temperatures after each time step t_s, and then update the power value with respect to the newly calculated 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 temperature calculation. Figure 5 plots the transient temperature of the BTB calculated using different t_s shown as the percentages of the thermal time constant, where 0.5% of the thermal time constant is equal to 50000 clock cycles for a 5GHz clock. When $t_s \leq 50000$ cycles (i.e. 0.5% of thermal constants), 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 constants. Furthermore, Table IX compares the running time normalized with respect to that without temperature calculation. By setting t_s to 50000 cycles, we not only introduce negligible error on temperature calculation, but also reduce run time by more than 23 times compared to $t_s = 1$ cycle, and achieve virtually the same computation efficiency as power simulation without temperature calculation. Since the clock frequencies are always faster than 5GHz in our experiments, 0.5% of thermal constants are always more than 50000 cycles. Since $ts = 50000$ cycles leads to negligible error on temperature calculation, we use this value for t_s throughout the rest of the paper.

t_s (cycle)	N.T.	1	100	1000	10000	50000
Running time	1.0	23.94	5.52	1.44	1.04	1.004

TABLE IX

NORMALIZED RUN TIME FOR VARYING PERIODS OF TEMPERATURE UPDATE. THE $N.T.$ MEANS WE DO NOT HAVE TO UPDATE TEMPERATURE AND POWER DURING THE WHOLE SIMULATION.

D. Temperature Dependent Leakage Power

Figure 6 shows the experimental results for total leakage power consumption at three different temperatures. From Figure 6 we can see that by changing the temperature from 65^oC to 110^oC, the total leakage energy can be changed by 38%. Figure 6 clearly shows that any study regarding leakage power is not accurate if the temperature dependence of leakage power is not considered. 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.

As an engineering approximation, one might consider assuming a fixed temperature appropriate for the processor and package, and then use leakage values at this reference temperature instead of directly considering the temperature variation of leakage power. There are many caveats to this approach. First, with dynamic throttling such as clock gating, it is difficult to decide the appropriate reference temperature a priori without cycle-accurate simulation with a temperature dependent leakage model since power and temperature are interrelated. Second, because different benchmarks will exhibit different thermal behavior, and unequal ratios between static and dynamic power, reference temperatures with this simple model are benchmark-dependent. Even with this careful consideration, since leakage power is strongly dependent on temperature, minor temperature variations can lead to large estimation errors in power and thermal simulation with potentially hazardous consequences (See Sections III-E, IV-A.1 and IV-A.2). Therefore, coupled power and thermal management is necessary. We have shown through this work that coupled power and thermal simulation is indeed highly practical for existing simulation tools.

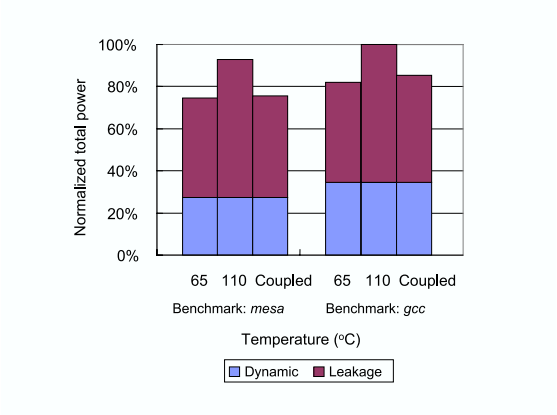


Fig. 6. Total power consumption with the breakdown of dynamic and leakage portions. The clock is 6.03GHz and V_{dd} is 1.3V. Clock gating is applied and removes 75% of dynamic power every idle cycle. The label *Coupled* represents coupled power and thermal simulation considering temperature-dependent leakage power.

E. Thermal Runaway

The thermal runaway problem in MOSFETs due to the positive feedback loop between on-resistance, temperature and power is well known [28]. In this section we will present another thermal runaway problem due to the interaction between leakage

power and temperature. As component temperature increases, its 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 increase to infinity, leading to thermal runaway and catastrophic thermal failure. Assuming no throttling[1], for transient temperature T_0 and T_1 at consecutive times t_0 and t_1 and corresponding power $P(T_0)$ and $P(T_1)$, we define the following two criteria as sufficient and necessary conditions[2] for thermal runaway:

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. This criteria is equivalent to $\frac{d^2T}{dt^2} > 0$, where T is temperature and t is time.

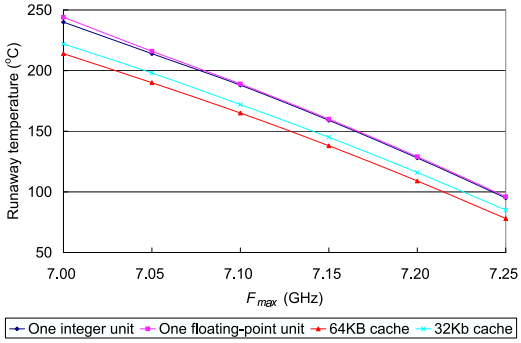


Fig. 7. Runaway temperatures.

We define the lowest temperature to meet both criteria 1 and 2 as the *runaway temperature*. As long as the transient temperature reaches the runaway temperature, thermal runaway cannot be avoided and the transient temperature will increase indefi nitely if no appropriate thermal management is applied. We calculate the runaway temperature according to criteria 1 and 2 for different f_{max} with appropriate voltage scaling. We choose the maximum temperature constraint 110^{o}C as it is the maximum temperature supported by current design technology. Figure 7 shows the runaway temperatures for clock from 7.0GHz to 7.25GHz. As clock increases, the runaway temperature decreases since the difference between power $P(T_1)$ and $P(T_0)$ increases. For clock at 7.25GHz, the runaway temperatures for integer units can be lower than the maximum temperature constraint 110^{o}C. Therefore, thermal runaway may become a severe problem in the near future as clock rates continue to increase. Special thermal management schemes are required to combat this problem.

F. Clock Gating

Due to its exponential dependence on temperature, leakage energy can be greatly affected by mechanisms which significantly reduce system power and temperature. Clock gating [29] reduces dynamic power by turning off the clock signal for idle components. It is shown in [9] that clock gating can indirectly affect leakage energy consumption by changing the temperatures of system components. In the rest of our experiments, we assume clock gating to all components and that clock gating can reduce dynamic power by 75%.

IV. Coupled Power and Thermal Management

In this section, we study coupled power and thermal management using fetch toggling with the proportional-integral (PI) feed-back controller presented in [23]. In fetch toggling, when the temperature is higher than a given threshold, the instruction fetch rate is decreased to reduce activity of processor components. A PI controller has two preset parameters: the *gain* and the temperature threshold to trigger thermal management (*setpoint*). 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, branch predictor and decode units with clock gating. Additionally, fetch toggling can reduce the number of instructions in the out-of-order window, thereby affecting activity of other units as well. We name the coupled power and thermal management with PI feedback controller as *Dynamic Power/Thermal Management* (DPTM).

[1]Any mechanism that slows down the processor's execution can be categorized as throttling.
[2]They are only necessary conditions when there is throttling.

A. Importance of Temperature Dependent Leakage Power Model

Although leakage power has exponential dependence on temperature, studies in the literature tend to choose a fixed leakage power model corresponding to a representative temperature point for low implementation and simulation overhead. In this section we show that in DPTM, ignoring the temperature dependence of leakage power may lead to either control failure or excessive performance penalty.

We implement both our new temperature dependent leakage power model (*accurate* model) and the fixed leakage power model (*simple* model) in DPTM. We choose the maximum temperature constraint 110^{o}C, V_{dd} 1.55V and f_{max} 6.5GHz according to (22). Since the component temperatures in our experiments in this section are usually in the range between 65^{o}C and 110^{o}C, we choose two temperature points 65^{o}C and 110^{o}C as reference temperatures for leakage power calculation in the simple model. Because leakage power at 65^{o}C and 110^{o}C are the lower and upper bounds of the leakage power in our accurate model, respectively, we further name them as *underestimated* model and *overestimated* model.

In this section, we design the PI controller using the following algorithm: first we select a few candidate of setpoints and gains, then we perform simulation for all the combinations of these candidates and finally we select the combination of setpoint and gain achieving the highest IPC (instructions per cycle) and no thermal constraint violations as the PI controller.

1) Control Failure by Underestimation of Leakage Power: We choose three candidates for setpoint: 109^{o}C, 109.4^{o}C and 109.8^{o}C, and three candidates for gain: 0.5, 1.0, 1.5. With the underestimated model, we design PI controller according to our algorithm choosing a setpoint of 109.8^{o}C and gain = 0.5. With this PI controller in DPTM, Figure 8 plots the transient temperature curves simulated by both the underestimated model and the accurate model. For the underestimated model, it appears that the feed-back thermal control effectively limits the maximum on-chip temperature. However, this appearance is erroneous due to underestimated leakage power. With accurate leakage model, the PI controller can no longer prevent thermal constraint violations. Clearly if we design the PI controllers according to underestimated leakage model, our PI controllers may fail to prevent the maximum on-chip temperature from exceeding the maximum temperature constraint. This example illustrates the importance of accurate leakage modeling in the study of dynamic thermal management.

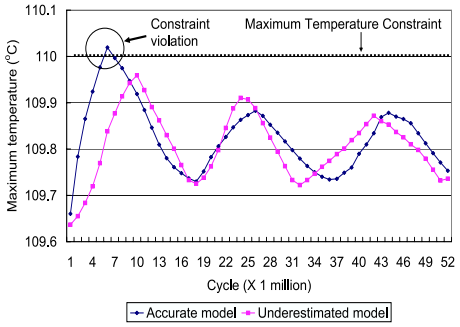


Fig. 8. Transient temperature curves obtained by accurate model and underestimated model. The benchmark is *gcc*.

2) Performance Penalty by Overestimation of Leakage Power: With the overestimated model, we choose three candidates for setpoint: 100^{o}C, 102.5^{o}C and 105^{o}C, and three candidates for gain: 1.0, 3.0 and 5.0. By choosing smaller setpoints and the larger gain, the PI controller can enforce throttling while the temperature is still low and become more sensitive to the increase of temperature, both of which help to eliminate temperature constraint violations. According to our algorithm, we obtain the PI controller with setpoint 102.5^{o}C and gain 1.0 for overestimated model. However, if we design the PI controller with accurate leakage model, we obtain another PI controller with setpoint 105^{o}C and gain 1.0. Table X shows the IPC results obtained under accurate model with PI controller designed by both accurate model and overestimated model. From Table X we can see that overestimated model leads to lower IPC due to excessive performance penalty by unnecessary throttling. The IPC obtained by a controller based on the overestimated model is up to 5.24% lower than that based on the accurate model. This result further indicates the necessity of coupled power and thermal modeling for thermal management.

V. Optimal Voltage Scaling with Dynamic Power and Thermal Management

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 maximized. System performance is defined as throughput in BIPS (Billion Instruction Per Second) in (23):

$$Throughput = \frac{IPC \times clock\_frequency}{10^9} \tag{23}$$

where $clock\_frequency$ is the processor clock frequency.

Benchmark	PI controller designed by accurate model	PI controller designed by overestimated model	Performance penalty by overestimation
art	1.71	1.64	4.09%
bzip2	1.16	1.14	1.36%
equake	1.27	1.27	0%
gcc	1.40	1.33	5.24%
gzip	1.83	1.80	1.85%
mesa	0.74	0.74	0%

TABLE X
IPC COMPARISON.

A. System Performance with Air Cooling

In this subsection we assume air cooling techniques with heatsink thermal resistance 0.8^{o}C/W. As in Section IV-A, we choose the PI controller and fetch toggling mechanism for DPTM. We examine a number of values for V_{dd} and maximum temperature constraints for best performance. Because it is not realistic to design a specific PI controller for each set of V_{dd} and maximum temperature constraints according to our previous algorithm in Section IV-A, we choose setpoint as 5^{o}C lower than the maximum temperature constraints and fix the gain as 1.0.

We first study the performance impact of DPTM. The maximum temperature constraint is no more than 110^{o}C, and the V_{dd} is between 0.9V to 1.4V. Without DPTM, the corresponding clock frequencies to guarantee temperature less than 110^{o}C for all benchmarks are between 5.0GHz and 6.41GHz. On the other hand, with DPTM, the solution space can be increased through the added flexibility of DPTM, and the choices of clock frequency can be between 5.0GHz and 6.86GHz. Table XI compares the maximum throughput between designs targeting at worst-case thermal scenario among the benchmark set without DPTM and those targeting at common-case thermal scenario with DPTM. It is easy to see that by allowing higher BIPS for common-case benchmarks and reducing BIPS for worst-case benchmarks to avoid temperature violation, DPTM helps to improve maximum throughput measured over the benchmark set by 6.59%.

Figure 9 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 therefore, higher throughput. However, higher V_{dd} leads to larger power consumption and higher temperature, which results in more throttling and larger IPC loss under DPTM. Therefore, higher V_{dd} does not always guarantee better throughput. Figure 9 shows that by increasing V_{dd} from 1.2V to 1.4V, throughput can actually be reduced by up to 57% (for cases with maximum temperature constraint 80^{o}C). 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.

	Design for worst-case benchmarks without DPTM	Design for common-case benchmarks with DPTM to avoid the worst-case
Performance (BIPS)	8.5	9.06 (+ 6.59%)

TABLE XI
PERFORMANCE COMPARISON. RESULTS ARE THE THE AVERAGE OVER SIX SPEC 2000 BENCHMARKS: *art*, *bzip2*, *equake*, *gcc*, *gzip* AND *mesa*.

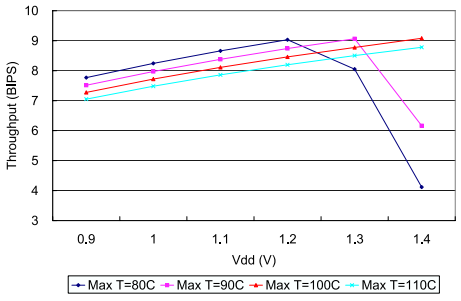


Fig. 9. Average throughput with DPTM under different V_{dd} and maximum temperature constraints for six SPEC 2000 benchmarks: *art*, *bzip2*, *equake*, *gcc*, *gzip* and *mesa*.

B. 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 [30] and direct water spray-cooling on electronic devices [31]. In this subsection, we consider two representative heatsink thermal resistances: (1) $R_t = 0.8^o$C/W for conventional air cooling, and (ii) $R_t = 0.067^o$C/W for water spray-cooling in [31], which we call *active cooling*, and study the impact of active cooling.

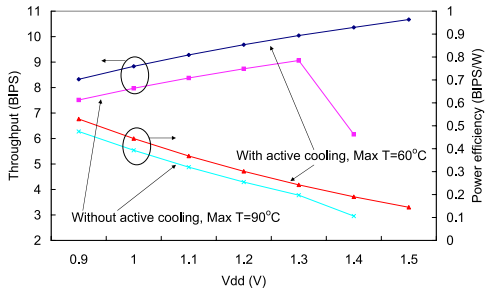


Fig. 10. Average throughput and power efficiency under different V_{dd}, maximum temperature constraints and different cooling conditions for six SPEC 2000 benchmarks: *art*, *bzip2*, *equake*, *gcc*, *gzip* and *mesa*.

With active cooling, the maximum on-chip temperature is greatly reduced. As a consequence, 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 10 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 15.1%, but also slows down the decay of power efficiency as V_{dd} increases and improves maximum power efficiency by 11.45%. Traditionally the research of active cooling techniques are only limited to mainframe computers or power electronics. Our results in Figure 10 clearly indicate that they can also be effective and may become necessary for microprocessors.

VI. Conclusions and Discussions

Considering cycle accurate simulation, we have presented dynamic and leakage power models with clock, supply voltage and temperature scaling, and developed a microarchitecture-level coupled thermal and power simulator PTscalar. With this simulator, we have shown that the leakage energy and total energy can be different by up to 38% and 24% for different temperatures, respectively. Hence, microarchitecture level power simulation is hardly accurate without considering a temperature dependent leakage model. We have studied the system-level thermal runaway problem induced by leakage and temperature interdependence and show that it may be a severe problem in the near feature. We have further demonstrated that for dynamic thermal management, underestimating temperature dependency of leakage violates temperature constraints and overestimating temperature dependency of leakage leads to up to 5.24% performance loss. Finally, we have studied the optimal voltage scaling for best performance with dynamic power and thermal management under different packaging options. We have shown that dynamic power and thermal management allows designs targeting at common-case thermal scenario among benchmark sets and enables dynamic throttling to avoid the worst-case thermal scenario. This can achieve 6.59% performance improvement compared to designs only targeting at the worst-case. Additionally, the optimal V_{dd} for the best performance may not be the largest V_{dd} allowed by the given packaging platform, and that advanced cooling techniques can improve throughput signifi cantly.

References

[1] S. Borkar, "Design challenges of technology scaling," *IEEE Micro*, pp. 23–29, Jul.-Aug. 1999.
[2] Y. Taur and T. H. Ning, *Fundamentals of Modern VLSI Devices*. Cambridge University Press, 1998.
[3] A. Chandrakasan, W. J. Bowhill, and F. Fox, *Design of High-Performance Microprocessor Circuits*. IEEE Press, 2001.
[4] A. S. Grove, "Changing vectors of moore's law," in *Keynote speech, International Electron Devices Meeting*, Dec 2002.
[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] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*. Addison-Wesley, 1993.
[12] D. Burger and T. Austin, *The simplescalar tool set version 2.0*. University of Wisconsin-Madison, 1997.
[13] "Impact architectural framework," in *http://www.crhc.uiuc.edu/Impact*.
[14] "Ptscalar," in *http://eda.ee.ucla.edu/PTscalar/*, 2004.
[15] 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.

[16] D. Lee, W. Kwong, D. Blaauw, and D. Sylvester, “Analysis and minimization techniques for total leakage considering gate oxide leakage,” in *DAC*, June 2003.
[17] F. Li, L. He, J. Basile, R. J. Patel, and H. Ramamurthy, “High level area and current estimation,” in *Proceedings of the Asia South Pacific Design Automation Conference*, Jan 2004.
[18] Z. Cheng, M. Johnson, L. Wei, and K. Roy, “Estimation of standby leakage power in cmos circuits considering accurate modeling of transistor stacks,” in *Proceedings of the International Symposium on Low Power Electronics and Design*, Aug 1998.
[19] W. Jiang, V. Tiwari, E. de la Iglesia, and A. Sinha, “Topological analysis for leakage prediction on digital circuits,” in *Proceedings of the 7th Asia and South Pacific Design Automation Conference, joint with the 15th International Conference on VLSI Design*, 2002.
[20] U. B. D. Group, *BSIM 4 MOSFET model*, July 2002.
[21] R. Cobbold, “Temperature effects on mos transistors,” *Electronic Letters*, vol. 2, pp. 190–192, 1966.
[22] J. V. D. Vegte., *Feedback Control System, 3rd Edition*. Prentice Hall, 1994.
[23] 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.
[24] P. Shivakumar and N. P. Jouppi, “Cacti 3.0: An integrated cache timing, power, and area model,” in *WRL Research Report 2001/2*, 2001.
[25] 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.
[26] N. S. Palacharla and J.E.Smith, “Quantifying the complexity of superscalar processors.,” in *Technology Report CSTR-96-1328, Univ. of Wisconsin-Madison*, November 1996.
[27] *The International Technology Roadmap for Semiconductors*. http://public.itrs.net/, 2001.
[28] R. Severns, “Safe operating area and thermal design for mospower transistors,” in *Siliconix applications note AN83-10*, Nov 1983.
[29] V. Tiwari, D. Singh, S. Rajgopal, and G. Mehta, “Reducing power in high-performance microprocessors,” in *DAC*, 1998.
[30] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*. Addison-Wesley, 1990.
[31] 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.