Energy-Optimal Dynamic Thermal Management: Computation and Cooling Power Co-Optimization

Donghwa Shin, *Student Member, IEEE*, Sung Woo Chung, *Member, IEEE*, Eui-Young Chung, *Member, IEEE*, and Naehyuck Chang, *Senior Member, IEEE*

***Abstract*—Conventional dynamic thermal management (DTM) assumes that the thermal resistance of a heat-sink is a given constant determined at design time. However, the thermal resistance of a common forced-convection heat sink is inversely proportional to the flow rate of the air or coolant at the expense of the cooling power consumption. The die temperature of the silicon devices strongly affects its leakage power consumption and reliability, and it can be changed by adjusting the thermal resistance of the cooling devices. Different from conventional DTM which aims to avoid the thermal emergency, our proposed DTM regards the thermal resistance of a forced-convection heat sink as a control variable, and minimize the total power consumption both for computation and cooling. We control the cooling power consumption together with the microprocessor clock frequency and supply voltage, and track the energy-optimal die temperature. Consequently, we reduce a significant amount of the temperature-dependent leakage power consumption of the microprocessor while spending a bit higher cooling power than conventional DTM, and eventually consume less total power. Experimental results show the proposed DTM saves up to 8.2% of the total energy compared with a baseline DTM approach. Our proposed DTM also enhances the Failures in Time (FIT) up to 80% in terms of the electromigration lifetime reliability.**

***Index Terms*—Dynamic thermal management (DTM), heat sink, liquid cooling, reliability, temperature-dependent leakage power.**

I. Introduction

TRANSISTOR scaling has been resulting in explosive power density increase. So far, elaborated device-level low-power techniques mitigate the thermal issues of large-scale semiconductor devices, and high-level dynamic thermal management (DTM) also plays an important role. Such DTM techniques basically reduce the source of the heat, but efficient heat dissipation is equally important to avoid thermal emergency. Consequently, high-performance systems are often equipped with active cooling mechanisms that reduce the thermal resistance of the heat sinks by the use of active devices such as fans and pumps. These active cooling methods include air-cooled heat sink, liquid cooling, thermoelectric cooling, and so on. A large-scale server cluster such as a datacenter, even uses air conditioners for better cooling efficiency.

A common characteristic of active cooling devices is that their thermal resistance is controllable; more cooling power, less thermal resistance. The thermal resistance of a forced-convection heat sink is inversely proportional to the rotational speed of the fan and the pump. For instance, the thermal resistance of an air-cooled forced-convection heat sink is determined by the fan rotational speed. A typical cooling fan is driven by a brushless DC motor with a feedback speed controller, such that the fan speed can be controlled by software. A higher speed produces a lower thermal resistance at the expense of power consumption. Typical operating systems are able to control the cooling fan speed by setting an external register.

If we do not care about the cooling power consumption, operating the active cooling devices at the maximum speed results in the least thermal resistance and thus the most efficient cooling. However, the power consumption is one of the most important metrics to be optimized for large-scale servers as well as for battery-operated systems. It goes without saying that the ultimate DTM should minimize the total power consumption, both computation and cooling power, avoiding the thermal emergency and keeping required performance of the system. We can achieve the goal if we jointly optimize the computation and cooling power, which is possible when we regard the thermal resistance of an active cooling device as a control variable together with common DTM control knobs. Unfortunately, none of previous DTM regards the thermal resistance of an active cooling device controllable. Instead, it is considered as a design-time given constant. This paper is the first work that addresses: i) thermal resistance of an active cooling device as a control variable for DTM and ii) a total energy-optimal DTM that jointly optimizes the computation and cooling power.

A lower thermal resistance of a cooling device reduces the thermal equilibrium temperature. This again reduces the temperature-dependent leakage power because the subthreshold leakage power is proportional to the die temperature. Thus, more cooling power results in less computation power even if the die temperature does not affect the dynamic power. Although commercial systems are capable of controlling the

Manuscript received November 25, 2009; revised May 07, 2010; accepted May 26, 2010. Date of publication June 28, 2010; date of current version August 06, 2010. This work was supported in part by the Brain Korea 21 Project, in part by the 'National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) under Grant 2009-0060054 and Grant 2010-0017680, in part by the Korea Research Foundation Grant funded by the Korean Government (KRF-2009-013-D00099), and in part by the IC Design Education Center (IDEC). The ICT at Seoul National University provides research facilities for this study. This paper was presented in part at the 2009 International Conference on Computer-Aided Design (ICCAD), San Jose, CA. Paper no. TII-09-11-0341.

D. Shin and N. Chang are with the Department of Electrical Engineering and Computer Science, Seoul National University, Seoul 151-742, Korea (e-mail: dhshin@elpl.snu.ac.kr; naehyuck@elpl.snu.ac.kr).

S. W. Chung is with the Division of Computer and Communication Engineering, Korea University, Seoul 136-713, Korea (e-mail: swchung@korea.ac.kr).

E.-Y. Chung is with the School of Electrical and Electronic Engineering, Yonsei University, Seoul 120-749, Korea (e-mail: eychung@yonsei.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2010.2052059

cooling device operating speed, their cooling device control strategies do not consider the total power consumption. In this paper, we introduce a systematic design framework to achieve both avoidance of the thermal emergency and less total power consumption.

The major contribution of this paper is summarized as follows: We introduce a new DTM technique in such a way that the power consumption of a microprocessor and its cooling device are jointly minimized. We formulate and solve an optimization problem where the temperature-dependent leakage power consumption of the microprocessor and the power consumption of the cooling device form a convex function. Experimental results show that the proposed DTM saves up to 8.2% of the total energy compared with a baseline DTM approach. Our proposed DTM also enhances the reliability up to 80% in terms of Failure in Time (FIT) for the electromigration.

II. Related Work

Most DTM techniques reduce the power consumption and in turn the die temperature. Common techniques include dynamic frequency scaling (DFS), dynamic voltage and frequency scaling (DVFS), decode throttling, speculation control, and I-cache toggling at the expense of the throughput [2]. Therefore, one typical DTM problem setup is maximization of the throughput while meeting the thermal constraints. It turns out that reaching the maximum allowable die temperature as soon as possible and keeping track of that temperature results in the maximum throughput [3]. Another approach addresses the problem of optimizing the performance of a set of periodic tasks using the discrete voltage/frequency states available on actual processors [4]. Both approaches optimize the throughput in a multiple-task environment. Another important problem setup is real-time task scheduling under thermal constraints. A new task scheduling considers the effect of the microprocessor temperature [5]. A recent work introduces practical aspects that should be considered in a task scheduling [6].

The quality of DTM is largely dependent on the quality of the power and thermal model. As transistor scaling progresses, DTM considers leakage power especially when DVFS controls the power consumption [7]. One of the most significant leakage power sources, subthreshold leakage, exponentially increases by temperature. Therefore, it is important to consider the temperature dependent leakage power in DTM for modern microprocessors [8]. The temperature-dependent leakage power results in a circular dependency between the power and temperature, and thus a proper DTM method should resolve the complexity of the solution method [9]. A practical DTM is not feasible when only the microprocessor is accounted for, and thus system-level modeling should be considered especially for web farm and server systems [10], [11].

Generally, semiconductor dies do not heat up evenly, but generate thermal hotspots. Identification of hotspots is important because a part of malfunction in a semiconductor device may incur complete failure of the entire system. Thermal sensors effectively measure the die temperature online, but their proper allocations significantly affect the correct hotspot identification [12]. The density of thermal sensors increases not only the quality of the hotspot identification, but also the cost and design complexity of a semiconductor device. A soft thermal sensing help reduce the number of thermal sensors [13].

A predictive DTM may enhance the performance of multimedia applications because information about the future workload helps increase the performance of DTM as in dynamic power management [14]. Practical DTM methods should consider discrete DVFS because most commercial microprocessors allow discrete frequency and voltage levels [4]. Modern DTM techniques accommodate multiprocessor environments because thermal constraints discourage the use of a high clock frequency single core microprocessor [15]. Thermal simulation such as HotSpot [16] is widely used to validate the thermal behavior taking advantage of the simulation over measurement.

None of the work mentioned above deals with the optimality of the total energy consumption of a system. As explained earlier, existing DTM schemes primarily focus on avoiding thermal emergency while considering the thermal resistance of the cooling devices as a constant.

III. Power and Thermal Models

A. Temperature-Aware Microprocessor Power Model

We characterize the power consumption model of a microprocessor with the following parameters: the effective switching capacitance C_e, the supply voltage $V_{\rm dd}$, the operating clock frequency f, and the technology constant K_n. The power consumption of a CPU is expressed as

$$P_{\rm cpu} = P_d + P_s + P_0 \tag{1}$$

where P_d, P_s and P_0, respectively, are the dynamic, static, and always-on power consumption. The dynamic power consumption is given by

$$P_d = \frac{1}{2} C_e {V_{\rm dd}}^2 f. \tag{2}$$

We consider the two major leakage power sources in the static power model, which are subthreshold leakage and gate leakage power. The static power consumption is dependent on the die temperature T_d, which can be expressed as follows:

$$P_s(T_d) = V_{\rm dd}\left(K_1 T_d^2 e^{\left(\frac{K_2 V_{\rm dd}+K_3}{T_d}\right)} + K_4 e^{(K_5 V_{\rm dd}+K_6)}\right). \tag{3}$$

We expand the right-hand side of (3) as a Taylor series and retain its linear terms

$$\begin{aligned} P_s(T_d) &= \sum_{n=0}^{\infty}\left(\frac{1}{n!}\right)\frac{d^n P_s(T_r)}{{dT_d}^n}(T_d - T_r)^n \\ &\approx P_s(T_r) + \frac{dP_s(T_r)}{dT_d}(T_d - T_r), \end{aligned} \tag{4}$$

where T_r is a reference temperature, which is usually an average value within the typical die temperature range. This linearization is subject to an error of less than 5% within an ordinary temperature range of 25°C to 120°C [17].

While this model provides relatively accurate power estimation, it does not reflect local hotspots. We exclude spatial variations in temperature from discussion without loss of generality

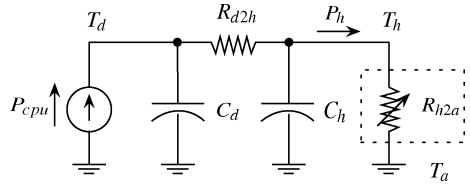


Fig. 1. RC-thermal circuit model with a variable thermal resistance.

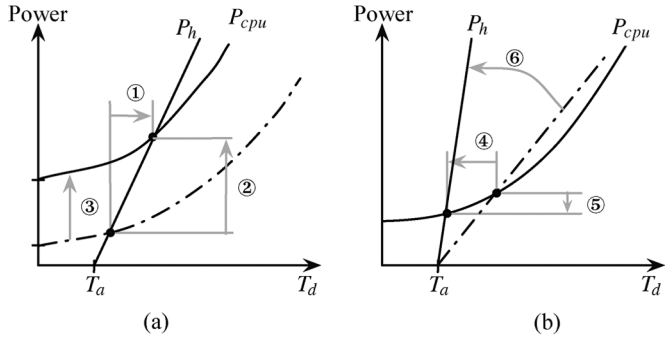


Fig. 2. The effect of a variable thermal resistance, achieved by controlling the cooling device, on the thermal equilibrium die temperature. (a) Variation in T_d ① and $P_{\rm cpu}$ ② with dynamic power ③. (b) Variation in T_d ④ and $P_{\rm cpu}$ ⑤ with thermal resistance R_{h2a} ⑥.

to avoid from divergence of the paper context. However, our future work will include the spatial variations.

B. RC-Thermal Model

We devise a new RC-thermal model adopting a variable resistor that replaces the fixed resistor in [16] and [18], as shown in Fig. 1. This makes the problem statement and the solution completely different from previous DTM problems. The thermal model includes characteristic parameters such that C_d is the thermal capacitance of the die; R_{d2h} is the thermal resistance from the die to the package combined with its heat sink; C_h is the thermal capacitance of the package combined with its heat sink; P_h is the heat dissipated by the heat sink; T_h is the temperature of the heat sink; and T_a is the ambient temperature. The thermal circuit in Fig. 1 can be rewritten as

$$P_{\rm cpu} = C_d \frac{dT_d}{dt} + \frac{T_d - T_h}{R_{d2h}} \tag{5}$$

$$\frac{T_d - T_h}{R_{d2h}} = C_h \frac{dT_h}{dt} + \frac{T_h - T_a}{R_{h2a}}. \tag{6}$$

Fig. 2(a) shows a conventional thermal management schemes in which the thermal resistance of the heat-sink, with or without a cooling device, is constant. In this case, the thermal equilibrium die temperature can be obtained as follows:

$$P_{\rm cpu} = P_h = \frac{T_d - T_a}{R_{d2h} + R_{h2a}}. \tag{7}$$

If P_d increases, the lower dot-dashed curve in Fig. 2(a) goes up [marked ③], and both T_d and $P_{\rm cpu}$ at thermal equilibrium increase [marked ① and ②]. The amount of the increase in $P_{\rm cpu}$ is larger than that in P_d because the temperature-dependent leakage power increases more as temperature increases.

A change in the thermal resistance results in different slope of the P_h line [⑥ in Fig. 2(b)]. If the thermal resistance $R_{d2h} + R_{h2a}$ were zero, which cannot of course happen in reality, T_d would be T_a. A lower thermal resistance of the cooling device produce a lower equilibrium T_d. A lower thermal resistance reduces T_d [④ in Fig. 2(b)] and thus the leakage power [⑤ of Fig. 2(b)]. If the amount of reduction in the temperature-dependent leakage power is larger than the additional power used by the cooling devices, the total power consumption is reduced.

We formulate a total power model which combines temperature-dependent leakage power and thermal resistance in a thermal equilibrium

$$\begin{aligned} P_{\rm total} &= P_{\rm cpu} + P_{\rm cooling} \\ &= P_d + \left(\alpha \cdot \frac{(R_{h2a} + R_{d2h})(\beta + P_d + P_0) + T_a}{1 - \alpha(R_{h2a} + R_{d2h})} + \beta\right) \\ &\quad + P_0 + P_{\rm cooling} \end{aligned} \tag{8}$$

where the temperature-dependent leakage power is linearized such that $\alpha = (dP_s(T_r))/(dT_d)$ and $\beta = P_s(T_r) - T_r(dP_s(T_r))/(dT_d)$. We derive the power consumption of the cooling devices $P_{\rm cooling}$ from following cooling device models.

C. Forced-Convection Air-Cooled Heat Sink Model

One of the most common types of cooling devices is a forced-convection air-cooled heat sink. It consists of a heat sink made of a low thermal resistance material such as copper and aluminum and a cooling fan that circulates ambient air through the heat sink.

The temperature of the microprocessor is determined by the amount of heat transferred from the device to the ambient air. The thermal resistance varies with the amount of convection, which is determined by the speed of the cooling fan. A typical forced-convection heat sink has an encoder that reads the speed of the fan. The microprocessor is equipped with temperature sensors, and controls the fan speed using pulse width modulation (PWM).

We model the thermal resistance R_{h2a} of a heat sink as a function of the power of the fan $P_{\rm fan}$. Among the analytical models of a forced-convection air-cooled heat sink [19]–[21], we use the thermal exchanger [19] to evaluate the effect of flow rate through the channel precisely. The thermal resistance model is given by

$$R_{h2a} = \left(mc_{p,a}\left(1 - e^{\left(-\frac{h A_e}{m c_{p,a}}\right)}\right)\right)^{-1} \tag{9}$$

where $m = (\rho_a \upsilon_a)/(\Delta)$ is the mass flow rate; υ_a is the velocity of the air; ρ_a is the density of the air; Δ is the cross-sectional area of the air channel; $c_{p,a}$ is the specific heat of the air; A_e is the effective area of the heat sink; $h_a = (k_a Nu)/(D_{h,a})$ is the heat transfer coefficient of the heat sink, in which the Nusselt number Nu can in turn be approximated as a function of the Reynolds number; $R_e = (\upsilon_a D_h)/(\nu_a)$ is the Reynolds number; D_h, a is the hydraulic diameter of the air channel; ν_a is the viscosity of the air; and k_a is the thermal conductivity of the heat sink material. We represent the thermal resistance as a

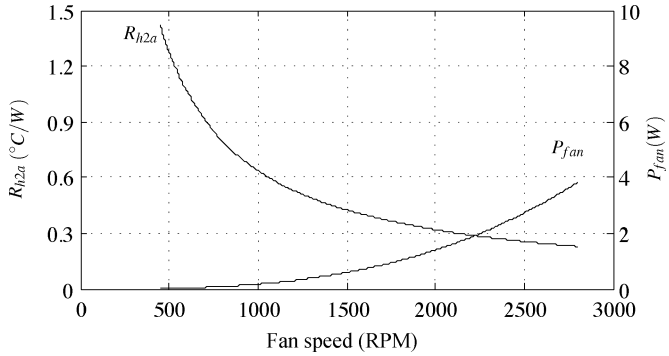


Fig. 3. The thermal resistance and power consumption of a forced-convection heat sink composed of a parallel plate copper fin heat sink (70 mm × 70 mm × 50 mm) and a 70 mm cooling fan.

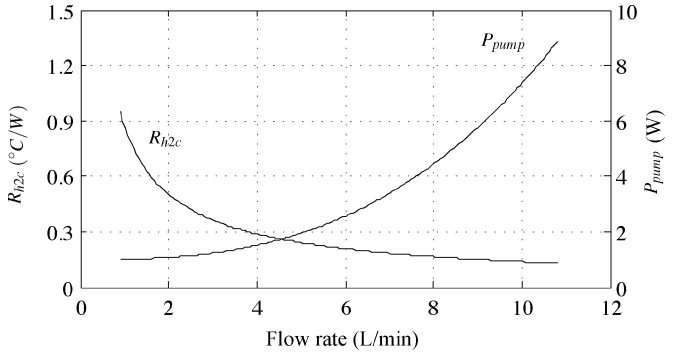


Fig. 4. The thermal resistance and power consumption of a liquid cooling heat sink composed of a Aluminum cooling block (40 mm × 40 mm × 10 mm) with 3/8" liquid channel and a liquid pump.

function of v_a with physical coefficients h_1, h_2, h_3, and h_4 as follows:

$$R_{h2a} = \left(h_1 \, v_a \left(1 - e^{\left(\frac{h_2 v_a^{h_3} + h_4}{h_1 v_a} \right)} \right) \right)^{-1}. \qquad (10)$$

The flow rate and the velocity of the air are determined by the speed of the fan for a fixed air channel. The energy consumed for rotating the fan is the same as the energy required to deliver the air by the conservation of energy

$$P_{\text{fan}} \propto {v_a}^3. \qquad (11)$$

The efficiency of air delivery is determined by factors which include the shape of the channel and friction. The thermal resistance of a forced-convection heat sink can be expressed as a function of its power consumption by substituting (11) into (10). So, we manipulate the thermal resistance denoted in (10) by changing P_{fan} rather than v_a.

Fig. 3 visualizes the tradeoff relationship between the thermal resistance of the heat sink and cooling fan power consumption. We measure the power consumption of the fan and the flow rate, and derive the associated thermal resistance values by (10) with the physical parameters of a CNPS-9700NT forced-convection air-cooled heat sink from Zalman Tech [22].

D. Liquid Cooler Model

Liquid cooling device is also a heat exchanger. It is able to transfer the large amount of heat, but it generally consumes much more power than that of air-cooled heat sink. A lower thermal resistance develops a higher cooling efficiency, but development of a lower thermal resistance requires more cooling power consumption. The liquid coolers is able to achieve lower thermal resistance with sufficient power, but it shows higher thermal resistance than the air-cooled heat-sink when the same amount of power is supplied, as shown in Figs. 3 and 4.

To develop an analytical model for the liquid cooling devices, we use an water-cooled single-phase rectangular channel heat sink model [23], [24]. Common liquid cooler for the computing system can be modeled as a single-phase heat exchanger which does not change the phase of the coolant, i.e., the liquid (water)

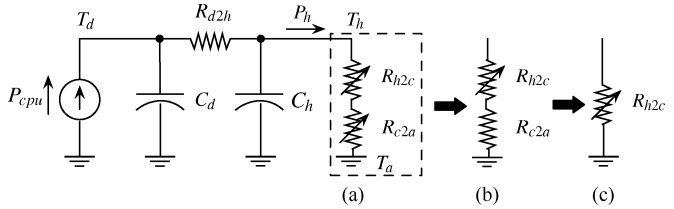


Fig. 5. Thermal model of liquid cooler as a variable thermal resistance.

does not vaporize in the cooling system. We model the thermal resistance as a function of the pump power consumption.

A liquid cooling system consist of a cooling block for the cooling of a microprocessor and an external radiator for the cooling of coolant. Therefore, a generalized thermal resistance model of the heat sink consists of two variable resistors, as shown in Fig. 5(a). However, we consider the thermal resistance of the radiator as a constant keeping the external radiator fan speed fixed while we change the liquid pump flow rate [Fig. 5(b)]. As a result, the thermal resistance has been simplified to one variable thermal resistance [Fig. 5(c)]. To keep focus on the joint optimization framework, we use a single variable resistance model for the liquid cooling system. The major contribution of this paper is a joint optimization of the cooling power and computation power with active cooling system that can change the thermal resistance by changing the cooling power. The proposed framework for the cooling and computation joint optimization can be applied as long as the thermal resistance is a function of the power consumption.

The thermal resistance between the heat sink to the coolant is expressed as follows:

$$R_{h2c} = \frac{1}{h_c A_e}. \qquad (12)$$

The heat transfer coefficient h_c is given by

$$h_c = j c_{p,c} \rho_c v_c \left(\frac{c_{p,c} \nu_c}{k_c} \right)^{-\frac{2}{3}} \qquad (13)$$

where $c_{p,c}$ is the specific heat of the coolant and v_c, ρ_c, and ν_c are the velocity, density, and viscosity of coolant, respectively. The Colburn factor j is a function of Reynolds number. The form of the function is changed according to the range of Reynolds number.

The power consumption of the pump is proportional to the pressure drop ΔP and the volumetric flow rate of the coolant m_c

$$P_{\text{pump}} \propto \Delta P \cdot m_c \tag{14}$$

where the frictional pressure drop ΔP can be calculated as

$$\Delta P = f\left(\frac{L}{D_h, c}\right)\left(\frac{\rho_c {v_c}^2}{2}\right) \tag{15}$$

where L is the channel length, and D_h, c is the hydraulic diameter of the coolant channel.

Fig. 4 visualizes the tradeoff relationship between the thermal resistance of the cooling block and the pump power consumption. We measure the power consumption of pump and the flow rate, and derive the associated thermal resistance values by (12) with the physical parameters of a Bigwater770 liquid-cooled heat sink from Thermaltake [25].

E. Verification of the Power and Thermal Models

We verify the accuracy of our approximated analytical models with HotSpot simulation. The exponential temperature-dependent leakage power model [8] is integrated to the Hotspot. We also modify HotSpot [16] so that the thermal resistance of the cooling device can be changed during runtime to accommodate an adjustable cooling devices, and integrated it with Wattch [26], [27]. We conduct simulation of the Intel Xeon Quadcore E7330 microprocessor [28] running the gcc benchmark from SPEC2000 [29] because it is known that gcc exhibits a large spatial variation in temperature [30]. The E7330 processor has enough power and performance characteristics to show the effect of the temperature. We use a performance monitoring unit on the microprocessor to obtain activity counts for each functional block [30]. Wattch estimates the power consumption of a microprocessor using these activity counts, and HotSpot generates a temperature profile using the power consumption values from Wattch. We also consider the circular dependencies between HotSpot and Wattch have circular dependencies.

We extract the parameters of the power model using HotSpot and Wattch, and calculated the total power consumption at four different die temperatures to verify our model. The result shown in Table I compares the HotSpot simulation results with evaluation of (8). The error between simulation result and analytic prediction is less than 5%, which confirms appropriate parameter extraction.

TABLE I
COMPARISON OF (8) WITH THE HOTSPOT SIMULATION

V_{dd} (V)	f (GHz)	P_{total} (W) from HotSpot			
		40°C	60°C	80°C	100°C
1.35	3.00	N/A	104.67	109.81	115.17
1.30	2.67	78.37	82.46	86.55	N/A
1.25	2.25	59.31	62.38	65.51	N/A
1.20	1.87	46.24	48.74	N/A	N/A
V_{dd} (V)	f (GHz)	P_{total} (W) from (8)			
		40°C	60°C	80°C	100°C
1.35	3.00	98.40	103.48	108.56	113.63
1.30	2.67	78.10	82.10	86.10	90.10
1.25	2.25	60.80	63.87	66.94	70.01
1.20	1.87	44.44	46.62	48.80	50.97

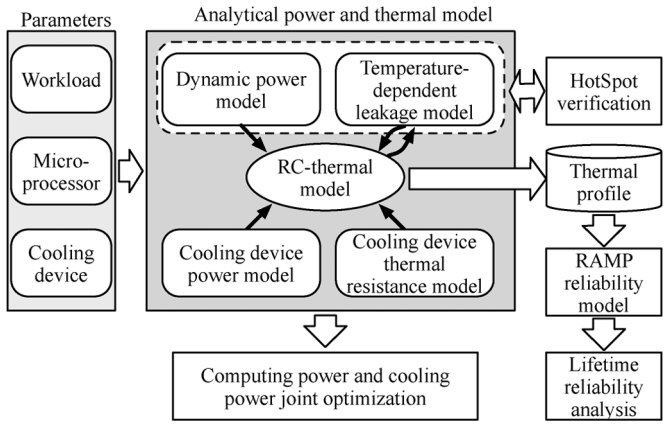


Fig. 6. The joint optimization framework for the computing power and cooling power.

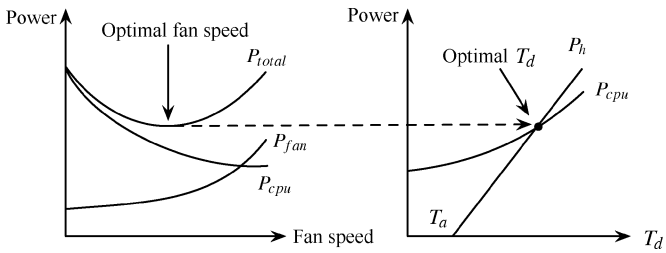


Fig. 7. Optimal cooling fan speed in thermal equilibrium.

IV. JOINT COMPUTATION AND COOLING POWER OPTIMIZATION

We address a joint optimization of cooling power and microprocessor power. Fig. 6 shows the framework for the computing power and cooling power joint optimization. We introduce an example with the air-cooled forced-convection heat sink in Fig. 3 only. However, the same framework can accommodate the liquid cooling case.

A. Power Optimal Cooling Power for Continuous Execution

First, we consider a continuous task execution with a fixed supply voltage and frequency. We derive the power optimal fan speed. As illustrated in Fig. 7, both P_{cpu} and P_{cooling} are convex functions of the fan speed. Thus, the total power consumption P_{total} is convex, so the optimal speed can be found where

$$\frac{dP_{\text{total}}}{dP_{\text{cooling}}} = 0 \tag{16}$$

or at the boundary of cooling device operating range.

For example, we can obtain the optimal fan speed for different supply voltage and operating frequency using analytical model as shown in Table II. We use the parameter of an Intel

TABLE II
VALUES OF OPTIMAL P_{fan} TO MINIMIZE TOTAL POWER CONSUMPTION

V_{dd} (V)	f (GHz)	Optimal fan speed (RPM)	P_{total} (W) (Optimal)
1.35	3.00	2699	109.01
1.30	2.67	2499	86.07
1.25	2.25	2310	65.79
1.20	1.87	2181	47.94

Xeon Quadcore E7330 microprocessor assembled with a parallel-plate finned copper heat sink (70 mm × 70 mm × 50 mm) and a 70 mm cooling fan.

B. General (Non-Real-Time) Workload With Continuous DVFS

We derive the total-energy-optimal fan speed together with the supply voltage and frequency scaling for a given non-real-time workload. The non-real-time workload in this section is defined with a parameter W_b which is the number of cycles needed to execute the workload. The supply voltage V_{dd} can be determined by the Alpha Power Law, which determines the minimum possible voltage that guarantees a stable operation of the microprocessor at frequency f. We consider this minimum voltage when selecting the supply voltage among supported voltage setups specified in the datasheet. For a supported clock frequency f, the scaling factor s can be obtained by dividing workload execution time t_e at frequency f by the execution time $t_{\max}$ at the maximum frequency $f_{\max}$, which is given by

$$s = \frac{t_e}{t_{\max}} = \frac{W_b/f}{W_b/f_{\max}} = \frac{f_{\max}}{f}. \tag{17}$$

Unlike previous DTM techniques, we have two control parameters that affect T_d: P_{cooling} and s. Therefore, we have multiple feasible solutions which achieve the desired value of T_d. Among these feasible solutions, we find the energy-optimal pair of (P_{cooling}, s). Either element of this pair (P_{cooling}, s) may be located outside the feasible range; the optimal solution may be found at the boundary of the feasible range of each variables, or both.

The total-energy-optimal fan speed control problem for a non-real-time workload is defined as follows:

Problem 1: Find the energy-optimal cooling power and scaling factor for a given non-real-time workload: minimize the energy consumption including the cooling power per cycle by controlling the fan speed together with supply voltage and frequency scaling, which is given by

$$E = \frac{P_{\text{cpu}} + P_{\text{cooling}}}{f}. \tag{18}$$

■

As long as s is continuous, the optimal solution pair is determined as follows:

$$(P_{\text{cooling}}, s) \in \left\{ (P_{\text{cooling}}, s) \,\middle|\, \frac{\partial E}{\partial P_{\text{cooling}}} = 0, \quad \frac{\partial E}{\partial s} = 0 \right\}. \tag{19}$$

Fig. 8 illustrates the solution space of Problem 1. It shows how the total energy consumption of the microprocessor and cooling fan varies with fan power and scaling factor for a given workload. As an example, we demonstrate P_{cooling} and s for an Intel Xeon Quadcore E7330 microprocessor assembled with a parallel-plate finned copper heat sink (70 mm × 70 mm × 50 mm) and a 70 mm cooling fan. We set the operating range of the processor from 1.87 GHz at 1.2 V to 3.00 GHz at 1.35 V. The optimal solution which minimizes the total energy consumption is found within the feasible range of the control variables. In this example, the total energy consumption varies up to 5% with different cooling power. It means that the cooling power control is able to achieve energy saving without loss of performance.

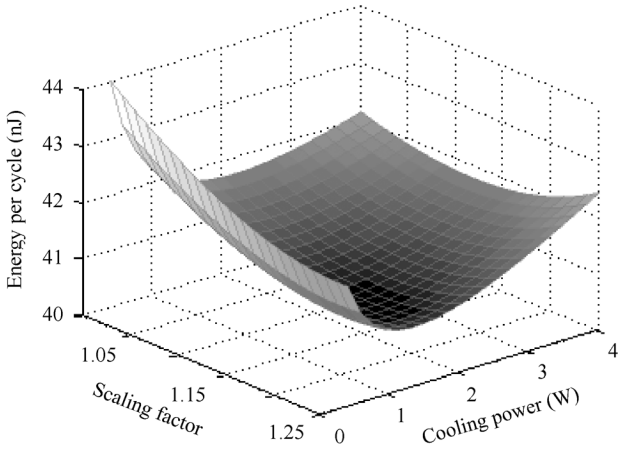


Fig. 8. Energy consumption for a non-real-time workload on Intel E7330 microprocessor with voltage and frequency scaling and a cooling fan with the fan speed control of air-cooled heat sink.

In most cases, only discrete s is available. The scaling factor s has several discrete levels in $S = (s_1, \ldots, s_n)$ since the frequency is discrete in practice. We obtain the optimal feasible solution by selecting the pair (P_{cooling}, s) which minimizes the total energy consumption from

$$\left\{ (P_{\text{cooling}}, s_i) \,\middle|\, \frac{\partial E}{\partial P_{\text{cooling}}} = 0, s_i \in S \right\}. \tag{20}$$

C. Stationary Periodic Real-Time Workload for Continuous DVFS

This subsection derives the total-energy-optimal cooling solution for periodic tasks: a task $\mathcal{T}$ is a pair (W_p, D), where W_p is the workload and D is the deadline. The task $\mathcal{T}$ is a stationary periodic real-time task, where W_p and D are constant and known in advance. We consider the typical dynamics of cooling fans such that the response time is in the order of dozens of milliseconds or longer. Therefore, it is reasonable to assume that we do not change the fan speed within a task execution period.

We start by considering the effect of the initial and final temperatures for a sequence of scheduled tasks. The final temperature of a task instance becomes the initial temperature of the next one. Conventional DTM often assumes that the initial temperature is an arbitrary value between the ambient temperature and the thermal emergency temperature, and the final temperature is forced to be lower than the initial temperature [4]. With a fixed thermal resistance, the energy-optimal initial temperature of a

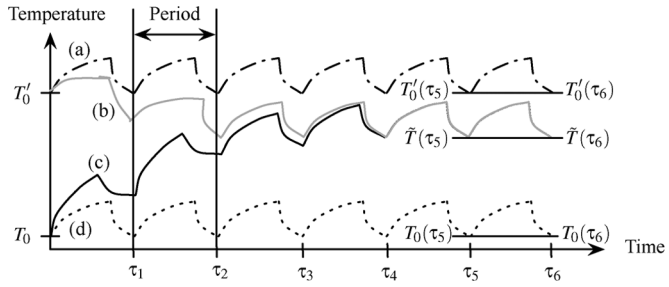


Fig. 9. Effect of the initial temperature on a periodic task.

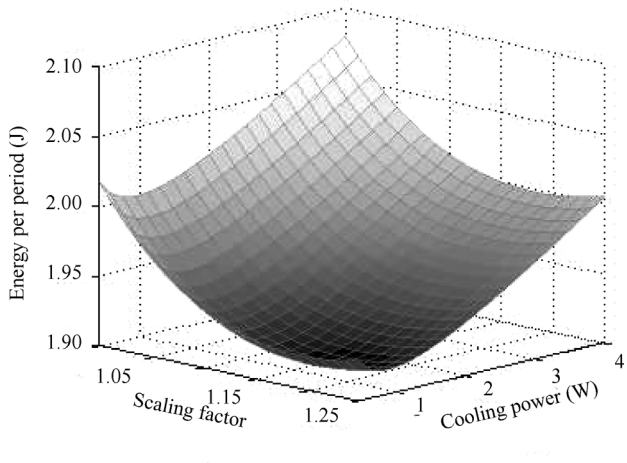


Fig. 10. Energy consumption during one execution period for a periodic real-time task running on the E7330 microprocessor using DVFS and different fixed cooling fan speed of the air-cooled heat sink during the task period.

period for a stationary periodic task converges to T_s, as shown in Fig. 9(b), which is the steady-state temperature for a given task. If we do not care about the energy-optimal steady-state temperature for a given periodic task, it might result in overheating a processor, as shown in Fig. 9(a), or overcooling, as shown in Fig. 9(c). The balance between the cooling power and temperature-dependent leakage power is determined by the efficiency of the cooling device. To sum up, it is crucial to find the energy-optimal T_s when the thermal resistance is fixed.

The thermal resistance of the forced convection cooling device strongly depends on the velocity of the air. It is very hard to control the fan or pump speed precisely within a task period due to the physical limitations. We restrict that the fan or pump speed cannot change within a task period, which is a practical assumption. Thus, we also assume that it is hard to control the thermal resistance of the cooling device in a task period and the value is fixed during a task execution.

Assumption 1: Slow fan dynamics: the fan is too sluggish to update its speed promptly at each period. ■

The total-energy-optimal fan speed control problem for a periodic task is defined as follows:

Problem 2: Find the energy-optimal steady-state pair (P_{cooling}, s): for given T_a and $\mathcal{T}$, determine the energy-optimal values of P_{cooling}, s, and T_s under the thermal constraint. ■

The energy-optimal steady-state temperature at the end of a period is obviously the same as its energy-optimal temperature at the start of the period: this follows from the definition of the steady state, which is $T_s = \tilde{T}(\tau_i) = \tilde{T}(\tau_{i+1})$. We determine T_s for each pair (P_{cooling}, s) to solve Problem 2. From (5) and (6), we represent T_d as a function of $t, T_0, (dT_0)/(dt)$, and P_{cpu}. The steady-state temperature T_s is found by solving the following equations:

$$T_{\text{peak}} = T_d\left(t_e, T_s, \frac{dT_d(\tau_i)}{dt}, P_{\text{cpu}}\right)$$
$$T_s = T_d\left(D - t_e, T_{\text{peak}}, \frac{dT_d(\tau_i + t_e)}{dt}, P_{\text{cpu}}\right) \quad (21)$$

with the constraint that the peak temperature T_{peak} has to be lower than the thermal emergency temperature T_{em}. The execution time t_e is equal to $W_p \cdot s/f_{\max}$. What (21) shows is the following: 1) temperature change and the peak temperature within a task period, 2) the condition for the starting temperature and the end temperature to be the same. The die temperature after time t is derived by (1)–(8) once the initial temperature, gradient of the initial temperature, and CPU power consumption are given. We derive a closed form solution by linearizing the temperature dependent leakage. The first term in (21) shows the peak temperature when the start and end temperatures are the same. The microprocessor reaches its peak temperature when the task execution ends and the idle mode begins, so the value is given as the first term in (21). The end temperature of a task period can be calculated by setting the initial temperature as the peak temperature, and the time as the remaining time in a task period after task execution finishes for function $T_d(t, T_0, (dT_0)/(dt), P_{\text{cpu}})$. This end temperature value should be the same as the start temperature of the task period, as shown in the second term in (21).

Finally, we derive the optimal pair (P_{cooling}, s) by determining the energy consumption of each pair with its corresponding T_s, which is given by

$$E = \int_0^{t_e} P_{\text{cpu}}(T_s, T_a, s, v_a)dt + D \cdot P_{\text{cooling}}. \quad (22)$$

As an example, we solve Problem 2 with $\mathcal{T} = (7 \times 10^7 \text{ cycles}, 40 \text{ ms})$ with the same microprocessor and the air-cooled heat sink model in Section IV.B. We also set the operating range of the E7330 processor from 1.87 GHz at 1.2 V to 3.00 GHz at 1.35 V. The result is shown in Fig. 10.

D. Stationary Periodic Real-Time Workload for Discrete DVFS

This subsection introduces an online algorithm to derive the total-energy-optimal fan speed together with the supply voltage and frequency of the microprocessor. We consider discrete DVFS for practical usage. It goes without saying that the optimal solution of the continuous DVFS case in Problem 2 is not the optimal solution for the discrete DVFS case. The optimal solution in the continuous domain may not be optimal or feasible in the discrete domain.

Thus, we introduce an online heuristic algorithm of which solution is close to the optimal based on the solution of Problem 2. We use a control-theoretic approach to keep track of the total energy-optimal DVFS scaling factor and cooling fan speed. As we assumed, we derive the optimal fan speed from Problem 2 and

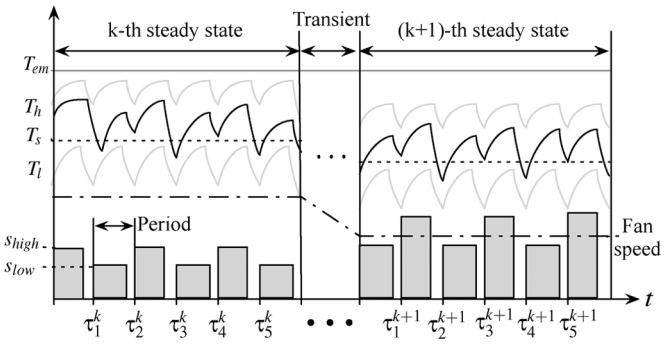


Fig. 11. Control-theoretic energy-optimal temperature-tracking DTM.

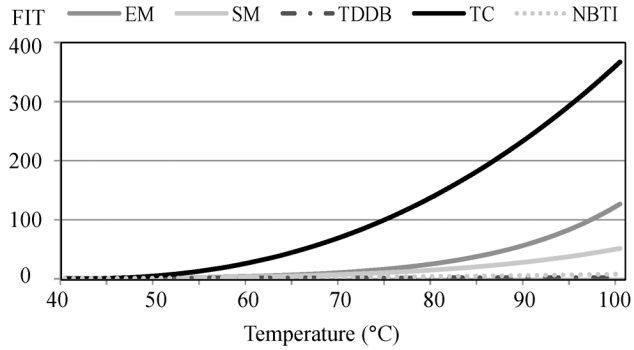


Fig. 12. The FIT values for five different lifetime reliability model with different average die temperature.

do not change the fan speed for the feedback control. Thus the only control knob is s. The detailed control policy is as follows:

1) For given $\mathcal{T}$ and T_a, obtain the energy-optimal T_s and the corresponding value of P_{cooling} from (21) and (22). Among the discrete levels of s, select two adjacent values which stabilize T_d most closely to T_s; one of them converges to a temperature T_h higher than T_s (s_{low} in Fig. 11), and the other converges to a temperature T_l lower than T_s (s_{high} in Fig. 11).
2) Fix the thermal resistance to achieve optimal temperature.
3) Operate the microprocessor at s_{low} until the temperature of the microprocessor exceeds T_s.
4) If the estimated die temperature in the next period with current scaling factor would be higher than T_s and the estimated die temperature in the next period with s_{high} is lower than T_{em}, adjust the scaling factor to s_{high}.
5) If the estimated die temperature in the next period with s_{high} level would be higher than T_{em}, adjust the scaling factor to minimum s_i among $S = (s_{\text{high}}, \cdots, s_n)$ which does not make the peak temperature exceed T_{em} when the next cycle is operated at s_i in the next period.
6) If the estimated die temperature in the next period with current scaling factor would be lower than T_s and the estimated die temperature in the next period with s_{low} is lower than T_{em}, adjust the scaling factor to s_{low} at the beginning of the next period.
7) Repeat procedures 4 to 6.

Fig. 11 is an example of the temperature profile produced by this policy.

V. Reliability Analysis Models

The high die temperature not only results in an immediate breakdown, but also degrades long-term reliability of a microprocessor. The proposed energy-optimal thermal management lowers the die temperature of the microprocessor compared with the conventional DTM, and thus eventually enhances the reliability.

In this section, we introduce models to evaluate the reliability of a processor when we apply the proposed thermal management to the microprocessor. The fundamental purpose of the thermal management is to keep the processor from the operation failure caused by the thermal emergency. Reliability is known to be dependent on temperature. The proposed energy-optimal thermal management tends to lower the temperature of the microprocessor with less system-level power consumption, which eventually results in reliability enhancement.

Various reliability models of silicon devices have been introduced so far including Electro Migration (EM), Stress Migration (SM), Time-Dependent Dielectric Breakdown (TDDB), Thermal Cycling(TC), Negative Bias Temperature Instability (NBTI) [31]. We use the RAMP model [32] to evaluate those reliability models. RAMP represents the lifetime reliability of the processor in terms of Mean Time To Failure (MTTF) with following models:

$$\begin{aligned}
\text{MTTF}_{\text{EM}} &\propto (J - J_{\text{critical}})^{R_1} e^{\frac{E_{a\text{EM}}}{kT_d}} \\
\text{MTTF}_{\text{SM}} &\propto |T_0 - T_d|^{R_2} e^{\frac{E_{a\text{SM}}}{kT_d}} \\
\text{MTTF}_{\text{TDDB}} &\propto \left(\frac{1}{V_{\text{dd}}}\right)^{(R_3 - R_4 T_d)} e^{\frac{R_5 + R_6/T_d + R_7 T_d}{kT_d}} \\
\text{MTTF}_{\text{TC}} &\propto \left(\frac{T_d}{T_d - T_a}\right)^q \\
\text{MTTF}_{\text{NBTI}} &\propto \left[ln\left\{ R_8 / \left(1 + 2e^{\frac{R_9}{kT_d}}\right)\right\} Te^{\frac{R_{10}}{kT_d}} \right. \\
&\quad - ln\left\{ R_{11} / \left(1 + 2e^{\frac{R_{12}}{kT_d}}\right) - R_{13}\right\} \\
&\quad \left. \times Te^{\frac{R_{15}}{kT_d}} \right]^{\frac{1}{\beta}}
\end{aligned} \tag{23}$$

where J is the current density in the interconnect; J_{critical} is for the critical current density needed for electromigration to occur; $E_{a\text{EM}}, E_{a\text{SM}}$ are the value of the activation energy for the electromigration and the stress migration to occur; T_0 is the initial temperature of the device; k is the Boltzmanns constant; q is the value of the Coffin–Manson exponent constant; and $R_i(\text{i} = 1, 2 \ldots 15)$ are constant parameters based on [31].

The standard method for representing constant failure rates is Failures in Time (FIT), which means the number of failures per 10^9 device operating hours. Fig. 12 shows the lifetime reliability values from the RAMP model considering average die temperature. As we can see, the die temperature strongly affects the EM, SM, and TC. The thermal effect on the TDDB and NBTI are relatively small.

TABLE III
TOTAL POWER CONSUMPTION WITH OPTIMAL P_{cooling} AND BASELINE

Air-cooled heat sink					
s	V_{dd} (V)	f (GHz)	Optimal fan speed (RPM)	P_{total} (W) (Baseline)	P_{total} (W) (Optimal)
s_1	1.35	3.00	2699	114.14	109.01
s_2	1.30	2.67	2499	87.72	86.07
s_3	1.25	2.25	2310	66.17	65.79
s_4	1.20	1.87	2181	48.72	47.94
Water-cooled liquid cooling block					
s	V_{dd} (V)	f (GHz)	Optimal flow rate (L/min)	P_{total} (W) (Baseline)	P_{total} (W) (Optimal)
s_1	1.35	3.00	1.73	126.64	112.56
s_2	1.30	2.67	1.53	105.13	89.78
s_3	1.25	2.25	1.32	82.92	66.80
s_4	1.20	1.87	1.09	63.03	46.53

VI. EXPERIMENT

A. Power-Optimal P_{cooling} for Discrete Scaling Factors

In this section, we show the advantages of the proposed DTM method over a conventional DTM scheme. The proposed method can be summarized as follows. We analytically derive the total-energy-optimal P_{cooling} from (8) for the continuous execution at each scaling factor in the microprocessor model, as described in Section IV. Table III summarizes the results and compares the total power consumption of the baseline cooling points and the proposed DTM which operates the cooling devices at the optimized points. We use the same analytical model of Intel E7330 introduced in Section IV. The discrete scaling factors are $s_1 = (1.35\ \text{V}, 3.00\ \text{GHz}), s_2 = (1.30\ \text{V}, 2.67\ \text{GHz}), s_3 = (1.25\ \text{V}, 2.25\ \text{GHz})$, and $s_4 = (1.20\ \text{V}, 1.87\ \text{GHz})$.

The cooling power of the baseline scheme is determined regarding the operating range of the microprocessor and physical constraints of the cooling devices. The fan in a common air-cooled heat sink is usually capable of rotating at a few hundreds revolution per minute (RPM) to a few thousands RPM. The fan assembled in the Zalman CNPS-9700NT heat sink is able to rotate at up to 2800 RPM [22]. The flow rate of a common water-cooled heat-exchanger for desktop computers is up to 2 or 3 gallons per minute (GPM). The pump in the Thermaltake Bigwater770 is able to circulate the coolant at up to 2.2 GPM [25]. To take the average, we set the baseline speed at a 1400 RPM and a 1.1 $\text{GPM}(= 4.163\ \text{L/min})$.

The results for the air-cooled heat sink and the liquid cooler show different aspect. It turns out that the baseline for air-cooled heat sink runs the microprocessor at a slower fan speed than the optimum. If we increase the fan speed and thus the cooling power of the heat sink, the total power consumption will be reduced. However, for the case of liquid cooling, the baseline flow rate may incur overcooling. The power optimal flow rate is much smaller than the baseline. In such a case, the liquid cooler consumes more power than the optimal solution. Conventionally, the cooling system has been designed under the consideration of the power consumed by itself. As the temperature-dependent leakage power is continuously increasing, the relation between the temperature-dependent leakage power and the cooling power need to be considered.

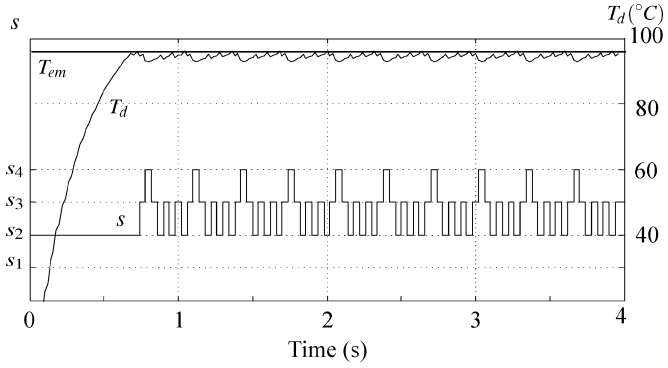


Fig. 13. DVFS scheduling result and temperature profile of a conventional threshold temperature triggered scheduling approach.

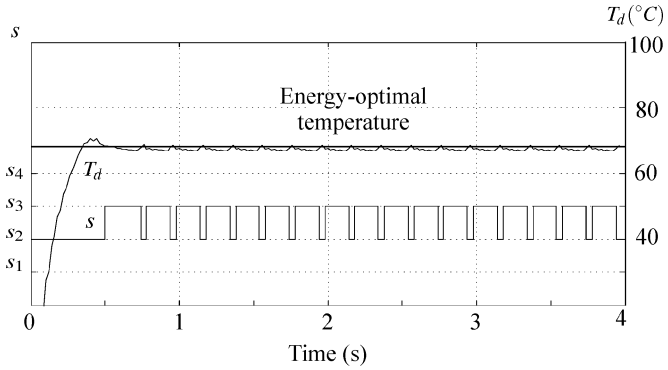


Fig. 14. DVFS scheduling result and temperature profile of the proposed optimal temperature tracking scheduling approach.

B. Control-Theoretic Energy-Optimal DTM With Discrete DVFS

We further perform a simulation of the proposed control-theoretic algorithm in Section IV-D to illustrate the difference between our approach and the baseline DTM. The task for the experiment has 7×10^7 cycle workload and 40 ms execution period. We also use scaling factors of $s_1 = (1.35\ \text{V}, 3.00\ \text{GHz}), s_2 = (1.30\ \text{V}, 2.67\ \text{GHz}), s_3 = (1.25\ \text{V}, 2.25\ \text{GHz})$, and $s_4 = (1.20\ \text{V}, 1.87\ \text{GHz})$ for the discrete DVFS scheduling. Figs. 13 and 14 show how our method maintains the die temperature at a lower value than the thermal emergency temperature of the baseline DTM.

In the proposed approach, we analytically derive the energy optimal P_{cooling} from (8) for a given task as described in Section IV. Then the control-theoretic algorithm adjusts the scaling factor to track the energy-optimal temperature. The baseline DTM method only uses a microprocessor throttling to avoid the thermal emergency. The baseline DTM scheme operates the microprocessor at the energy-optimal speed without considering the effect of temperature until the die temperature reaches a thermal emergency temperature, and periodically adjusts the scaling factor to avoid thermal emergency.

The baseline DTM results in relatively high die temperature compared to the proposed scheme because it controls the die temperature only when the emergency temperature has been reached on the basis of avoiding thermal emergency. The task for the experiment is the same as the example in Section VI-A.

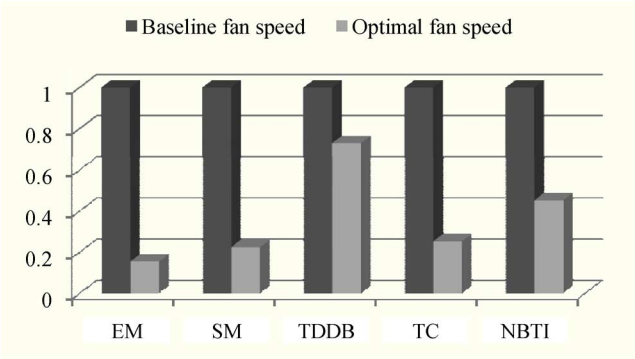


Fig. 15. Normalized FIT values for five different lifetime reliability models with the emergency temperature-triggered thermal management and energy-optimal temperature tracking thermal management using the parameters of the E7330.

A die temperature of 68°C minimizes the total energy consumption, and running the system at this temperature leads to 8.2% energy saving than running it at the threshold temperature of 95°C.

C. Reliability Analysis

We evaluate the lifetime reliability of the baseline DTM and proposed DTM by analyzing the result of the experiment in Section VI-B. If we use the baseline emergency temperature-triggered DTM with average cooling power, the die temperature may increase up to the emergency temperature which is higher than the energy-optimal temperature, and the high die temperature deteriorates the reliability of the microprocessor. On the other hand, we can achieve reliability enhancement as well as power reduction with the proposed scheme because it keeps the die temperature around the energy-optimal temperature which is lower than the emergency temperature. Fig. 15 shows normalized FIT values for five reliability models. The baseline DTM runs the cooling fan at 1400 RPM and uses scaling factors of s_1, s_2 and s_3, which results in average die temperature of 95°C. The proposed energy-optimal DTM runs the cooling fan 2260 RPM and uses scaling factors of s_2 and s_3, which result in average die temperature of 68°C. We see up to 84% FIT reduction thanks to the lowered die temperature.

D. Temperature-Dependent Leakage Power Variation of Real Microprocessor With the Fan Speed Control in the Air-Cooled Heat Sink

Our analysis of the simulation results show that the energy-optimal DTM significantly reduces the total power. Although we justified the proposed idea by the simulation results, we also make sure the feasibility of the proposed idea by performing actual power measurement of two different real microprocessors, the Intel E6850 and Q9650 assembled with the Zalman CNPS-9700NT heat sink.

Fig. 16 shows the measurement setup. We use high-precision measurement equipments including Agilent A34401 multimeter, Tektronix TDS2024B oscilloscope, TX3 multimeter, PS2521G power supply, Fluke 87III multimeter, and K-type temperature sensor to measure the fan power and speed, ambient

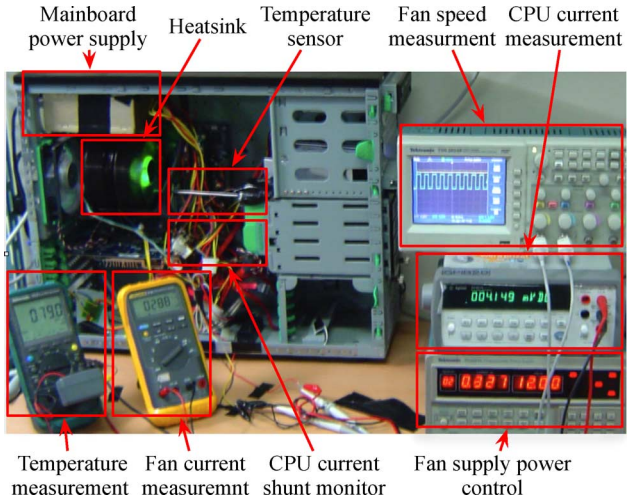


Fig. 16. Experimental setup for Intel E6850 and Q9650 microprocessors and Zalman CNPS-9700NT air-cooled heat sink.

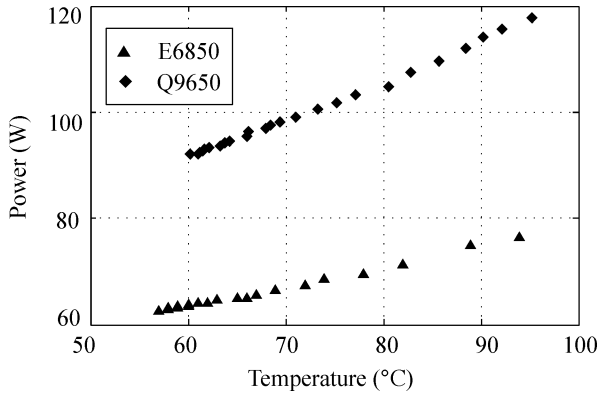


Fig. 17. Power variation of Intel E6850 and Q9650 microprocessors with different temperature.

temperature, and microprocessor power supply current, respectively. The fan speed can be calculated from the encoder pulse output of the fan motor while the fan power is measured by the multimeter. We use the Prime95 [33], which is a stress-test tool based on fast Fourier transforms as the microprocessor workload. The die temperature is measured directly from the on-chip thermal sensor in the microprocessor.

We vary the fan speed to measure the power consumption at different die temperatures. If we decrease the fan speed while the microprocessor performs the same workload, it increases the die temperature of the microprocessor. On the other hand, if we increase the fan speed when the microprocessor performs the same workload, it may decrease the die temperature of the microprocessor. The result of the measurement shows the tradeoff relationship between the temperature-dependent leakage power and the cooling power consumption.

As shown in Fig. 17, the power consumption of the E6850 and Q9650 microprocessors with Prime95 workload increases up to 18% and 22% respectively when the temperature of the microprocessors increases from 55°C to 95°C. The curve of total power consumption against the fan speed is convex as shown in Figs. 18 and 19. When compared to the power consumption at the baseline fan speed, the power reduction of the E6850 and

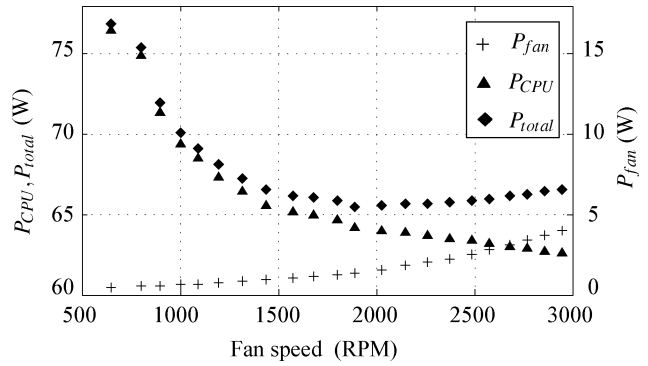


Fig. 18. Measured power consumption of the E6850 microprocessor and the fan with different cooling fan speed.

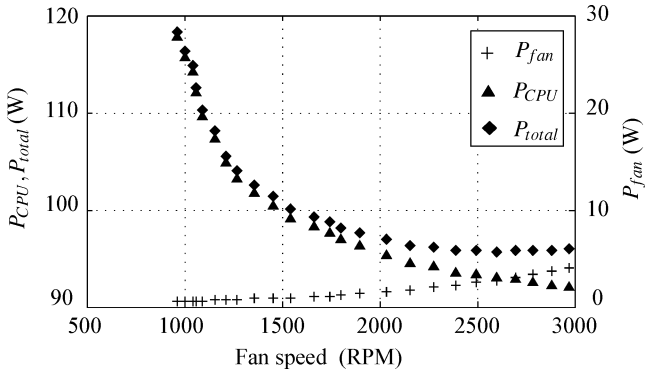


Fig. 19. Measured power consumption of the Q9650 microprocessor and fan with different cooling fan speed.

Q9650 is 4.4% and 9.6%, respectively. Note that the Q9650 microprocessor which is a quad-core processor using a 45 nm technology shows more temperature-dependent leakage power consumption than the E6850 microprocessor which is a dual-core processor using a 65 nm technology. This is because the higher integration density and smaller device size tend to increase the temperature-dependent leakage power consumption.

VII. Conclusion

This is the first paper that addresses the holistic energy optimization considering both computation and cooling power. We introduce a new dynamic thermal management (DTM) framework such that the thermal resistance of an active cooling device is a control variable. The goal of the proposed DTM is the total power consumption minimization while avoiding the thermal emergency. Our optimization framework derives the energy-optimal die temperature and tracks it as far as the throughput constraint is met, while conventional DTM maintains the die temperature as close as possible to the thermal emergency temperature to yield the maximum throughput. This enhances reliability of the semiconductor device as well. Experimental results demonstrate an 8.2% total energy saving and more than an 80% electromigration lifetime reliability enhancement. Our new DTM framework can be applied to various sorts of active cooling devices.

References

[1] D. Shin, J. Kim, J. Choi, S. W. Chung, E.-Y. Chung, and N. Chang, "Energy-optimal dynamic thermal management for green computing," in *Proc. ICCAD'09*, Nov. 2009, pp. 652–657.
[2] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors," in *Proc. HPCA'01*, 2001, pp. 171–182.
[3] R. Rao and S. Vrudhula, "Performance optimal processor throttling under thermal constraints," in *Proc. of CASES'07*, 2007, pp. 257–266.
[4] S. Zhang and K. S. Chatha, "Approximation algorithm for the temperature-aware scheduling problem," in *Proc. ICCAD'07*, 2007, pp. 281–288.
[5] S. Wang and R. Bettati, "Delay analysis in temperature-constrained hard real-time systems with general task arrivals," in *Proc. IEEE RTSS'07*, 2007, pp. 323–334.
[6] J.-J. Chen, S. Wang, and L. Thiele, "Proactive speed scheduling for real-time tasks under thermal constraints," in *Proc. 15th IEEE Real-Time and Embedded Technol. Appl. Symp.*, 2009, pp. 141–150.
[7] R. Jejurikar, C. Pereira, and R. Gupta, "Leakage aware dynamic voltage scaling for real-time embedded systems," in *Proc. DAC'04*, 2004, pp. 275–280.
[8] W. Liao, L. He, and K. M. Lepak, "Temperature and supply voltage aware performance and power modeling at microarchitecture level," *IEEE Trans. on CAD*, vol. 24, no. 7, pp. 1042–1053, Jul. 2005.
[9] L. Yuan, S. Leventhal, and G. Qu, "Temperature-aware leakage minimization techniques for real-time systems," in *Proc. ICCAD'06*, 2006, pp. 761–764.
[10] T. Heath, A. P. Centeno, P. George, L. Ramos, Y. Jaluria, and R. Bianchini, "Mercury and Freon: Temperature emulation and management for server systems," in *Proc. 12th Int. Conf. Architectural Support for Programming Languages and Operating Systems*, 2006, pp. 106–116.
[11] A. Ferreira, D. Moss, and J. Oh, "Thermal faults modeling using a RC model with an application to web farms," in *Proc. 19th Euromicro Conf. Real-Time Syst.*, 2007, pp. 113–124.
[12] R. Mukherjee and S. O. Memik, "Systematic temperature sensor allocation and placement for microprocessors," in *Proc. DAC'06*, 2006, pp. 542–547.
[13] J. S. Lee, K. Skadron, and S. W. Chung, "Predictive temperature-aware dvfs," *IEEE Trans. n Computers*, vol. 59, no. 1, pp. 127–133, 2010.
[14] J. Srinivasan and S. V. Adve, "Predictive dynamic thermal management for multimedia applications," in *Proc. ICS'03*, 2003, pp. 109–120.
[15] A. Merkel, F. Bellosa, and A. Weissel, "Event-driven thermal management in SMP systems," in *Proc. TACS'05*, 2005, pp. 1659–1664.
[16] W. Huang, K. Sankaranarayanan, K. Skadron, R. J. Ribando, and M. R. Stan, "Accurate, pre-RTL temperature-aware design using a parameterized, geometric thermal model," *IEEE Trans. Computers*, vol. 57, no. 9, pp. 1277–1288, Sep. 2008.
[17] Y. Liu, R. P. Dick, L. Shang, and H. Yang, "Accurate temperature-dependent integrated circuit leakage power estimation is easy," in *Proc. DATE'07*, 2007, pp. 1526–1531.
[18] M. Pedram and S. Nazarian, "Thermal modeling, analysis, and management in VLSI circuits: Principles and methods," in *Proc. IEEE*, 2006, vol. 94, pp. 1487–1501.
[19] R. J. Moffat, "Modeling air-cooled heat sinks as heat exchangers," in *Proc. Semi-Therm'07*, 2007, pp. 200–207.
[20] P. Teertstra, M. M. Yovanovich, and J. R. Culham, "Analytical forced convection modeling of plate fin heat sink," in *Proc. Semi-Therm'99*, 1999, pp. 34–41.
[21] K. Azar and B. Trabassoli, "How much heat can be extracted from a heat sink," *Electronics Cooling*, pp. 1–9, May 2003.
[22] Zalman CNPS-9700 NT User's Manual, 2007.
[23] S. G. Kandlikar and W. J. Grande, "Evaluation of single phase flow in microchannels for high heat flux chip cooling—Thermohydraurlic performance enhancement and fabrication technology," *Heat Transfer Engineering*, vol. 25, no. 8, pp. 5–16, 2004.
[24] D. S. Steinberg, *Cooling Techniques for Electroni Equipment*. New York: Wiley, 1980.
[25] Thermaltak BigWater770 User's Manual, 2008.
[26] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations," in *Proc. ISCA'00*, 2000, pp. 83–94.
[27] C. Isci and M. Martonosi, "Runtime power monitoring in high-end processors: Methodology and empirical data," in *Proc. MICRO'03*, 2003, pp. 93–104.
[28] Dual-Core Intel Xeon Processor 7200 Series and Quad-Core Intel Xeon Processor 7300 Series Datasheet, 2007.

[29] *Standard Performance Evaluation Corporation, CPU2000*, [Online]. Available: http://www.spec.org/osg/cpu2000
[30] K. Lee and K. Skadron, "Using performance counters for runtime temperature sensing in high-performance processors," in *Proc. IPDPS'05*, 2005, pp. 232–239.
[31] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, "The case for lifetime reliability-aware microprocessors," in *Proc. ISCA'04*, 2004, p. 276.
[32] *RAMP: Reliability-Aware MicroProcessors*, [Online]. Available: http://rsim.cs.illinois.edu/ramp/
[33] *Great Internet Mersenne Prime Search (GIMPS)*, [Online]. Available: http://www.mersenne.org/

Donghwa Shin (S'05) received the B.S. degree in computer engineering and the M.S. degree in electrical engineering and computer science from Seoul National University, Seoul, Korea, in 2005 and 2007, respectively. He is currently working towards the Ph.D. degree in electrical engineering and computer science at Seoul National University.

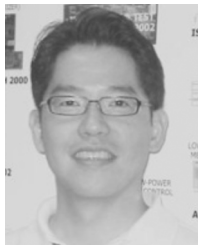
His research interest includes system-level thermal and power management for embedded system and portable electronic system design using next-generation power sources.

Sung Woo Chung (M'06) received the B.S. degree in computer engineering and the Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, Korea, in 1996 and 2003, respectively.

From 2003 to 2005, he was a Senior Engineer with SoC R&D Center, Samsung Electronics. He is currently an Associate Professor with the Division of Computer and Communication Engineering, Korea University. His research interest includes thermal management and process variation support for microprocessors.

Eui-Young Chung (M'06) received the B.S. and M.S. degrees in electronics and computer engineering from Korea University, Seoul, in 1988 and 1990, respectively, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 2002.

From 1990 to 2005, he was a Principal Engineer with SoC R&D Center, Samsung Electronics. He is currently an Associate Professor with the School of Electrical and Electronic Engineering, Yonsei University. His research interest includes system architecture and VLSI design, including all aspects of computer-aided design with the special emphasis on low-power applications and flash memory applications.



Naehyuck Chang (M'97–SM'05) received the B.S., M.S., and Ph.D. degrees from the Department of Control and Instrumentation, Seoul National University, Seoul, Korea, in 1989, 1992, and 1996, respectively.

He joined Department of Computer Engineering, Seoul National University, in 1997, and is currently a Professor with the Department of Electrical Engineering and Computer Science, Seoul National University. His current research focuses on low-power embedded systems, thermal management of large-scale computing systems, next-generation energy sources, and electrical energy storage systems.

Prof. Change is a Senior Member of ACM. He serves (and served) on the Technical Program Committee in many EDA conferences including DAC, ICCAD, ISLPED, DATE, CODES+ISSS, and ASP-DAC. He was a TPC (Co)-Chair of RTCSA 2007, ISLPED 2009 and ESTIMedia 2009 and 2010. He is General Vice-Chair of ISLPED 2010. He is currently an Associate Editor of IEEE TCAD and ACM TODAES, and a Guest Editor of ACM TECS and ACM TODAES in 2010 for real-time multimedia systems and low-power systems, respectively. He served as the Chair of ACM SIGDA Low-Power Technical Committee for 2008 and 2009. He is ASP-DAC SIGDA Representative and serves on the ACM SIGDA Executive Committee (Technical Activity Chair).