Leakage-Aware Real-Time Scheduling For Maximal Temperature Minimization

Gang Quan
Electrical and Computer Engineering Department
Florida International University
gang.quan@fiu.edu

Shangping Ren
Department of Computer Science
Illinois Institute of Technology
ren@iit.edu

***Abstract*—Thermal management problem has become a prominent issue as power consumption continues to grow exponentially. The leakage/temperature dependency becomes critical in power and thermal aware design as the processor continues to evolve into the the deep sub-micron domain. This paper seeks to explore fundamental principles in thermal aware design when taking the leakage/temperature dependency into considerations. We show and formally prove that, under certain realistic conditions, using the lowest constant processor speed that can guarantee deadlines of all real-time tasks is an optimal method to minimize the maximal temperature for a real-time system. We also use empirical results to justify the validation of this conclusion. We then discuss the possible future extension of this work.**

I. Introduction

The power consumption of the processors has been growing exponentially with each technology generation, and is expected to continuously grow rapidly in the future [1]. The soaring power consumption of processors has posed challenges not only on how to provide enough power source for a system, and but also how to manage the heat generated by the system. The escalating heat has directly led to high packaging and cooling costs, and threaten to significantly degrade the performance, life span, and reliability of computing systems, or even cause the system to fail [2], [3]. Therefore, as processors power consumption continues to rise, the thermal management problem has become an ever increasingly critical issue in the design of computing systems.

As semiconductor technology continues to scale down, the leakage plays a more and more important role [4], [5]. This is particularly true since the leakage power consumption is comparable or even dominates the dynamic power consumption in the deep sub-micron IC circuits. High power consumption causes high temperature, and high temperature increases leakage power and thus the overall power consumption. A thermal-conscious or power-conscious technique becomes ineffective if this temperature/leakage relation is not properly addressed in the deep sub-micron domain.

While reducing power consumption in general helps to lower the temperature, the temperature-constrained scheduling problem is drastically different from the energy-aware scheduling problem, as evidenced in recent studies [6], [7], [8], [9]. Therefore, new guidelines and principles on thermal aware computing need to be developed. Taking the leakage/temperature dependency into considerations makes the thermal aware design problem even more complex.

Consider a real-time job with deadline of $t = D$ and execution time of E. A well-known principle to reduce the energy, as shown by schedule S_1 in Figure 1, is to apply the lowest constant speed (i.e. v_0) within the entire interval so that the task just meets its deadline. Note that, when the leakage is taken into consideration [10], [11], S_1 is not necessarily optimal in terms of the overall energy reduction. In addition, previous researches [7], [8], [12] have shown that an optimal solution for energy minimization is not necessarily the optimal solution for peak temperature minimization. It is very suspicious that such a schedule has the lowest peak temperature. Alternative schedules include the one (i.e S_2) that first runs with a lower speed (i.e $v_1 < v_0$) and then a lower speed (i.e $v_2 > v_0$), or vice versa (S_3). When considering the leakage/temperature dependency, each schedule seems to have its own reasons to decrease or increase the maximal temperature. Then, the questions are: How should we execute the task judiciously such that the maximal temperature within the interval can be minimized? Are there any general guidelines that we can follow or we will have to deal with different scenarios case by case?

In this paper, we show that, under some realistic conditions, using the constant speed is the best way to minimize the peak temperature within an interval. We formulate this conclusion as a theorem and formally prove it. We also use empirical results to justify the conditions in the theorem. In the rest of the paper, Section II introduces system models and motivates our research. Section III presents our theorem and proof, as well as the empirical results to justify our theorem. We draw conclusions and point out our future work in Section IV.

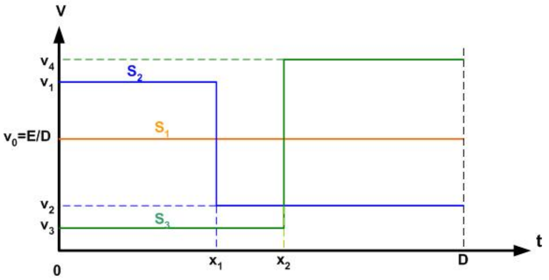

Fig. 1. Three schedules for a job set with deadline D and total execution time E.

II. System Model

We consider a real-time application consisting of n jobs, i.e. $\mathcal{J} = \{J_0, J_1, \cdots, J_{n-1}\}$, and all jobs have a common deadline D. Each job J_i has a worst case execution cycle of e_i, and the total workload of the job set is denoted as E. Since all jobs have the same deadline, we can equivalently treat the model as a single job with deadline D and work load E, and $E = \sum_{i=0}^{n-1} e_i$.

A. Thermal Model

The thermal model used in our paper is similar to that in Shadorn et al. [13]. Specifically, assuming a fixed ambient temperature (T_{amb}), let $T(t)$ be the temperature at time t, and we have

$$R_{th}C_{th}\frac{dT(t)}{dt} + T(t) - R_{th}P(t) = T_{amb}, \qquad (1)$$

where $P(t)$ denotes the power consumption (in $Watt$) at time t, and R_{th}, C_{th} denote the thermal resistance (in $J/^oC$) and thermal capacitance (in $Watt/^oC$), respectively. We can then scale T such that T_{amb} is zero and get

$$\frac{dT(t)}{dt} = aP(t) - bT(t), \qquad (2)$$

where $a = 1/C_{th}$ and $b = 1/R_{th}C_{th}$. For the rest of the paper, we assume that the initial temperature for the processor equals to its ambient temperature.

B. Power Model

According to Liao et al. [4], the leakage power can be estimated by

$$P_{leak} = N_{gate} \cdot I_{leak} \cdot v \qquad (3)$$

where N_{gate} is the total number of gates, I_{leak} is the leakage current, v is the supply voltage, and

$$I_{leak} = I_s \cdot (A \cdot T^2 \cdot e^{((\alpha \cdot v + \beta)/T)} + B \cdot e^{(\gamma \cdot v + \delta)}) \qquad (4)$$

where I_s is the leakage current based on a pre-determined reference temperature and supply voltage, T is the system's operating temperature, and A, B, α, β, γ, and δ are technology dependent constants. Some researches, such as that by Bao et al. [14], employ equation (4) directly to capture the leakage/temperature dependency in scheduling analysis. However, due to the non-linear and high-order magnitude terms in equation (4), such a model or tool can be too complex and cumbersome to be used for more rigorous real-time analysis and scheduling technique development.

Liu et al. [12] showed that, for a given supply voltage, the leakage changes with temperature super linearly. Based on this observation, a number of researches (such as [15], [16]) adopt a simple temperature/leakage linear model that assumes the leakage current changes linearly *only* with temperature. However, as can be seen from equation (4), leakage varies not only with temperature but also supply voltage as well. We thus approximate the leakage power for a processor with the following linear function

$$P_{leak}(t) = c_0 v(t) + c_1 T(t) \cdot \qquad (5)$$

where c_0 and c_1 are constants, and $v(t)$ is the supply voltage at time t. Constants c_0 and c_1 can be determined by curve fitting based on equation (4). As can be seen from equation (5), we model the leakage such that it changes with both the temperature and supply voltages.

For dynamic power, we assume [17]

$$P_{dyn}(k) = c_2 \cdot v^3(t) \qquad (6)$$

c_2 is also a constant and can be determined through profiling. Based on equation (5), (6), and (2), we have

$$\frac{dT(t)}{dt} = A(v(t)) - BT(t) \qquad (7)$$

where

$$A(v(t)) = a(c_0 v(t) + c_2 v^3(t)) \qquad (8)$$

and

$$B = (b - ac_1). \qquad (9)$$

Furthermore, if the processor runs at a constant speed $v(t) = v$ during the interval $[t_0, t_e]$, let the starting temperature be T_0, by solving equation (2), the ending temperature can be formulated as below:

$$T_e = G(v) + (T_0 - G(v))e^{-B(t_e - t_0)} \qquad (10)$$

where

$$G(v) = \frac{A(v)}{B}. \qquad (11)$$

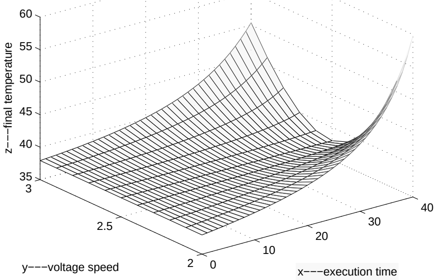

Fig. 2. Temperature varies with different supply voltages.

C. Motivating Example

We are not sure if there exist some general guidelines or we have to develop appropriate scheduling techniques case by case to minimize the peak temperature when the leakage/temperature relationship is taken into considerations. Therefore, we conducted some experiments to obtain some intuitions. We generated a large number of different schedules for a real-time job with deadline $D = 50$ and total workload as $E = 125$. We simulated the maximal temperature for each schedule and the results are shown in Figure 2.

From Figure 2, we can see that the peak temperatures by different schedules exhibit a "U" shape, and the peak temperature reaches its minimum when the lowest constant speed is applied. This seems to imply that using the lowest constant speed can minimize the maximal temperature. In what follows, we formulate this conclusion into a theorem and formally prove its correctness.

III. Minimize Peak Temperature

The experiments conducted in previous section seem to indicate that executing a real-time job with the lowest constant speed minimizes the peak temperature. This observation is valid under certain conditions. We formulate the conclusion by Theorem 1.

Theorem 1: Given a real-time job set $\mathcal{J}$, its deadline D and total execution time E, assume that the processor speed is continuously changeable. Then using the lowest constant speed that meets the deadline, i.e., $v_0 = E/D$, is the optimal scheduling solution in terms of minimizing the maximal temperature, if the following condition hold:

- $B > 0$;
- $G(v)$ is a non-negative, monotonically increasing, and convex function of v,

where B, G are defined in equation (9) and (11), respectively. **Proof Sketch:** Due to page limit, we only prove the case that, for the two schedules S_1 and S_2 shown in Figure 1, the temperature by S_1 at $t = D$ is no greater than that by S_2. For simplicity, we set $D = 1$ and also assume that $T_{amb} = 0$.

Let $T(S_1)$ and $T(S_2)$ be the ending temperatures for S_1 and S_2, respectively. Then from equation (10), we have

$$\begin{aligned} T(S_1) &= G(v_0)(1-e^{-B}), \\ T(S_2) &= G(v_2)(1-e^{-B(1-x)}) + G(v_1)(1-e^{-Bx})e^{-B(1-x)}. \end{aligned}$$

To prove that $T(S_1) \leq T(S_2)$, we only need to show that

$$\begin{aligned} G(v_0)(1-e^{-B}) &\leq G(v_2)(1-e^{-B(1-x)}) \\ &+ G(v_1)(1-e^{-Bx})e^{-B(1-x)}, \end{aligned} \tag{12}$$

Or

$$G(v_0) \leq kG(v_1) + (1-k)G(v_2), \tag{13}$$

where

$$k = \frac{e^{-B(1-x)} - e^{-B}}{1-e^{-B}}, 1-k = \frac{1-e^{-B(1-x)}}{1-e^{-B}}. \tag{14}$$

Since

$$v_0 = v_1 x + v_2(1-x), \tag{15}$$

and G_i is a convex function, we have

$$G(v_0) \leq xG(v_1) + (1-x)G(v_2). \tag{16}$$

Therefore, to show that equation (13) holds, we only need to show that

$$xG(v_1) + (1-x)G(v_2) \leq kG(v_1) + (1-k)G(v_2), \tag{17}$$

or

$$(G(v_1) - G(v_2))(x-k) \leq 0. \tag{18}$$

As G_i is monotonically increasing and $v_1 < v_2$, so we have $G(v_1) \leq G(v_2)$, and thus we only need to prove that

$$x \geq k = 1 - \frac{1-e^{-B(1-x)}}{1-e^{-B}}. \tag{19}$$

Or, equivalently,

$$\frac{1-e^{-B(1-x)}}{1-e^{-B}} \geq 1-x. \tag{20}$$

Now consider function

$$F(z) = \frac{1-e^{-Bz}}{1-e^{-B}} - z. \tag{21}$$

with $0 \leq z \leq 1$. We can readily show that function $F(z)$ is a concave function since $F''(z) < 0$. Note that the curve $F(z)$ passes two points, i.e. $(0,0)$ and $(1,0)$, as $F(0) = 0$ and $F(1) = 0$. Let $H(z)$ be the line that crosses these two points. Since $F(z)$ is concave, we have $F(z) \geq H(z) = 0$ for $0 \leq z \leq 1$. Therefore,

$$F(1-x) = \frac{1-e^{-B(1-x)}}{1-e^{-B}} - (1-x) \geq 0. \tag{22}$$

As a result, we prove that equation (19) and thus equation (19) holds. □

A. Justifications for Theorem 1

Theorem 1 holds only when several conditions are satisfied. In this subsection, we justify these conditions.

Consider equation (5). Note that $c_0 v(t)$ represents the leakage power at the ambient temperature, and $c_1 T(t)$ represents the increased leakage power consumption as temperature rises above the ambient temperature. From equation (4), it is not difficult to see that the leakage current increases as the temperature increases. Therefore, constants c_0 and c_1 must be non-negative and thus $A(v(t)) > 0$.

Moreover, based on (7), if $B = b - ac_1 < 0$, we would have

$$\frac{dT(t)}{dt} = A(v(t)) - BT(t) > 0, \tag{23}$$

and temperature will continue to increase indefinitely. This occurs only when the processor heat generation surpasses its heat removal capability, and thus the temperature will continue to rise and eventually cause the processor to break down. This scenario is called the "thermal run-away" [4]. Processor with this characteristic cannot work stably. Therefore, to avoid this scenario, $B > 0$ must hold. As a result, we can also conclude that

$$G(v) = \frac{A(v)}{B} \geq 0. \tag{24}$$

Theorem 1 also requires that $G(v)$ is a convex function of v. However, it is difficult to analytically prove that $G(v)$ is a convex function, since the temperature invariants c_0 and c_1 depend not only on the supply voltages but also on the technology parameters. Furthermore, c_0 and c_1 are obtained through curve-fitting rather than a closed analytical formula. In what follows, we try to make the justification empirically.

We built our processor model based on the work by Liao et al. [4] using the 65nm technology. We used (4) to compute the leakage currents for temperature from 40^oC to 110^oC with step size of 10^oC, and supply voltage from 0.65Volt to 1.05Volt with step size of 0.05V. These results were used to determine the temperature invariants c_0 and c_1 in (5) through curve-fitting. To obtain the leakage power consumption, the

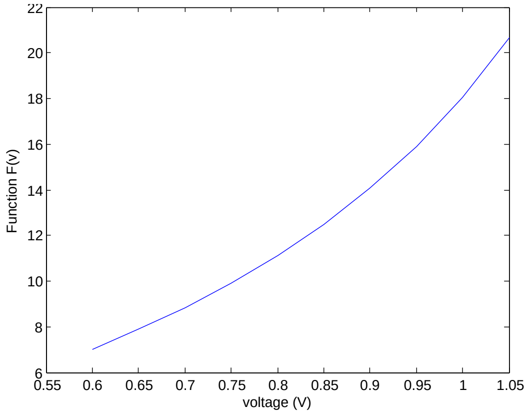

Fig. 3. Function $G(v)$ based on 65nm technology.

total number of gate, i.e., N_{gate} in (3), was set to be 10^6. The dynamic power consumption (and thus constant c_2) was determined based on the experimental results reported in [4] on benchmark *gcc*. For the thermal constants, we selected $Rth = 0.8K/W$, $Cth = 340J/K$ [2], and the ambient temperature was set to 25^oC. Figure 3 depicts the behavior of function $G(v)$ based on our experimental set up. We can clearly see from Figure 3 that function $G(v)$ is a non-negative, monotonic increasing, and most importantly, convex function. This justifies the conditions in Theorem 1.

IV. Conclusions and Future Work

As semiconductor technology continues to scale down in size, the positive feedback loop between the temperature and leakage becomes a critical issue not only for the power/energy minimization problem, but also for the temperature constrained design problem. In this paper, we intend to explore some fundamental principles that can be used when considering the leakage/temperature dependency in thermal aware real-time analysis. Our experimental results reveal that using the lowest constant speed is the optimal method to reduce the maximal temperature. We formulate this observation into a theorem and prove it formally. We also use empiric results to justify the conditions we present in the theorem. The significance of our work is that it clearly demonstrates the feasibility to incorporate the leakage/temperature into a more rigorous and analytical system level analysis. It also reveals a fundamental principle which can be applied in analyzing and developing leakage-aware temperature-constrained real-time scheduling techniques.

Our work can be extended in a number of ways. First, in this paper, we develop our theorem based on a processor model with continuously supply voltage. We want to extend this principle for processor models with discrete level of supply voltages. Second, this paper uses a very simple real-time model. How to extend the real-time model to a more practical and complex ones, such as those with priority assignments and preemption effects, will be an interesting problem. Note that, while our theorem seems to be very close to the well-known principles in power-aware scheduling, it does not mean that the existing methods for reducing energy consumption can be readily migrated for maximal temperature constraint. How to develop more effective techniques based on the principle we formulate in this paper will be an important future work for us. Third, our theorem is based on 65nm technology. As technology continues to scale down, it is not clear if all conditions supporting our theorem will still hold. Our next task is to study these cases.

Acknowledgment

This work is supported in part by NSF under Grants CNS-0545913, CNS-0917021, and CNS-0746643.

References

[1] ITRS, *International Technology Roadmap for Semiconductors (2005 Edition)*. Austin, TX.: International SEMATECH, http://public.itrs.net/.

[2] K. Skadron, M. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, “Temperature-aware microarchitecture,” *ICSA*, pp. 2–13, 2003.

[3] L.-T. Yeh and R. C. Chu, *Thermal Management of Microelectronic Equipment: Heat Transfer Theory, Analysis Methods, and Design Practices*. New York, NY: ASME Press, 2002.

[4] W. Liao, L. He, and K. Lepak, “Temperature and supply voltage aware performance and power modeling at microarchitecture level,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 7, pp. 1042 – 1053, 2005.

[5] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan, “Hotleakage: a temperature-aware model of subthreshold and gate leakage for architects,” *University of Virginia Dept. of Computer Science Technical Report*, 2003.

[6] K. Skadron, M. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, “Temperature-aware computer systems: opportunities and challenges,” *IEEE Micro*, vol. 23, no. 6, pp. 52–61, 2003.

[7] N. Bansal, T. Kimbrel, and K. Pruhs, “Speed scaling to manage energy and temperature,” *Journal of the ACM*, vol. 54, no. 1, pp. 1–39, 2007.

[8] G. Quan, Y. Zhang, W. Wiles, and P. Pei, “Guaranteed scheduling for repetitive hard real-time tasks under the maximal temperature constraint,” *ISSS+CODES*, 2008.

[9] S. Zhang and K. S. Chatha, “Approximation algorithm for the temperature-aware scheduling problem,” in *ICCAD*, 2007, pp. 281–288.

[10] R. Jejurikar, C. Pereira, and R. Gupta, “Dynamic slack reclamation with procrastination scheduling in real-time embedded systems,” *DAC*, 2005.

[11] G. Quan and L. Niu, “Fixed priority scheduling for reducing overall energy on variable voltage processors,” *RTSS’04*, Dec 2004.

[12] Y. Liu, H. Yang, R. P. Dick, H. Wang, and L. Shang, “Thermal vs energy optimization for dvfs-enabled processors in embedded systems,” in *ISQED*, 2007, pp. 204–209.

[13] K. Skadron, T. Abdelzaher, and M. R. Stan, “Control-theoretic techniques and thermal-rc modeling for accurate and localized dynamic thermal management,” in *HPCA ’02: Proceedings of the 8th International Symposium on High-Performance Computer Architecture*, 2002, p. 17.

[14] M. Bao, A. Andrei, P. Eles, and Z. Peng, “On-line thermal aware dynamic voltage scaling for energy optimization with frequency/temperature dependency consideration,” in *Design Automation Conference*, 2009, pp. 490–495.

[15] J.-J. Chen, S. Wang, and L. Thiele, “Proactive speed scheduling for real-time tasks under thermal constraints,” *RTAS*, vol. 0, pp. 141–150, 2009.

[16] N. Fisher, J.-J. Chen, S. Wang, and L. Thiele, “Thermal-aware global real-time scheduling on multicore systems,” *RTAS*, vol. 0, pp. 131–140, 2009.

[17] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*. Prentice Hall, 2003.