

Energy-Efficient Real-Time Task Scheduling with Temperature-Dependent Leakage

Chuan-Yue Yang[†], Jian-Jia Chen[‡], Lothar Thiele[‡], Tei-Wei Kuo[†]

[†]Department of Computer Science and Information Engineering, National Taiwan University

[‡]Computer Engineering and Networks Laboratory (TIK), ETH Zürich

Email: [†]{r92032, ktw}@csie.ntu.edu.tw, [‡]{chen, thiele}@tik.ee.ethz.ch

Abstract—Leakage power consumption contributes significantly to the overall power dissipation for systems that are manufactured in advanced deep sub-micron technology. Different from many previous results, this paper explores leakage-aware energy-efficient scheduling if leakage power consumption depends on temperature. We propose a pattern-based approach which divides a given time horizon into several time segments with the same length, where the processor is in the active (dormant, respectively) mode for a fixed amount of time at the beginning (end, respectively) of each time segment. Computation is advanced in the active mode, whereas the dormant mode helps reduce the temperature via cooling as well as the leakage power consumption. Since the pattern-based approach leads to a steady state with an equilibrium temperature, we develop a procedure to find the optimal pattern whose energy consumption in steady state is the minimum. Compared to existing work, our approach is more effective, has less run-time scheduling overhead, and requires only a simple scheduler to control the system mode periodically. The paper contains extensive simulation results which validate the new models and methods.

Index Terms—energy-efficient task scheduling, temperature-dependent leakage power consumption, real-time systems.

I. INTRODUCTION

Due to the urgent demand to prolong the life time of batteries in embedded systems and to reduce power bills for cutting the maintenance cost in server systems, power management has become a prominent system design issue. For micrometer-scale semiconductor technology, the dynamic power due to switching activities, which can be reduced with the dynamic voltage scaling (DVS) technique accompanied by DVS scheduling algorithms, e.g., [1], [14], [17], dominates the power consumption of a processor. However, for technology in the deep sub-micron (DSM) domain, the leakage power consumption due to leakage current is comparable to or even more than the dynamic power dissipation. In order to reduce the leakage power, dynamic power management (DPM) is proposed to change the system to a mode with lower leakage power consumption, i.e., a dormant/standby mode. But, it takes time and additional energy to change the system mode because

of the wakeup/dormition/hibernation of the processor and data fetch in the register/cache/memory [6], [15].

Recently, it has been shown that leakage current is highly dependent upon the temperature and it goes up rapidly when the temperature increases [7], [8]. For example, the leakage current in 65nm technology in [7] will increase by 21% when the temperature is raised from 60°C to 80°C. Research for thermal management mostly focuses on how to minimize the peak temperature under performance constraints [2], [13], [16], or how to maximize the performance under the peak temperature constraint, e.g., [4], [11]. For multiprocessor/multicore systems, temperature-aware scheduling has recently been widely explored, e.g., [5], [9].

This paper explores how to execute a given periodic workload under its timing constraint on a processor so that the energy consumption is minimized, where the leakage current is temperature-dependent. Yuan, Leventhal, and Qu also proposed a dynamic programming and an on-line algorithm for this problem [15]. However, the approach in [15] might not derive good steady solutions for periodic systems since it only considers how to schedule in one time horizon. Moreover, the algorithms in [15] ignore the energy overhead for mode switches to make scheduling decisions, which will lead to energy inefficiency. The contributions of this paper are as follows: (1) We provide a formula to approximate the existing model of the leakage current so that the resulting temperature of a schedule can be derived efficiently. (2) We propose a pattern-based approach to derive a regular pattern for deciding when to turn the processor to the active mode for task executions and when to turn to the dormant mode for cooling. (3) Since the pattern-based approach leads to a steady state with equilibrium temperature, we present how to find the optimal steady state to minimize the energy consumption. (4) Compared to the existing approach [15], our approach has less run-time scheduling overhead, and has 5 ~ 8% improvement for energy savings in our simulations. If only the manageable energy consumption, i.e., the energy consumption resulting from the leakage power and mode switches, are considered, the improvement for the energy reduction can achieve 38%.

The rest of this paper is organized as follows: Section II shows the system models and the problem definition. Section III presents how to calculate the temperature function, while Section IV covers the derivation of the equilibrium temperature in a steady state for energy consumption minimiza-

This research was done while Chuan-Yue Yang was visiting TIK, ETH Zürich, and is supported in part by grants from the Ministry of Economic Affairs under grant No. 96-EC-17-A-01-S1-034, the NSC under grant No. NSC-096-2917-I-564-121 and NSC-97-2221-E-002-206-MY3, the Excellent Research Projects of National Taiwan University under grant No. 98R0062-01 and No. 98R0062-05, and European Community's Seventh Framework Programme FP7/2007-2013 project Predator (Grant 216008).

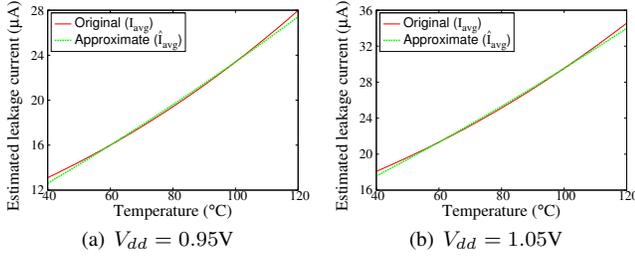


Fig. 1. The leakage current estimated by different formulae.

tion. Simulation results are shown in Section V. Section VI concludes this paper.

II. SYSTEM MODELS

In this section, the problem definition, the workload model, and the hardware characteristics, including the power consumption model and the thermal model, are presented.

The Power Consumption Model: This work considers the temperature dependency of leakage current, which has been modeled by Liao et al. based on the curve fitting of power consumption for different circuit types at multiple temperatures through SPICE simulations [7]:

$$I_{avg}(T, V_{dd}) = I(T_0, V_0) \left(AT^2 e^{\left(\frac{\alpha \cdot V_{dd} + \beta}{T}\right)} + B e^{(\gamma \cdot V_{dd} + \delta)} \right), \quad (1)$$

where $I(T_0, V_0)$ is the reference leakage current at a given temperature T_0 and with the reference supply voltage V_0 . $I_{avg}(T, V_{dd})$ denotes the leakage current at the temperature T and with the supply voltage V_{dd} . Note that the unit of temperature is the absolute temperature (K). Moreover, $A, B, \alpha, \beta, \gamma,$ and δ are empirical constants. Since the $e^{\frac{1}{T}}$ value of Equation (1) varies very slightly in the range of ordinary operating temperature of processors, e.g., $e^{\frac{1}{303}} = 1.0033$ at 30°C and $e^{\frac{1}{373}} = 1.00268$ at 100°C, the equation can be further simplified as follows with two reference temperatures T_H and T_L for a fixed supply voltage V_{dd} :

$$\hat{I}_{avg}(T) = \hat{A}T^2 + \hat{B}, \quad (2)$$

where $\hat{A} = (I_{avg}(T_H, V_{dd}) - I_{avg}(T_L, V_{dd})) / (T_H^2 - T_L^2)$ and $\hat{B} = I_{avg}(T_L, V_{dd}) - \hat{A}T_L^2$ are two constants.

For example, for the empirical data in [7] for 65nm technology, Figure 1 shows the leakage current estimated by different formulae. As shown in Figure 1, the leakage current values estimated by Equation (1) (denoted as *Original*) and Equation (2) (denoted as *Approximate*) are almost the same. For the rest of this paper, we use the formula of Equation (2) to model the temperature-dependent leakage current.

Processors considered in this work are assumed to have two system modes: The *active* and *dormant* modes. Task executions are only possible when their residing processor operates in the active mode; otherwise, the processor is in the dormant mode to reduce the power consumption and cool down. The power consumption of a processor in the active mode comes from two parts: The dynamic power P_{dyn} and the leakage power P_{lkg} , where the leakage power P_{lkg} of

the processor can be derived as $N_{gate} \cdot \hat{I}_{avg}(T, V_{dd}) \cdot V_{dd}$ for a constant N_{gate} depending on the characteristics of the target hardware. Since the supply voltage is fixed, the dynamic power P_{dyn} is a constant, and the leakage power P_{lkg} can be formulated as $\mathcal{A}T^2 + \mathcal{B}$, where \mathcal{A} and \mathcal{B} are $N_{gate}\hat{A}V_{dd}$ and $N_{gate}\hat{B}V_{dd}$, respectively. When a processor is in the dormant mode, its power consumption is bounded by a constant P_{dor} . However, it takes extra energy overhead E_{sw} and time delay t_{sw} to switch a processor between the active and dormant modes (including back and forth).

Thermal Model: Since there is a duality between the heat transfer and the electrical phenomena of RC circuits, the thermal RC model which can be formulated as the following differential equation is widely adopted to characterize the temperature behavior of a processor [3], [5], [10], [13], [15].

$$\begin{aligned} \frac{dT}{dt} &= \frac{1}{C_{th}}P - \frac{1}{R_{th}C_{th}}(T - T_{amb}) \\ &= \hat{\alpha}P - \hat{\beta}(T - T_{amb}), \end{aligned} \quad (3)$$

where T and T_{amb} are the temperature of the processor and the ambient temperature (in K), respectively, P is the power consumption of the processor (in Watt), R_{th} is the thermal resistance (in K/Watt), C_{th} is the thermal capacitance (in Joule/K), and $\frac{dT}{dt}$ is the temperature gradient in time (in K/second). When a processor operates in the active mode, the temperature might raise because part of its power consumption P will be converted into heat. The conversion ratio depends on the hardware characteristics of the processor and is captured by the constant $\hat{\alpha}$. In order to prevent the processor from overheating, the heatsink of a system is used to cool down the temperature by removing heat generated by the processor. The amount of temperature degradation is affected by the temperature gap between the processor and the environment, i.e., $T - T_{amb}$, and the hardware characteristics of the cooling system modeled by the constant $\hat{\beta}$.

Workload Model: In this work, we will focus our study on how to guarantee C time units of computation time within every consecutive D time units. The determination of C and D can be done based on different approaches according to the characteristics of tasks. In practice, such kind of workload may come from periodic or sporadic real-time tasks. For example, if $D - C$ is no greater than the procrastination interval [6] of each task in a periodic real-time task set, the set of tasks can be schedulable under such a setting of C and D . Moreover, the Real-Time Calculus [12] is also applicable to find a proper setting of C and D which can form a service curve to satisfy periodic and/or sporadic real-time tasks.

Problem Definition: The Temperature-Aware Leakage Energy Minimization (TALEM) problem is defined as follows: *Given a workload of C time units that must be done in every consecutive D time units, the TALEM problem is to schedule the given workload with the minimal total energy consumption and without violation of any timing constraint.* Although the energy consumption can be reduced by turning the processor to the dormant mode, frequent mode switchings might lead to

energy inefficiency because of the mode-switching overhead. Thus, the energy consumption resulting from the temperature-dependent leakage power and the mode-switching overhead should be taken into consideration at the same time when we determine when to perform mode switching.

III. TEMPERATURE FUNCTIONS

Since the leakage power consumption is temperature-dependent, after the processor has been in the active mode to execute tasks for a while, the temperature might be too high so that turning the processor to the dormant mode can cool down the processor to reduce the energy consumption. On the other hand, when the processor is in the dormant mode, we have to decide whether the temperature is low enough to switch to the active mode for task executions. Thus, to decide when to switch the processor modes, we need to know the temperature function when the processor is in the active mode as well as that when the processor is in the dormant mode.

Now, we explain how to derive the temperature functions of the target system according to the thermal model in Equation (3) and the leakage model in Equation (2). When the processor is in the active mode, the power consumption P includes both the dynamic power P_{dyn} and the leakage power P_{lkg} . By substituting $P_{dyn} + P_{lkg}$ for P in Equation (3), the temperature gradient in time when the processor is in the active mode can be modeled as the following equation:

$$\begin{aligned} \frac{dT}{dt} &= \hat{\alpha}(P_{dyn} + P_{lkg}) - \hat{\beta}(T - T_{amb}) \\ &= \dagger \hat{\alpha}(P_{dyn} + (\mathcal{A}T^2 + \mathcal{B})) - \hat{\beta}(T - T_{amb}) \\ &= (\hat{\alpha}\mathcal{A})T^2 - \hat{\beta}T + (\hat{\alpha}(P_{dyn} + \mathcal{B}) + \hat{\beta}T_{amb}) \\ &= aT^2 + bT + c, \end{aligned} \quad (4)$$

where \dagger comes directly from the definition of P_{lkg} . Then, by solving the differential equation shown in Equation (4), we can derive the temperature function of the processor when it operates in the active mode as follows:

$$T_{act}(\hat{t}, t) = \frac{-(k\theta_1 e^{\theta_1 t} + \theta_2 e^{\theta_2 t})}{a(k e^{\theta_1 t} + e^{\theta_2 t})} = \frac{-(k\theta_1 e^{(\theta_1 - \theta_2)t} + \theta_2)}{a(k e^{(\theta_1 - \theta_2)t} + 1)}, \quad (5)$$

where $\theta_1 = \frac{b + \sqrt{b^2 - 4ac}}{2}$, $\theta_2 = \frac{b - \sqrt{b^2 - 4ac}}{2}$, and k is related to the initial temperature at time \hat{t} at which the processor is turned to the active mode, i.e., $T_{act}(\hat{t}, 0) = \frac{-(k\theta_1 + \theta_2)}{a \cdot (k + 1)}$ and the processor is in the active mode during the time interval $(\hat{t}, \hat{t} + t]$.

On the other hand, as the power consumption of the processor in the dormant mode is a constant P_{dor} , the temperature gradient in time is as the follows:

$$\frac{dT}{dt} = \hat{\alpha}P_{dor} - \hat{\beta}(T - T_{amb}) = -\hat{\beta}(T - (T_{amb} + \frac{\hat{\alpha}}{\hat{\beta}}P_{dor})). \quad (6)$$

Solving the differential equation in Equation (6) leads to the temperature function when the processor is in the dormant mode:

$$T_{dor}(\check{t}, t) = (1 - e^{-\hat{\beta}t})(T_{amb} + \frac{\hat{\alpha}}{\hat{\beta}}P_{dor}) + T_{dor}(\check{t}, 0) \cdot e^{-\hat{\beta}t}, \quad (7)$$

where the processor is in the dormant mode in time $(\check{t}, \check{t} + t]$, and $T_{dor}(\check{t}, 0)$ is the temperature at time \check{t} .

IV. ENERGY MINIMIZATION

This section first presents the pattern-based approach and its properties. Then how to minimize the energy consumption accordingly is illustrated. At the end of this section, we will provide some remarks for relevant extensions.

A. The Pattern-Based Approach

The pattern-based approach is to divide the D time units into n time segments. In each time segment, we turn the processor to the active mode to execute tasks for $u\Delta$ time units at the beginning of the time segment, where $\Delta = \frac{D}{n}$ is the length of each time segment and u is $\frac{C}{D}$. Then, for the rest of $(1 - u)\Delta$ time units in the time segment, we turn the processor to the dormant mode to cool down its temperature if $(1 - u)\Delta \geq t_{sw}$. After the execution pattern is repeated for all n time segments, the C time units workload can be finished within the given D time units. Moreover, by repeating the scheduling pattern for every D time units, there will be C time units in which the processor is in the active mode within any consecutive D time units starting from any time instant. Thus, the requirement of the workload can be satisfied with the pattern-based approach.

The benefit of the proposed approach is the simplicity of implementation since the control of the processor conforms to a regular pattern. In addition, as there is no need for any on-line calculation, the on-line scheduling overhead is also reduced. Given a time segment, if the heat generated by the processor is more than that can be removed from the heatsink, the temperature of the processor at the end of the time segment will be higher than that at the beginning of the time segment. Since the end of a time segment is the beginning of the next time segment, by performing the execution pattern repeatedly, the temperatures at the beginning of time segments will get higher and higher. However, eventually, the heat that can be removed from the processor will be equivalent to that generated by the processor because the heat that can be removed from the heatsink is proportional to the difference of the temperature between the processor and the environment, i.e., the term $\hat{\beta}(T - T_{amb})$ in Equation (3).

Equilibrium Temperature in the Steady State: If we repeat the execution pattern for a sufficient number of times, the initial temperature and the ending temperature at the beginning and the end, respectively, of a time segment will become the same eventually. In such cases, the initial/ending temperature is called *equilibrium temperature* and the system is said in a *steady state*. Figure 2 depicts examples to illustrate such a phenomenon. When the processor executes tasks, the temperature of the processor grows as the curve of the function in Equation (5). Then the processor is turned to the dormant mode after $u\Delta$ time units, i.e., 0.06 second and 0.14 second in Figures 2(a) and 2(b), respectively. After the processor is turned to the dormant mode, the cooling down of the temperature follows the curve of the function in Equation (7).

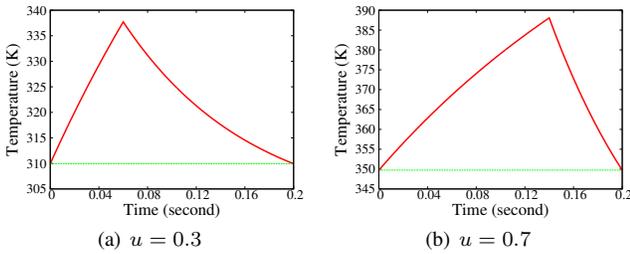


Fig. 2. Examples for different steady states.

Suppose that a time segment in the steady state starts at time \hat{t} . By the definition of the equilibrium temperature, the ending temperature $T_{dor}(\hat{t} + u\Delta, (1-u)\Delta)$ at time $\hat{t} + \Delta$ should be equal to the initial temperature $T_{act}(\hat{t}, 0)$ at time \hat{t} . Therefore, given a length Δ of a time segment, we can derive equilibrium temperature T_{eq} by solving the following equations:

$$T_{act}(\hat{t}, 0) = T_{eq} = T_{dor}(\hat{t} + u\Delta, (1-u)\Delta) = \frac{-(k \cdot \theta_1 + \theta_2)}{a \cdot (k+1)}$$

$$T_{dor}(\hat{t} + u\Delta, 0) = T_{act}(\hat{t}, u\Delta) \quad (8)$$

By adopting the definition of $T_{act}()$ and $T_{dor}()$ in Equations (5) and (7), respectively, the equations in Equation (8) can be rephrased as follows.

$$\frac{-(k\theta_1 + \theta_2)}{a(k+1)} = (1 - e^{-\hat{\beta}(1-u)\Delta})(T_{amb} + \frac{\hat{\alpha}}{\hat{\beta}}P_{dor})$$

$$+ \frac{-(k\theta_1 e^{(\theta_1-\theta_2)u\Delta} + \theta_2)}{a(ke^{(\theta_1-\theta_2)u\Delta} + 1)} e^{-\hat{\beta}(1-u)\Delta} \quad (9)$$

For simplicity of representation, let \bar{T}_{amb} denote $T_{amb} + \frac{\hat{\alpha}}{\hat{\beta}}P_{dor}$. Then, k can be derived by solving Equation (10).

$$\left[(a\bar{T}_{amb} + \theta_1)(1 - e^{-\hat{\beta}(1-u)\Delta}) \cdot e^{(\theta_1-\theta_2)u\Delta} \right] \cdot k^2$$

$$+ \left[(a\bar{T}_{amb} + \theta_1)(1 - e^{(\theta_1-\theta_2)u\Delta - \hat{\beta}(1-u)\Delta}) \right.$$

$$\left. + (a\bar{T}_{amb} + \theta_2)(e^{(\theta_1-\theta_2)u\Delta} - e^{-\hat{\beta}(1-u)\Delta}) \right] \cdot k \quad (10)$$

$$+ \left[(a\bar{T}_{amb} + \theta_2)(1 - e^{-\hat{\beta}(1-u)\Delta}) \right]$$

$$= \tilde{a}k^2 + \tilde{b}k + \tilde{c} = 0$$

Therefore, k will be either $\frac{-\tilde{b} + \sqrt{\tilde{b}^2 - 4\tilde{a}\tilde{c}}}{2\tilde{a}}$ or $\frac{-\tilde{b} - \sqrt{\tilde{b}^2 - 4\tilde{a}\tilde{c}}}{2\tilde{a}}$. After k is obtained, the equilibrium temperature $T_{eq} = \frac{-(k\theta_1 + \theta_2)}{a \cdot (k+1)}$ is also known.

In order to verify which k is valid, we need to know the feasible temperature range since the equilibrium temperature T_{eq} must be in the feasible temperature range. As the heat that can be removed from the heatsink is dependent on the temperature of the processor, the temperature of the processor will converge on a convergent temperature T_{conv} if we keep the processor in the active mode for a sufficient long time interval. Thus, the equilibrium temperature should not be higher than the convergent temperature, i.e., $T_{amb} \leq T_{eq} \leq T_{conv}$. By rephrasing Equation (5) as $T_{act}(0, t) = \frac{-\theta_1}{a} + \frac{\theta_1 - \theta_2}{a(ke^{(\theta_1-\theta_2)t} + 1)}$, we know the convergent temperature T_{conv} is $\frac{-\theta_1}{a}$ because

T_{conv} should be $T_{act}(0, \infty)$. If both values obtained from Equation (10) can derive equilibrium temperature in the feasible temperature range, the equilibrium temperature should be the one that is closer to the temperature of the processor right after the system is powered on, i.e., usually the ambient temperature.

B. Optimization of Time Segments

As any Δ with $(1-u)\Delta > t_{sw}$ and integral D/Δ is a feasible solution for the pattern-based approach, we have to determine the value of Δ for energy consumption minimization. Given a length of a time segment Δ , the pattern-based approach converges to the steady state in which $T_{act}(\hat{t}, t) = T_{act}(\hat{t} + \Delta, t)$ for any $0 \leq t \leq u\Delta$. Thus, we use $T_{act}^s(t)$ to denote $T_{act}(\hat{t}, t)$ for brevity. Let $E(n)$ be the overall energy consumption to complete the given workload when we divide the given time interval into n time segments. We have the following equation:

$$E(n) = n \left(\int_0^{\frac{C}{n}} P_{act}(t) dt + \int_0^{\frac{D-C}{n}} P_{dor} dt + E_{sw} \right)$$

$$= n \left(\int_0^{\frac{C}{n}} (P_{dyn} + \mathcal{A}(T_{act}^s(t))^2 + \mathcal{B}) dt + E_{sw} \right) + (D-C)P_{dor}$$

$$= \Psi(n) + C(P_{dyn} + \mathcal{B}) + (D-C)P_{dor}, \quad (11)$$

where $\Psi(n)$ is the *reducible energy consumption*, which is defined as $n \left(\int_0^{\frac{C}{n}} \mathcal{A}(T_{act}^s(t))^2 dt + E_{sw} \right)$. Since $C(P_{dyn} + \mathcal{B}) + (D-C)P_{dor}$ is a constant, we formulate the optimization of the pattern-based approach as the following formula:

$$\text{minimize } \Psi(n) = n \mathcal{A} \int_0^{\frac{C}{n}} (T_{act}^s(t))^2 dt + n \cdot E_{sw} \quad (12)$$

$$\text{subject to } (1-u)\frac{D}{n} \geq t_{sw}.$$

In Formula (12), the objective function $\Psi(n)$ can be further reformulated as follows:

$$\Psi(n) = n \frac{\mathcal{A}}{a^2} \cdot \int_0^{\frac{C}{n}} \left(\frac{k(n) \cdot \theta_1 \cdot e^{(\theta_1-\theta_2)t} + \theta_2}{k(n) \cdot e^{(\theta_1-\theta_2)t} + 1} \right)^2 dt + n \cdot E_{sw}$$

$$= n \frac{\mathcal{A}}{a^2} \left(\theta_2^2 t + (\theta_1 + \theta_2) \cdot \ln(k(n) \cdot e^{(\theta_1-\theta_2)t} + 1) \right.$$

$$\left. + \frac{\theta_1 - \theta_2}{k(n) \cdot e^{(\theta_1-\theta_2)t} + 1} \right) \Big|_0^{\frac{C}{n}} + n \cdot E_{sw}, \quad (13)$$

where $k(n)$ stands for the value of k when we divide the D time units into n time segments and can be derived from Equation (10) by setting $\Delta = \frac{D}{n}$. Afterwards, we can minimize the total energy consumption to finish C time units of workload within D time units by solving $\Psi'(n) = 0$.

To solve $\Psi'(n) = 0$, we can simply apply some mathematical tools. However, as we have a constraint on the values of n due to the overhead of mode switches, there will be a feasible range for the values of n , i.e., $1 \leq n \leq \min \left\{ \lfloor \frac{(1-u)D}{t_{sw}} \rfloor, \lfloor \frac{\Psi(1)}{E_{sw}} \rfloor \right\}$, where $\lfloor \frac{(1-u)D}{t_{sw}} \rfloor$ and $\lfloor \frac{\Psi(1)}{E_{sw}} \rfloor$ come from $\frac{(1-u)D}{n} \leq t_{sw}$ and $n \cdot E_{sw} \leq \Psi(1)$, respectively. As a result, the minimal $\Psi(n)$ can be also found by a sequential search of n from 1 to $\min \left\{ \lfloor \frac{(1-u)D}{t_{sw}} \rfloor, \lfloor \frac{\Psi(1)}{E_{sw}} \rfloor \right\}$.

C. Remarks

Now, we are going to discuss how to deal with peak temperature constraint, non-constant power consumption in the dormant mode, and dynamic voltage scaling. Since the peak temperature of the resulting schedule of the pattern-based approach is a decreasing function of n , the peak temperature constraint can be considered by adding a lower bound constraint of n . That is, $T_{act}(u\frac{D}{n}) \leq T_{cons}$ is added as a constraint in Equation (12), where T_{cons} is the temperature constraint. For systems with temperature-dependent dormant power consumption, the temperature function $T_{dor}()$ in Equation (7) must be revised into a similar form to Equation (5), and how to derive an optimal pattern is the same as the procedures in Section IV. For DVS systems with discrete supply voltages $V_{dd,1}, V_{dd,2}, \dots, V_{dd,M}$ with corresponding speeds s_1, s_2, \dots, s_M , suppose that C time units of workload is measured under the maximum speed s_M . Then, when the processor speed is scaled to s_i , we can recalculate the values of \mathcal{A} and \mathcal{B} according to $V_{dd,i}$, and replace C with $C\frac{s_M}{s_i}$ to estimate the optimal total energy consumption. We then choose a supply voltage that leads to the minimum total energy consumption as the supply voltage of the system and perform the corresponding execution pattern accordingly.

V. PERFORMANCE EVALUATIONS

In this section, we show the performance of our proposed pattern-based approach through a set of simulations.

Simulation Setup: In our simulations, we considered numerous hardware platforms by varying the hardware-specific parameters in different dimensions. We fixed the values of β , P_{dor} , t_{sw} , T_{amb} , and V_{dd} as 9.52/second, $50\mu\text{Watt}$, 5ms, 300K, and 1V, respectively, which were the same as the settings in [15], and the other parameters $\hat{\alpha}$, P_{dyn} , P_{lkg} , and E_{sw} were left as variables. We used the same workload configuration as [15], including eleven benchmarks. Due to the energy induced by the dynamic power P_{dyn} and dormant power P_{dor} was a constant under different approaches, we adopted the reducible energy consumption which was the summation of the energy resulting from the leakage power P_{lkg} and the energy overhead of mode switches, e.g., the value of $\Psi(n)$ for an optimal n in our pattern-based approach, as our performance metrics. Moreover, we used the naïve approach that only turned the processor to the dormant mode when the given workload was finished as the baseline. Then, the *normalized reducible energy* (abbreviated as NRE) is defined as the reducible energy consumption of an evaluated approach divided by that of the naïve approach. In addition to our pattern-based approach, we also evaluated the on-line approach presented in [15] for comparisons, where the lengths of each time interval, denoted by σ , for deciding whether the processor should be turn to the dormant mode or to the active mode were set to 0.1, 0.05, and 0.02 second. To compare the long term behavior among different approaches, each benchmark was simulated for $100D$ under each approach and configuration. Moreover, the *additional energy improvement* (abbreviated as AEI), defined as $\frac{NRE_{on-line} - NRE_{pattern-based}}{1 - NRE_{on-line}}$, is used to

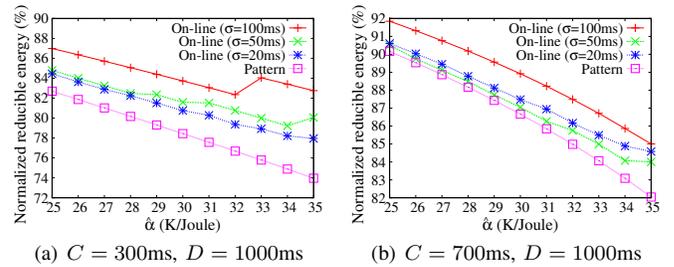


Fig. 3. Results when P_{dyn} is 5Watt and E_{sw} is 0.01Joule.

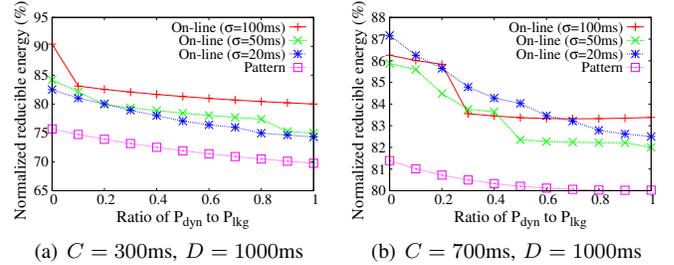


Fig. 4. Results when $\hat{\alpha}$ is 35.62K/Joule and E_{sw} is 0.01Joule.

show the improvement of the pattern-based approach compared to the approach in [15], where $NRE_{pattern-based}$ and $NRE_{on-line}$ are the NRE of our pattern-based approach and the best NRE among the approaches in [15] with different σ settings, respectively. Even though we use Equation (2) for approximation, the error in energy consumption in our simulations was no more than 0.3% compared to the leakage current estimation by Equation (1).

Simulation Results: Table I shows the simulation results for 11 benchmarks when \mathcal{A} is 0.0002188Watt/K², \mathcal{B} is -8.5143Watt , P_{dyn} is 5Watt, $\hat{\alpha}$ is 35.62K/Joule, and E_{sw} is 0.01Joule, where the number of mode switches in $100D$ is abbreviated as #sw. According to Table I, our pattern-based approach improves the NRE by 2 ~ 5%, where the AEI is 10 ~ 31%. Moreover, the smaller the value of u , the greater improvement the pattern-based approach can achieve. This is because the processor would stay longer in the dormant mode for cooling, when u is smaller. However, cooling is inefficient when the temperature is low. As a result, for smaller u , the on-line algorithm in [15] becomes energy inefficiency since the decision of when to turn the processor to/from the dormant mode is not determined carefully.

To evaluate the performance of different approaches on different hardware platforms, we performed some simulations by varying values of $\hat{\alpha}$, P_{dyn} , and E_{sw} while the other parameters are the same as the parameters for deriving Table I. The simulation results are presented in Figures 3, 4, and 5. Figures 3(a) and 3(b) show the results by varying $\hat{\alpha}$ when P_{dyn} is 5Watt and E_{sw} is 0.01Joule. A platform with a larger $\hat{\alpha}$ is easier to be heated up, and it is more crucial to make proper decisions for switching between different system modes. Therefore, as shown in Figures 3(a) and 3(b), our approach outperforms the on-line approach when $\hat{\alpha}$ is larger.

Since both dynamic power and leakage power contribute

Benchmark	D(ms)	W(ms)	naïve		On-line($\sigma=100$ ms)		On-line($\sigma=50$ ms)		On-line($\sigma=20$ ms)		Pattern		
			Reducible energy (Joule)	#sw	NRE	#sw	NRE	#sw	NRE	#sw	NRE	#sw	AEI
MPEG4	60000	50000	1878.9	100	72.3%	10000	72.2%	18900	74.0%	43600	69.0%	53900	11.51%
CH2	1000	300	5.9	100	82.3%	300	79.6%	600	77.9%	1300	73.4%	900	20.36%
CO	1000	150	2.4	100	89.7%	200	88.5%	400	85.5%	800	81.0%	600	31.03%
airflow	2000	200	3.5	100	86.7%	300	79.7%	500	76.8%	1000	72.8%	900	17.24%
ADSL1	576	285	5.7	100	91.9%	300	89.1%	500	88.8%	1099	85.8%	600	26.79%
ADSL2	2048	864	23.7	100	63.4%	900	62.9%	1600	63.4%	3500	58.4%	2100	12.13%
Bmk1	1000	400	8.7	100	81.8%	300	77.3%	700	77.0%	1600	72.5%	1000	19.57%
Bmk2	1000	500	11.7	100	77.3%	400	76.9%	800	77.8%	1700	73.7%	1100	13.85%
Bmk3	1000	600	14.9	100	79.5%	400	79.1%	799	80.3%	1600	76.6%	1100	11.96%
Bmk4	1000	700	18.4	100	84.4%	300	83.4%	600	84.3%	1400	81.4%	1100	12.05%
Bmk5	1000	800	22.4	100	91.4%	201	88.9%	400	89.6%	1000	87.7%	1000	10.81%

TABLE I
SIMULATION RESULTS FOR DIFFERENT BENCHMARKS WHEN P_{dyn} IS 5WATT, $\hat{\alpha}$ IS 35.62K/JOULE, AND E_{sw} IS 0.01JOULE.

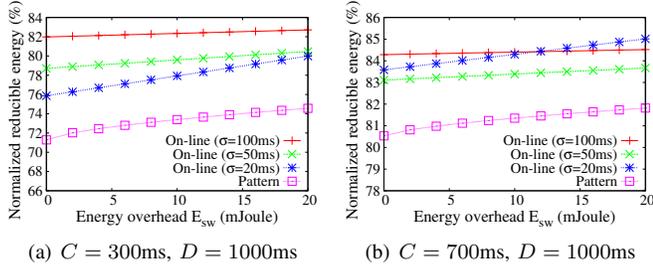


Fig. 5. Results when $P_{dyn} = 5$ Watt and $\hat{\alpha} = 35.62$ K/JOULE.

to the heat generation, we are curious about the effect of the dynamic power on the performance of different approaches. Figures 4(a) and 4(b) show the NRE for different ratios of the dynamic power consumption P_{dyn} to the leakage power consumption P_{lkg} at the ambient temperature when $\hat{\alpha}$ is 35.62K/JOULE, and E_{sw} is 0.01JOULE. When the ratio is low, the leakage power dominates the total power consumption. Thus, a good temperature-dependent leakage minimization approach is more important for such cases. As a result, our approach can improve the normalized reducible energy by up to 8% when the dynamic power consumption is smaller.

Figures 5(a) and 5(b) show the NRE for different energy overheads E_{sw} of the mode switch when P_{dyn} is 5Watt and $\hat{\alpha}$ is 35.62K/JOULE. Since the on-line approach proposed in [15] does not consider the energy overhead resulting from mode switches, it tends to switch the system mode frequently when length of interval is 0.02 second, and its performance becomes worse for larger energy overhead of the mode switch. Moreover, in Figure 3, the additional energy improvement (AEI) is 2 ~ 38%.

VI. CONCLUSION

This paper exploits the possibility to turn the processor to the dormant mode for cooling to minimize the energy consumption for real-time workload when the leakage power consumption is dependent on the temperature. We propose a pattern-based approach to derive a regular pattern for deciding when to turn the processor to the active and dormant modes for task executions and cooling, respectively. With a regular execution pattern, the amount of workload that can be done within any consecutive fixed length of time interval is constant. Thus, our pattern-based approach can be applied to not only

sporadic but also periodic tasks. We also present how to derive the optimal pattern for energy consumption minimization. Since no run-time calculation is required, our approach has less run-time scheduling overhead. In terms of performance, compared to the existing approach, our approach has 5 ~ 8% improvement for energy savings, i.e., the additional energy improvement is up to 38%, in our simulations. For future work, we would like to explore how to extend the pattern-based approach to multicore/multiprocessor systems.

REFERENCES

- [1] H. Aydin, R. Melhem, D. Mossé, and P. Mejía-Alvarez. Dynamic and aggressive scheduling techniques for power-aware real-time systems. In *RTSS*, pages 95–105, 2001.
- [2] N. Bansal, T. Kimbrel, and K. Pruhs. Speed scaling to manage energy and temperature. *J. ACM*, 54(1), 2007.
- [3] M. Bao, A. Andrei, P. Eles, and Z. Peng. Temperature-aware voltage selection for energy optimization. In *DATE*, pages 1083–1086, 2008.
- [4] D. Brooks and M. Martonosi. Dynamic thermal management for high-performance microprocessors. In *HPCA*, page 171, 2001.
- [5] T. Chantem, R. P. Dick, and X. S. Hu. Temperature-aware scheduling and assignment for hard real-time applications on MPSoCs. In *DATE*, 2008.
- [6] R. Jejurikar, C. Pereira, and R. Gupta. Leakage aware dynamic voltage scaling for real-time embedded systems. In *DAC*, pages 275–280, 2004.
- [7] W. Liao, L. He, and K. M. Lepak. Temperature and supply voltage aware performance and power modeling at microarchitecture level. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 24(7):1042–1053, 2005.
- [8] Y. Liu, R. P. Dick, L. Shang, and H. Yang. Accurate temperature-dependent integrated circuit leakage power estimation is easy. In *DATE*, pages 1526–1531, 2007.
- [9] S. Murali, A. Mutapcic, D. Aienza, R. Gupta, S. Boyd, and G. D. Micheli. Temperature-aware processor frequency assignment for mpsoCs using convex optimization. In *CODES+ISSS*, 2007.
- [10] K. Skadron, T. F. Abdelzaher, and M. R. Stan. Control-theoretic techniques and thermal-rc modeling for accurate and localized dynamic thermal management. In *Proc. of HPCA*, pages 17–28, 2002.
- [11] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture. In *ISCA*, pages 2–13, 2003.
- [12] L. Thiele, S. Chakraborty, and M. Naedele. Real-time calculus for scheduling hard real-time systems. In *ISCAS*, pages 101–104, 2000.
- [13] S. Wang and R. Bettati. Delay analysis in temperature-constrained hard real-time systems with general task arrivals. In *RTSS*, 2006.
- [14] F. Yao, A. Demers, and S. Shenker. A scheduling model for reduced CPU energy. In *FOCS*, pages 374–382. IEEE, 1995.
- [15] L. Yuan, S. Leventhal, and G. Qu. Temperature-aware leakage minimization technique for real-time systems. In *ICCAD*, pages 761–764, 2006.
- [16] S. Zhang and K. S. Chatha. Approximation algorithm for the temperature-aware scheduling problem. In *ICCAD*, 2007.
- [17] Y. Zhang, X. Hu, and D. Z. Chen. Task scheduling and voltage selection for energy minimization. In *DAC*, pages 183–188, 2002.