An Analytical Solution for Multi-Core Energy Calculation with Consideration of Leakage and Temperature Dependency

Ming Fan Vivek Chaturvedi Shi Sha Gang Quan
Electrical and Computer Engineering Department
Florida International University
Miami, FL, 33174
{mfan001, vchat001, ssha001, gaquan}@fiu.edu

***Abstract*—Energy minimization is a critical issue and challenge when considering the cyclic dependency of leakage power and temperature as IC technology reaches deep sub-micron level. In this paper, we present an analytical method to calculate the energy consumption efficiently and effectively for a given voltage schedule on a multi-core platform, with the leakage/temperature dependency taken into consideration. Our experiments show that the proposed method can achieve a speedup of 15 times compared with the numerical method, with a relative error of no more than 1.5%.**

***Keywords*—multi-core systems, energy calculation, leakage/temperature dependency**

I. Introduction

The increasing popularity of multi-core systems and the rising performance demand have made energy efficiency a critical design objective for system designers. Catalyzed by continuous transistor scaling, an exponential increase in transistor density for higher performance platforms has led to sharp rise in power/energy consumption [1], [2]. The continuously increased power consumption has resulted in soaring chip temperature, which adversely impacts the performance, reliability, and packaging/cooling costs [3]. More importantly, as design paradigm shifts to deep sub-micron domain, high chip temperature leads to a substantial increase in leakage power consumption [4]. For instance, Liao *et al.* [5] showed an increase in leakage power consumption by 38% with chip temperature rising from 65^0C to 110^0C. This signifies the need for incorporating leakage/temperature dependency into the system power model.

A key problem in energy efficiency design is calculating the energy consumption for a design alternative. Earlier research, e.g. [6], [7], has been exclusively focused on dynamic energy consumption. Some later research such as that in [8] takes the leakage power into consideration, but assumes that leakage power is constant. Under this assumption, the calculation of energy consumption for a given voltage schedule is trivial, since the overall power consumption remains the same as long as a system keeps the same running voltage and frequency. However, when considering the leakage/temperature dependency, the problem substantially becomes more challenging since the leakage power consumption (and thus the overall power consumption) varies with the temperature, and temperature changes with the power consumption as well. The energy calculation problem becomes even more complicated for multi-core platforms when the leakage power of one core depends not only on its own temperature, but also temperatures from other cores as well. As a result, many existing research on thermal and energy management (e.g. [9]) do not explicitly formulate the energy consumption.

To calculate the overall energy consumption with leakage/temperature dependency taken into consideration, one intuitive and commonly adopted approach is to use the numerical method. According to this method, the entire voltage schedule is split into a set of small time intervals such that within each interval the voltage/frequency and temperature of all cores can be regarded as invariant. The temperature and power trace, and thus the energy consumption, for a schedule can be obtained accordingly. For example, Liu *et al.* [10] formulated the energy minimization under a peak temperature as a non-linear programming problem, and then employed the above mentioned method to calculate the energy consumption. Bao *et al.* [11] also used a similar approach to keep track of temperature variations, and proposed an energy minimization method by dynamically selecting the supply voltage. One major problem with this approach is that the accuracy significantly depends on the variation rate of power and temperature. To achieve high accuracy, the length of the interval needs to be kept very small and thus the computation cost can be very high. Huang *et al.* [12] proposed a different approach to calculate the energy consumption. Based on leakage/temperature dependency model proposed in [13], they developed an analytical closed-form energy estimation method for a schedule. However, their work can only be applied for single core platforms but not for multi-core platforms, considering the heat transfer among different cores. We are not aware of any other technique published to analytically calculate the multi-core energy consumption with temperature/leakage dependency taken into consideration.

In this paper, we present a fast and accurate method to calculate the overall energy consumption for a given voltage schedule on multi-core platforms. Different from the traditional numerical method for energy calculation, we develop a closed-form analytical solution for the overall energy consumption of a given schedule. To the best of our knowledge, this is the first work that presents an analytical solution for

energy calculation on multi-core platforms by taking leakage/temperature dependency into consideration. In addition, our proposed method is rather general and fundamental, and thus can be applied for both homogeneous and heterogeneous multi-core platforms. Our experiments show that the proposed analytical method can achieve an average speedup of 15*X* with a relative error of no more than 1.5%.

The rest of this paper is organized as follows. We first introduce the system models used in this paper in Section II. We then introduce how to formulate temperature dynamics in Section III. Our proposed analytical solution of energy calculation for multi-core scheduling is presented in Section IV. We show our experimental results in Section V, and conclude this paper in Section VI.

II. Preliminary

A. Processing core and task model

The real-time system considered in this paper consists of M cores, denoted as $\mathbb{P} = \{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_M\}$. Each core has N *running modes*, each of which is characterized by a pair of parameters (v_k, f_k), where v_k and f_k are the supply voltage and working frequency under mode k, respectively.

Let $\mathbb{S}$ represent a *voltage schedule* or *speed schedule* which indicates how the supply voltage and working frequency is varied for each core at different times. We assume $\mathbb{S}$ is known. For example, $\mathbb{S}$ can be a design alternative during the design space exploration process, or an energy efficient solution based on a certain heuristic. In this paper, we use *voltage schedule* and *speed schedule* interchangeably. Let L be the schedule length of $\mathbb{S}$. We define the concept of *state interval* as below:

Definition 1: Given a speed schedule $\mathbb{S}$ for a multi-core system, an interval $[t_{q-1}, t_q]$ is called a *state interval* if each core runs only at one mode during that interval.

According to Definition 1, a speed schedule $\mathbb{S}$ essentially consists of a number of non-overlapped state intervals, i.e. Q state intervals, such that

1) $\bigcup_{q=1}^{Q}[t_{q-1}, t_q] = [0, L]$
2) $[t_{q-1}, t_q] \bigcap [t_{p-1}, t_p] = \emptyset$, if $q \neq p$

In addition, for a single state interval $[t_{q-1}, t_q]$, we use κ_q to denote the interval mode, which consists of the running modes of all cores in that interval, i.e. $\kappa_q = \{k_1, ..., k_M\}$ where k_i is the running mode of core $\mathcal{P}_i$ in that interval.

B. Power model

The overall power consumption (in *Watt*) is composed of dynamic power P_{dyn} and leakage power P_{leak}. In our power model, P_{dyn} is independent of the temperature, while P_{leak} is sensitive to both temperature and supply voltage. The dynamic power consumption is proportional to the square of supply voltage and linearity of working frequency [14]. In this paper, we assume that the working frequency is linearly proportional to supply voltage, thus the dynamic power consumption of core $\mathcal{P}_i$ can be formulated as [3], [12]

$$P_{dyn,i} = \gamma_{k_i} \cdot v_{k_i}^3 \tag{1}$$

where v_{k_i} is the supply voltage of core $\mathcal{P}_i$ and γ_{k_i} is a constant, both of which depend on the running mode of core $\mathcal{P}_i$, i.e. mode k_i.

While the circuit level study reveals a very complicated relation between leakage power and temperature, Liu *et al.* [15] found that a linear approximation of the leakage temperature dependency is fairly accurate. As such, similar to the work in [3], we approximate the leakage power of core $\mathcal{P}_i$ as follows

$$P_{leak,i} = \left(\alpha_{k_i} + \beta_{k_i} \cdot T_i(t)\right) \cdot v_{k_i} \tag{2}$$

where α_{k_i} and β_{k_i} are constants depending on the core running mode, i.e. mode k_i.

Consequently, the total power consumption of core $\mathcal{P}_i$ at time t, denoted as $P_i(t)$, can be formulated as:

$$P_i(t) = \left(\alpha_{k_i} + \beta_{k_i} \cdot T_i(t)\right) \cdot v_{k_i} + \gamma_{k_i} \cdot v_{k_i}^3 \tag{3}$$

For convenience in our presentation, we rewrite the above formula by separating the elements into temperature independent/dependent parts such that

$$P_i(t) = \psi_i + \phi_i \cdot T_i(t) \tag{4}$$

where

$$\psi_i = \alpha_{k_i} \cdot v_{k_i} + \gamma_{k_i} \cdot v_i^3 \tag{5}$$

$$\phi_i = \beta_{k_i} \cdot v_{k_i} \tag{6}$$

As such, the power consumption for a multi-core system can be represented as

$$\begin{bmatrix} P_1(t) \\ \vdots \\ P_M(t) \end{bmatrix} = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_M \end{bmatrix} + \begin{bmatrix} \phi_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \phi_M \end{bmatrix} \begin{bmatrix} T_1(t) \\ \vdots \\ T_M(t) \end{bmatrix} \tag{7}$$

or

$$\mathbf{P}(t) = \mathbf{\Psi} + \mathbf{\Phi}\mathbf{T}(t) \tag{8}$$

In our paper, we use the bold text for a vector/matrix and the unbolded text for a value, e.g. $\mathbf{T}$ represents a temperature vector while T represents a temperature value.

C. Thermal model

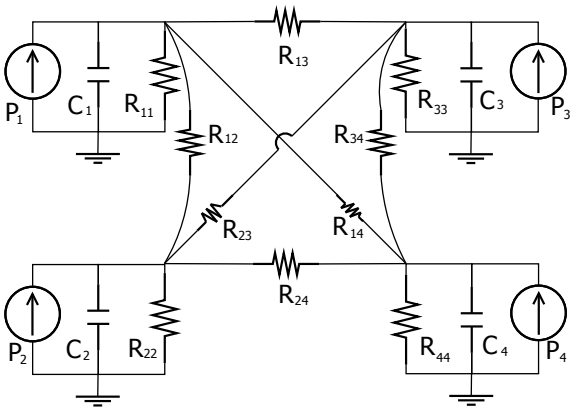


Fig. 1. Illustration for thermal phenomena on multi-core system

The thermal model used in this paper is similar to the one used in related researche [9], [16]. Figure 1 illustrates the thermal model for a 4-core system. C_i and R_{ij} denote the thermal capacitance (in $Watt/^oC$) of core $\mathcal{P}_i$ and the thermal resistance (in $J/^oC$) between core $\mathcal{P}_i$ and $\mathcal{P}_j$, respectively. Let T_{amb} denote the ambient temperature, then in general, the thermal phenomena of core $\mathcal{P}_i$ can be formulated as

$$C_i \cdot \frac{dT_i(t)}{dt} + \frac{T_i(t) - T_{amb}}{R_{ii}} + \sum_{j \neq i} \frac{T_i(t) - T_j(t)}{R_{ij}} = P_i(t) \tag{9}$$

Let $\delta_i = \frac{T_{amb}}{R_{ii}}$ and

$$g_{ij} = \begin{cases} \sum_{j=1}^{M} \frac{1}{R_{ij}}, & \text{if } \ j = i \\ \frac{-1}{R_{ij}}, & \text{otherwise} \end{cases} \tag{10}$$

Then the thermal model in equation (9) can be rewritten as

$$C_i \cdot \frac{dT_i(t)}{dt} + \sum_{j=1}^{M} g_{ij} \cdot T_j(t) = P_i(t) + \delta_i \tag{11}$$

Accordingly, for the entire system, the thermal model can be represented as

$$\mathbf{C}\frac{d\mathbf{T}(t)}{dt} + \mathbf{g}\mathbf{T}(t) = \mathbf{P}(t) + \boldsymbol{\delta} \tag{12}$$

where

$$\mathbf{C} = \begin{bmatrix} C_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_M \end{bmatrix} \mathbf{g} = \begin{bmatrix} g_{11} & \cdots & g_{1M} \\ \vdots & \ddots & \vdots \\ g_{M1} & \cdots & g_{MM} \end{bmatrix} \boldsymbol{\delta} = \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_M \end{bmatrix} \tag{13}$$

Note that $\mathbf{C}$, $\mathbf{g}$ and $\boldsymbol{\delta}$ are all constants that only depend on the multi-core architecture, i.e. capacitance and/or conductance. It is worth mentioning that our thermal model is very general and accounts for the heat transfer impacts among different cores. It can be used for thermal analysis for both the temperature transient states as well as the temperature stable state.

III. Temperature formulation

Our goal is to formulate the overall energy consumption for a given voltage schedule. Before we introduce our method, in this section, we first present how to formulate the temperature dynamics on multi-core systems analytically.

Note that, by applying the power model (see equation (8)) into the thermal model (see equation (12)), we can directly obtain that

$$\mathbf{C}\frac{d\mathbf{T}(t)}{dt} + \mathbf{g}\mathbf{T}(t) = \boldsymbol{\Psi} + \boldsymbol{\Phi}\mathbf{T}(t) + \boldsymbol{\delta} \tag{14}$$

Let $\mathbf{G} = \mathbf{g} - \boldsymbol{\Phi}$, then the above equation can be rewritten as

$$\mathbf{C}\frac{d\mathbf{T}(t)}{dt} + \mathbf{G}\mathbf{T}(t) = \boldsymbol{\Psi} + \boldsymbol{\delta} \tag{15}$$

Since $\mathbf{C}$ is the capacitance matrix with no zero values only on the diagonal, we know $\mathbf{C}$ is nonsingular. Thus, the inverse of $\mathbf{C}$, i.e. $\mathbf{C}^{-1}$ exists. Then equation (15) can be further represented as

$$\frac{d\mathbf{T}(t)}{dt} = \mathbf{A}\mathbf{T}(t) + \mathbf{B} \tag{16}$$

where $\mathbf{A} = -\mathbf{C}^{-1}\mathbf{G}$ and $\mathbf{B} = \mathbf{C}^{-1}(\boldsymbol{\Psi} + \boldsymbol{\delta})$. The system thermal model shown in equation (16) has a form of first order *Ordinary Differential Equations* (ODE), which has the following solution under constant coefficients:

$$\mathbf{T}(t) = e^{t\mathbf{A}}\mathbf{T}_0 + \mathbf{A}^{-1}(e^{t\mathbf{A}} - \mathbf{I})\mathbf{B} \tag{17}$$

where $\mathbf{T}_0$ is the initial temperature.

Specifically, for a state interval $[t_{q-1}, t_q]$, and let κ_q be the corresponding interval mode, once the temperates at the starting point, i.e. $\mathbf{T}(t_{q-1})$, are given, according to equation (17), the ending temperatures of that interval, i.e. $\mathbf{T}(t_{q-1})$, can be directly formulated as

$$\mathbf{T}(t_q) = e^{\Delta t_q \mathbf{A}_{\kappa_q}}\mathbf{T}(t_{q-1}) + \mathbf{A}_{\kappa_q}^{-1}(e^{\Delta t_q \mathbf{A}_{\kappa_q}} - \mathbf{I})\mathbf{B}_{\kappa_q} \tag{18}$$

where $\mathbf{A}_{\kappa_q} = -\mathbf{C}^{-1}\mathbf{G}_{\kappa_q}$, $\mathbf{B}_{\kappa_q} = \mathbf{C}^{-1}(\boldsymbol{\Psi}_{\kappa_q} + \boldsymbol{\delta})$, and $\Delta t_q = t_q - t_{q-1}$. Note that since $\mathbf{A}_{\kappa_q}$ and $\mathbf{B}_{\kappa_q}$ are only dependent on the core running modes, i.e. κ_q, within a state interval $[t_{q-1}, t_q]$, both $\mathbf{A}_{\kappa_q}$ and $\mathbf{B}_{\kappa_q}$ are constant.

Consequently, given a speed schedule $\mathbb{S}$ and the corresponding initial temperature $\mathbf{T}(0)$, with the method introduced above, we can obtain the temperature traces of $\mathbb{S}$ by successively calculating the temperature from one state interval to another.

IV. Energy formulation for multi-core systems

With the temperature formulation introduced as above, we are now ready to discuss our method to formulate the energy consumption on multi-core systems considering the interdependence of leakage power and temperature. In what follows, we first present an analytical solution to calculated the energy consumption for one state interval. Then we formulate the total energy consumption for the entire speed schedule.

Consider a state interval, i.e. $[t_{q-1}, t_q]$ with initial temperature of $\mathbf{T}(t_{q-1})$. The energy consumption of all cores within that interval can be simply formulated as

$$\mathbf{E}(t_{q-1}, t_q) = \int_{t_{q-1}}^{t_q} \mathbf{P}(t)dt \tag{19}$$

Based on our system power model, given by equation (8), we have

$$\mathbf{E}(t_{q-1}, t_q) = \Delta t_q \boldsymbol{\Psi} + \boldsymbol{\Phi} \int_{t_{q-1}}^{t_q} \mathbf{T}(t)dt \tag{20}$$

For a given state interval and multi-core platform, according to equation (5) and (8), $\boldsymbol{\Psi}$ is a constant. Therefore, to calculate $\mathbf{E}(t_{q-1}, t_q)$, we only need to get $\int_{t_{q-1}}^{t_q} \mathbf{T}(t)dt$.

Recall that the analytical solution for $\mathbf{T}(t)$ is given by equation (17). One intuitive approach is therefore to find $\int_{t_{q-1}}^{t_q} \mathbf{T}(t)dt$ as follows:

$$\begin{aligned} & \int_{t_{q-1}}^{t_q} \mathbf{T}(t)dt \\ = & \int_{t_{q-1}}^{t_q} \big(e^{t\mathbf{A}}\mathbf{T}(t_{q-1}) + \mathbf{A}^{-1}(e^{t\mathbf{A}} - \mathbf{I})\mathbf{B}\big)dt & (21) \\ = & \int_{t_{q-1}}^{t_q} e^{t\mathbf{A}}dt\mathbf{T}(t_{q-1}) + \mathbf{A}^{-1}\big(\int_{t_{q-1}}^{t_q} e^{t\mathbf{A}}dt - t\mathbf{I}\big)\mathbf{B} & (22) \end{aligned}$$

The problem of this approach is that we need to find $\int_{t_{q-1}}^{t_q} e^{t\mathbf{\Lambda}}dt$, but unfortunately, we are not aware of any existing method or mathematical tools that can be used to solve the problem of exponential matrix integration. Therefore, to replace $\mathbf{T}(t)$ in equation (20) with equation (17) does not seem to be a promising approach.

Note that, as long as we can get $\int_{t_{q-1}}^{t_q} \mathbf{T}(t)dt$, we find the solution to the overall energy consumption for state interval $[t_{q-1}, t_q]$. If we let $\mathbf{X} = \int_{t_{q-1}}^{t_q} \mathbf{T}(t)dt$, then the above can be simplified as

$$\mathbf{E}(t_{q-1}, t_q) = \Delta t_q \mathbf{\Psi} + \mathbf{\Phi X} \tag{23}$$

In what follows, we introduce a novel method to calculate $\mathbf{X}$. Recall that the system thermal model can be formulated as (see equation (15)):

$$\mathbf{C}\frac{d\mathbf{T}(t)}{dt} + \mathbf{G}\mathbf{T}(t) = \mathbf{\Psi} + \boldsymbol{\delta}$$

Since $\mathbf{C}, \mathbf{G}, \mathbf{\Psi}$ and $\boldsymbol{\delta}$ are all constants within interval $[t_{q-1}, t_q]$, if we integrate on both sides of the above equation with respect to time t, where $t \in [t_{q-1}, t_q]$, we have

$$\mathbf{C}\Delta\mathbf{T}_q + \mathbf{G}\int_{t_{q-1}}^{t_q} \mathbf{T}(t)dt = \Delta t_q(\mathbf{\Psi} + \boldsymbol{\delta}) \tag{24}$$

where $\Delta\mathbf{T}_q = \mathbf{T}(t_q) - \mathbf{T}(t_{q-1})$ and $\Delta t_q = t_q - t_{q-1}$. If we further replace $\int_{t_{q-1}}^{t_q} \mathbf{T}(t)dt$ with $\mathbf{X}$, we have

$$\mathbf{C}\Delta\mathbf{T}_q + \mathbf{G}\mathbf{X} = \Delta t_q(\mathbf{\Psi} + \boldsymbol{\delta}) \tag{25}$$

Now let $\mathbf{H}$ be that

$$\mathbf{H} = \Delta t_q(\mathbf{\Psi} + \boldsymbol{\delta}) - \mathbf{C}\Delta\mathbf{T}_q \tag{26}$$

Note that, based on equation (18), $\Delta\mathbf{T}_q$ can be easily calculated as

$$\Delta\mathbf{T}_q = \mathbf{T}(t_q) - \mathbf{T}(t_{q-1}). \tag{27}$$

Therefore, $\mathbf{H}$ can be easily obtained once the state interval $[t_{q-1}, t_q]$ is defined. Accordingly, from equation (25), we can get

$$\mathbf{G}\mathbf{X} = \mathbf{H} \tag{28}$$

Assuming $\mathbf{G}$ is nonsingular, $\mathbf{X}$ can thus be solved as

$$\mathbf{X} = \mathbf{G}^{-1}\mathbf{H} \tag{29}$$

By applying equation (29) into (23), we can get that

$$\mathbf{E}(t_{q-1}, t_q) = \Delta t_q \mathbf{\Psi} + \mathbf{\Phi}\mathbf{G}^{-1}\mathbf{H} \tag{30}$$

As such, given a multi-core platform and a state interval, the energy consumption within the interval can be calculated using equation 30 analytically. We formally present our energy calculation method for a state interval in Theorem 1.

Theorem 1: Given a state interval $[t_{q-1}, t_q] \in \mathbb{S}$ with $\mathbf{T}_{q-1}$ the temperature at time t_{q-1}, the overall system energy consumption within interval $[t_{q-1}, t_q]$ can be formulated as

$$\mathbf{E}(t_{q-1}, t_q) = \Delta t_q \mathbf{\Psi}_{\kappa_q} + \mathbf{\Phi}_{\kappa_q}\mathbf{G}_{\kappa_q}^{-1}\mathbf{H}_{\kappa_q} \tag{31}$$

Note that given a speed schedule and initial temperature, the temperature at the ends of each state interval can be readily determined using equation (18). For a speed schedule $\mathbb{S}$ consisting of Q state intervals, the total system energy consumption under $\mathbb{S}$ can be obtained by summing up the energy consumptions of all state intervals. We conclude this energy calculation method in Theorem 2.

Theorem 2: Given an initial temperature T_0 and a speed schedule $\mathbb{S}$ consisting of Q state intervals, the total system energy consumption under $\mathbb{S}$, denoted as $E_{total}(\mathbb{S})$, can be calculated as

$$E_{total}(\mathbb{S}) = \sum_{q=1}^{Q}\sum_{i=1}^{M} E_i(t_{q-1}, t_q) \tag{32}$$

where $E_i(t_{q-1}, t_q)$ can be calculated from equation (31).

The computational complexity for our energy calculation of each state interval mainly comes from the matrix multiplications and inversions, with a complexity of $O(M^3)$. To calculate the overall energy consumption for a schedule with Q state intervals, the complexity is thus $O(Q \times M^3)$. In what follows, we use experiments to evaluate the performance of our proposed method.

V. Experimental validation

In this section, we validated the proposed energy calculation method with simulations. We compared our proposed method with the traditional *numerical method* to obtain some insights with regard to the effectiveness and efficiency of an energy estimation approach. In what follows, we first introduce the settings for our experiments. We then present and discuss the experimental results.

A. Experimental set-up

TABLE I
HotSpot Parameters and Floorplan

Parameter	*Value*
Total Cores	9 (3x3)
Area per Core	4 mm^2
Die Thickness	0.15 mm
Heat Spreader Side	20 mm
Heat Sink Side	30 mm
Convection Resistance	0.1 K/W
Convection Capacitance	140 J/K
Ambient Temperature	30^oC

TABLE II
Power/thermal parameters

$V_{dd}(V)$	α	β	γ
0.0	0.0	0.0	0.0
0.8	1.4533	0.0760	6.0531
0.9	2.4173	0.0844	5.8008
1.0	4.0533	0.0936	5.8906

We performed our experimental simulations based on a 3×3 multi-core system. The granularity of the floorplan was restricted to core-level. Our core model was based on $65nm$ technology as presented in [5]. We assumed that each core supports 3 active modes with supply voltage ranging

from $0.8V$ to $1.0V$ and a step size of $0.1V$. We also set one inactive/sleep mode with supply voltage equal to $0V$.

We adopted the same thermal parameters as used in work [3] (see Table V-A). We set the power consumption under the peak temperature constraint of 110^0C. The thermal parameters, including thermal conductance, capacitance etc. were taken from HotSpot-4.02 [17]. The thermal nodes in our thermal model included active layer, interface layer, heat spreader and heat sink. The relevant useful parameters were shown in Table V-A. We set the ambient temperature T_{amb} as well as the initial temperature T_0 as 30^oC.

We randomly generated 50 multi-core speed schedules as our test cases. The running mode for each scheduling interval was randomly chosen from $[0, 0.8, 0.9, 1.0]$V (see Table V-A). The total length of the scheduling interval was evenly distributed within $[100, 200]$, and the length of each scheduling interval was evenly distributed within $[30, 50]$. For each test case, our proposed method as well as the traditional numerical method with sampling interval length varied from 0.5 seconds to 3.0 seconds were used to calculate the energy consumption. The baseline was obtained by setting the length of sampling interval to 0.01. When applying the numerical method, we calculated the leakage power consumption based on the accurate circuit level leakage temperature model [5], i.e.

$$I_{leak} = I_s \cdot (\mathcal{A} \cdot T^2 \cdot e^{((a \cdot V_{dd}+b)/T)} + \mathcal{B} \cdot e^{(c \cdot V_{dd}+d)}) \qquad (33)$$

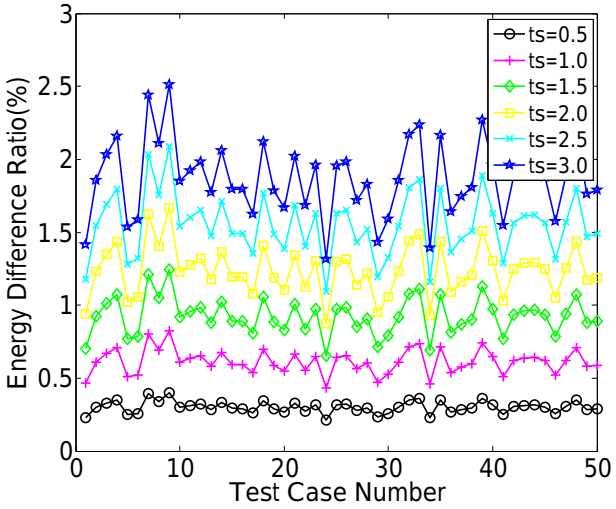
where I_s is the leakage current at a certain reference temperature and supply voltage, T is the core temperature, and $\mathcal{A}, \mathcal{B}, a, b, c, d$ are physically determined constants (i.e. fitting parameters). All simulations were conducted on a *Dell Precision T1500 Desktop Workstation* with CPU type of *Intel i5 750 Quad Core* and $4GB$ memory capacity.

B. Accuracy analysis

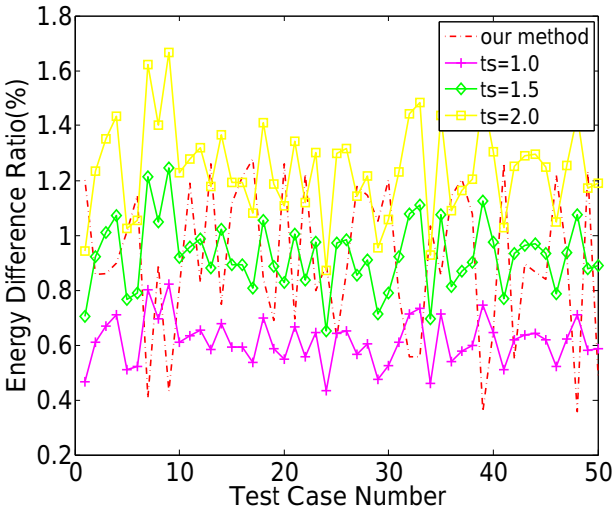
In this subsection, we investigate the performance of our proposed method in terms of accuracy. Note that, while we analytically developed the energy formulation as shown in equation (31) and (32), its accuracy is contingent upon our leakage/temperature dependency assumption as listed in equation (2).

To compare the accuracy of different energy estimation approaches, we need to identify the accurate energy consumption for a given speed schedule. We resorted to the numerical method with a very short sampling interval to achieve this goal. The question is how short the sampling interval should be.

In our experiments, we set the length of sampling interval t_s from 0.5 seconds to 3.0 seconds with a step length of 0.5 seconds and calculated the energy consumption for different schedules. Particularly, we set $t_s = 0.01$ second as the baseline since we found that the largest relative energy difference between $t_s = 0.01$ second and $t_s = 0.5$ seconds was smaller than 0.4%. We then normalized the energy consumption by other approaches to the baseline results. Figure 2(a) shows the relative differences of energy consumption estimation results using a numerical approach with different sampling intervals, i.e. from $t_s = 0.5$ seconds to $t_s = 3.0$ seconds. The relative differences of energy consumption based on our proposed approach and comparable numerical results are presented in Figure 2(b).



(a) Numerical method



(b) Our proposed method

Fig. 2. Accuracy analysis, compared with the numerical method under $t_s = 0.01$

From Figure 2(a), it is not surprising to see that the smaller the sampling interval, the smaller the energy difference ratio becomes. For example, when t_s is decreased from 3.0 to 0.5, the average energy difference ratio is reduced from 1.7% to 0.4%. This is because the smaller the sampling interval is, the less the temperature can change. Since the numerical method estimates the leakage consumption within an interval assuming temperature within a sampling interval does not change, the error of the estimated leakage energy can be kept small if the sampling interval is small enough.

On the other hand, we can see from Figure 2(b) that our proposed method performed well from the aspect of accuracy. For example, the largest relative error observed in Figure 2(b) is no more than 1.5%. As shown in Figure 2(b), we can see that our method outperforms the numerical method with $t_s = 2.0$ seconds for most test cases, and compatible with

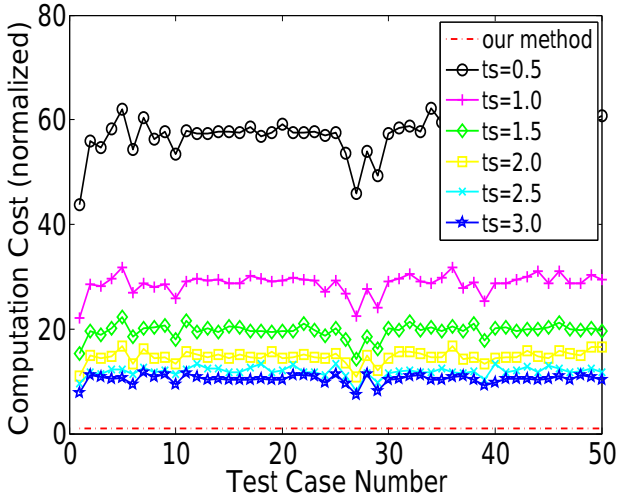


Fig. 3. Time efficiency analysis, normalized with our method

the method with $t_s = 1.5$ seconds. The experimental results clearly show that our proposed approach can achieve very good accuracy in estimating the overall energy consumption for a given speed schedule.

C. Time efficiency analysis

We next want to evaluate the computational efficiency of our proposed method. We collected the CPU times for different approaches for all test cases. We then used the CPU times of our method as the baseline results. The normalized results are shown in Figure 3.

From Figure 3, we can see that the numerical method with a small sampling interval can have a substantially large computational overhead than our approach. For example, as shown in Figure 3, our method is more than 50 times (on average) faster than the numerical approach with $t_s = 0.5$, and 10 times (on average) faster than that with $t_s = 3.0$. Compared with the numerical method with $t_s = 1.5$, which is compatible with our method from the perspective of accuracy, our method can achieve an average speedup of 15 times. Note that, the computational complexity of our approach is determined only by the number of state intervals in a speed schedule, while the complexity of the numerical approach depends on both the schedule length (L) and sampling interval (t_s). As shown in Figure 2(a), in order to achieve high accuracy, the sampling interval must be very small and thus very timing consuming. From Figure 3, we can conclude that the proposed method is much more time efficient than the numerical approach.

VI. Conclusion

Energy consumption optimization is a critical design issue in the design of multi-core computing systems. It becomes more challenging in deep submicron domains when leakage consumption becomes more and more significant and the interdependency of leakage and temperature becomes substantial. A key to solve this problem is to calculate the energy consumption efficiently and effectively.

In this paper, we presented a fast and accurate solution for energy calculation on multi-core systems that takes the interdependency of leakage, temperature and supply voltage into consideration. Different from the traditional numerical approach, we developed an analytical formulation for the energy consumption, and based on which, to calculate the overall energy consumption rapidly and accurately. Our system models are rather general and can be easily extended for different platforms and applications. Our experiments showed that the proposed method can achieve a speedup of 15 times compared with the numerical method, with a relative error no more than 1.5%.

Acknowledgement

This work is supported in part by NSF under projects CNS-0969013, CNS-0917021, and CNS-1018108.

References

[1] 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 (DAC), 46th ACM/IEEE*, Jul. 2009, pp. 490 –495.

[2] S. Borkar, "Thousand core chips: a technology perspective," in *Design Automation Conference (DAC), 44th ACM*. New York, NY, USA: ACM, 2007, pp. 746–749. [Online]. Available: http://doi.acm.org/10.1145/1278480.1278667

[3] G. Quan and V. Chaturvedi, "Feasibility analysis for temperature-constraint hard real-time periodic tasks," *Industrial Informatics, IEEE Transactions on*, vol. 6, no. 3, pp. 329 –339, Aug. 2010.

[4] ITRS, *International Technology Roadmap for Semiconductors (2011 Edition)*. Austin, TX.: International SEMATECH, http://public.itrs.net/.

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

[6] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced cpu energy," in *Foundations of Computer Science(FOCS), 36th Annual Symposium on*, 1995, pp. 374–382.

[7] C.-H. Lee and K. Shin, "On-line dynamic voltage scaling for hard real-time systems using the edf algorithm," in *Real-Time Systems Symposium (RTSS), 25th IEEE International*, Dec. 2004, pp. 319–335.

[8] R. Jejurikar and R. Gupta, "Dynamic slack reclamation with procrastination scheduling in real-time embedded systems," in *Design Automation Conference (DAC), 42nd IEEE*, Jun. 2005, pp. 111–116.

[9] S. Sharifi, R. Ayoub, and T. Rosing, "Tempomp: Integrated prediction and management of temperature in heterogeneous mpsocs," in *Design, Automation Test in Europe(DATE)*, Mar. 2012, pp. 593–598.

[10] Y. Liu, H. Yang, R. Dick, H. Wang, and L. Shang, "Thermal vs energy optimization for dvfs-enabled processors in embedded systems," in *Quality Electronic Design (ISQED), 8th International Symposium on*, Mar. 2007, pp. 204–209.

[11] M. Bao, A. Andrei, P. Eles, and Z. Peng, "Temperature-aware voltage selection for energy optimization," in *Design, Automation and Test in Europe (DATE)*, 2008, pp. 1083–1086.

[12] H. Huang and G. Quan, "Leakage aware energy minimization for real-time systems under the maximum temperature constraint," in *Design, Automation Test in Europe(DATE)*, Mar. 2011, pp. 1–6.

[13] G. Quan and Y. Zhang, "Leakage aware feasibility analysis for temperature-constrained hard real-time periodic tasks," in *Real-Time Systems (ECRTS), 21st Euromicro Conference on*, Jul. 2009, pp. 207–216.

[14] J. Rabaey, A. Chandrakasan, and B. Nikolic, "Digital integrated circuits: A design perspective," in *Englewood Cliffs, NJ: Prentice-Hall*, 2003.

[15] L. Yongpan and Y. Huazhong, "Temperature-aware leakage estimation using piecewise linear power models," *IEICE Transactions on Electronics*, vol. 93, no. 12, pp. 1679–1691, 2010.

[16] I. Ukhov, M. Bao, P. Eles, and Z. Peng, "Steady-state dynamic temperature analysis and reliability optimization for embedded multiprocessor systems," in *Design Automation Conference (DAC), 49th ACM/EDAC/IEEE*, Jun. 2012, pp. 197–204.

[17] "Hotspot 4.2 temperature modeling tool," *University of Virgina*, p. http://lava.cs.virginia.edu/HotSpot, 2009.