IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, VOL. 32, NO. 5, MAY 2013 695

Accurate Modeling of the Delay and Energy Overhead of Dynamic Voltage and Frequency Scaling in Modern Microprocessors

Sangyoung Park, Student Member, IEEE, Jaehyun Park, Student Member, IEEE, Donghwa Shin, Student Member, IEEE, Yanzhi Wang, Student Member, IEEE, Qing Xie, Student Member, IEEE, Massoud Pedram, Fellow, IEEE, and Naehyuck Chang, Fellow, IEEE

Abstract—Dynamic voltage and frequency scaling (DVFS) has been studied for well over a decade. Nevertheless, existing DVFS transition overhead models suffer from significant inaccuracies; for example, by incorrectly accounting for the effect of DC–DC converters, frequency synthesizers, voltage, and frequency change policies on energy losses incurred during mode transitions. Incorrect and/or inaccurate DVFS transition overhead models prevent one from determining the precise break-even time and thus forfeit some of the energy saving that is ideally achievable. This paper introduces accurate DVFS transition overhead models for both energy consumption and delay. In particular, we redefine the DVFS transition overhead including the underclocking-related losses in a DVFS-enabled microprocessor, additional inductor IR losses, and power losses due to discontinuous-mode DC–DC conversion. We report the transition overheads for a desktop, a mobile and a low-power representative processor. We also present DVFS transition overhead macromodel for use by high-level DVFS schedulers.

Index Terms—Delay and energy overhead, dynamic voltage and frequency scaling (DVFS), macromodel.

I. INTRODUCTION

DYNAMIC voltage and frequency scaling (DVFS) has proved itself as one of the most successful energy saving techniques for a wide range of processors. DVFS is enabled by a programmable DC–DC converter and a programmable clock generator. These devices naturally incur overhead whenever the system changes its voltage and frequency setting. Since the DVFS break-even time is strongly dependent on the DVFS transition overhead [2], correct overhead estimation is crucial in achieving the maximum DVFS benefit.

Manuscript received May 2, 2012; revised September 24, 2012; accepted November 14, 2012. Date of current version April 17, 2013. This work was supported under a grant from the Brain Korea 21 Project, and the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MEST), under Grant 2011-0005640. The IET at Seoul National University provided research facilities for this work. Preliminary version of this paper have been presented in part in the International Symposium on Low Power Electronics and Design (ISLPED) 2010 [1]. This paper was recommended by Associate Editor I. Corradella.

S. Park, J. Park, and N. Chang are with Seoul National University, Seoul 151-744, Korea (e-mail: stpark@elpl.snu.ac.kr; jpark@elpl.snu.ac.kr; naehyuck@elpl.snu.ac.kr).
D. Shin is with the Politecnico di Torino, Turin 10129, Italy (e-mail: donghwa.shin@polito.it).
Y. Wang, Q. Xie, and M. Pedram are with the University of Southern California, Los Angeles, CA 90089-2562 USA (e-mail: yanx@usc.edu; xqing@usc.edu; pedram@usc.edu).
Digital Object Identifier 10.1109/TCAD.2012.2235512

DVFS transition overhead may be negligible or significant depending on how often we change the DVFS setting. Modern microprocessors tend to change their DVFS setting rather frequently in response to rapid changes in the application behavior. In addition, DVFS is widely used for dynamic thermal management (DTM), which requires frequent change of the DVFS setting (such as in a millisecond) to achieve thermal stability. Incorrect DVFS transition overhead may cause failure in the thermal stability of the system. Correct modeling of the DVFS transition overhead is not a trivial undertaking since it requires detailed understanding of the DC–DC converter, frequency synthesizer, voltage and frequency transition policies, and so on.

Unfortunately, existing DVFS transition overhead models have limitations and are not applicable to modern DVFS setups. In particular, they are significantly simplified, contain technical fallacies, or are limited to uncommon setups. In fact, among the 120 DVFS-related papers published in the last 10 years, only 17% of the DVFS papers have considered the transition overhead. The majority of DVFS studies simply ignore the transition overhead [3]–[5]. Among the 17% of DVFS papers, 75% of papers are based on the analytical transition overhead models introduced in [6] and [7]. Some of the previous work (e.g. [6], [8], and [9]) assume voltage controlled oscillators for the clock generator, which is unusual in today’s microprocessors (or even in embedded microcontrollers). Surprisingly, more than a few prior work references have assumed that the microprocessor stops operation during the entire voltage transition period, something that is neither desirable nor practical [10]. Most of all, a majority of the prior art papers consider a DVFS transition overhead model that is based on incorrect assumptions. A recent work has raised this problem and suggested the correct definition of DVFS transitions [1]. Evidently, there is a strong need to construct a correct DVFS transition overhead model because even recent DVFS work is still based on the previous models as shown in Section III.

In this paper, we provide a formal definition of the DVFS transition overhead, analyze various components of the overhead, and finally construct a macromodel for DVFS transition overhead.

This paper takes into account all the major power and performance loss components in the modern DVFS setups as...
IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, VOL. 32, NO. 5, MAY 2013

696

Fig. 1. Breakdown of an upscaling DVFS transition overhead in energy and delay. (Intel Core2 Duo E6800, upscaling from 1.05 to 1.3-V): (a) Energy overhead. (b) Delay overhead.

(a)
(b)

Conventional model
(Bulk capacitor charging)

Additional IR
18.5%

Underclocking
18.8%

PLL lock time
7.8%

Proposed model

Fig. 2. Upscaling and downscaling current flows.

Fig. 3. Concept of underclocking loss and PLL lock time loss in a DVFS transition.

scaling as shown in Fig. 3, which is a major source of the delay overhead [92.4% of total delay overhead as shown in Fig. 1(b)].

5) The microprocessor consumes static power even during the PLL lock time though it halts. This is another source of energy waste [3.2% of total energy overhead as shown in Fig. 1(a)].

The aforesaid observations are the key contributions of this paper, based on which we derive accurate, yet compact energy and delay overhead models for DVFS transitions. We present a relatively simple analytical model with parameters that can be easily acquired from the datasheets and/or passive component values (R, L, and C values). We also provide case studies for three distinct and representative microprocessors, Intel Core2 Duo E6800, Samsung Exynos 4210, and TI MSP430. Some programmers who have no hardware knowledge may use the numbers. Finally, we emphasize the importance of considering the DVFS transition overhead for a DTM example.

II. BACKGROUND

A. System Setup for DVFS

DVFS setups require a programmable voltage regulator and a programmable clock generator. A microprocessor is generally powered by a buck-type switching-mode DC-DC converter as shown in Fig. 2. The upper and lower MOSFETs control the inductor current and the output voltage. The inductor current never changes abruptly, which results in adiabatic charging and discharging to and from the bulk capacitor. In other words, the bulk capacitor is not subject to switching loss that is proportional to the square of the terminal voltage.

DC-DC converters are subject to power loss aside from the DVFS transition overhead, which includes conduction loss, switching loss, and controller loss [11]. Conduction loss is the IR loss in the MOSFET and inductor given by

\[ P_{\text{con}} = I_G^2 \left( D \cdot R_{SW1} + (1 - D) \cdot R_{SW2} + R_L \right) \]

\[ + \frac{1}{3} \left( \frac{\Delta I}{L} \right)^2 \left( D \cdot R_{SW1} + (1 - D) \cdot R_{SW2} + R_L \right) \]

(1)

where \( I_G \), \( D \), \( R_{SW1} \), \( R_{SW2} \), \( R_L \), \( R_C \) are output current, duty ratio, upper switch on-resistance, lower switch on-resistance, inductor resistance, and capacitor ESR, respectively. Switching loss is the power dissipation due to gate drive which is given by

\[ P_{\text{sw}} = V_T \cdot f_s \cdot (Q_{SW1} + Q_{SW2}) \]

(2)

where \( V_T \), \( f_s \), \( Q_{SW1} \), and \( Q_{SW2} \) are input voltage, switching frequency, gate charges of the upper and lower MOSFET,
respectively. Controller loss is power loss in the PWM/PPM controller that is independent of the load condition.

B. DC–DC Converter Control Methods

Many modern switching power supplies perform pulse width modulation (PWM) and use either voltage- or current-mode control to regulate the output voltage level.

Current-mode control is usually used in modern switching regulator designs to overcome the disadvantages of voltage-mode control [12]. The error in voltage directly reflected in the peak switching current. Fast response time is achieved by direct inductor current sensing. Modeling the behavior of current-mode controlled DC–DC converters is not a trivial task because this type of converter exhibits highly nonlinear characteristics. There are large-signal models for PWM controlled DC–DC converters capable of modeling dynamic behavior such as state-space averaging models [13], [14], but it is still very hard to analyze the feedback loop and obtain a closed-form solution of the output voltage and current over time. Term \( i_L \) in [13] and \( d \) in [14] have to be found in a closed form to calculate the trace of output voltage and current.

DC–DC converters for low-power applications usually adopt pulse-frequency modulation (PFM) since the PFM method exhibits higher efficiency with light load. We restrict the modeling to the buck type DC–DC converter with peak current-mode PWM control and PFM control throughout this paper, which is the most general setup for microprocessor systems.

C. Voltage Transition Sequences in Continuous and Discontinuous Modes

1) Upscaling Transition Sequence Using Continuous and Discontinuous Mode DC–DC Converters: Upscaling stands for increasing the supply voltage and clock frequency. The microprocessor sets a new (voltage identifier) VIID to make the DC–DC converter generate a higher output voltage that increases the duty ratio of the upper MOSFET. This increases the inductor current and eventually increases the bulk capacitor voltage.

Voltage upscaling pumps more charge into the bulk capacitor by increasing \( I_L(t) \). Fig. 4 illustrates an SPICE simulation of an upscaling transition of Intel Core2 Duo E6850 processor using LTSPICE [15]. The shaded area denotes the amount of additional charge transferred to the bulk capacitor during upscaling. Briefly, higher transient \( I_L(t) \) larger than 60 A flows through the inductor while normal operating \( I_L(t) \) is approximately 30 A.

2) Discontinuous Mode DC–DC Converters: Fig. 6 illustrates that a discontinuous-mode downscaling transition turns off the lower MOSFET as soon as the inductor current becomes negative, which prevents the bulk capacitor from further discharging. Instead, \( I_{dL}(t) \) discharges the bulk capacitor and makes the DC–DC converter output voltage converge to \( V_L \). Downscaling takes longer to stabilize in the discontinuous mode compared to the continuous mode because only \( I_{dL}(t) \) discharges the bulk capacitor. Discontinuous mode plays an important role in the modern DC–DC converter design by maintaining high conversion efficiency even when the load current is light. Modern DVFS setups prefer to use discontinuous-mode for more efficient use of the stored energy in the bulk capacitor.
D. Clock Frequency Transition and Underclocking

The relationship between the supply voltage and clock frequency is approximately explained by the alpha power-law [16]. Early DVFS works assume a voltage-controlled oscillator (VCO) for the clock generator [6]. The VCO performs automatic and continuous frequency change according to the transient voltage. The gradual frequency change allows the microprocessor to keep operating during the entire voltage transition period. However, VCOs are not commonly used in typical high-performance microprocessor systems due to their unstable and imprecise clock frequency output except low-performance microcontrollers running at around a few megahertz, such as TI MSP430.

On the other hand, PLLs are widely used for the programmable clock generators thanks to the accuracy of the frequency setting. Upscaling first attempts voltage change and waits until the voltage is stabilized. Once the voltage is stabilized, the microprocessor changes the PLL setting (see Fig. 7). This ensures a safe operation of the microprocessor even while the supply voltage is changing. The microprocessor, however, stops operating during the PLL lock time. Downscaling is the opposite; we change the PLL setting first and the voltage setting later. This sequence is commonly used in modern voltage-scaled processors, including the Intel Core Duo processor architecture [17].

One of the most important observations is that the microprocessor is supplied by an unnecessarily high-voltage during the voltage transition period. We refer to this situation as the microprocessor-underclocking phenomenon. The microprocessor consumes unnecessarily large amounts of dynamic and static power due to underclocking.

PLL lock time takes typically tens of microseconds for a modern digital PLL [18]. Modern processors such as the Intel’s Nehalem architecture typically have PLLs with several microseconds of lock time [19], [20]. A StrongARM 1100 processor measurement result shows that the PLL lock time is insensitive to the difference between the present and target frequencies [21].

III. PREVIOUS DVFS TRANSITION OVERHEAD MODELS

A. Constant Transition Overhead Models

Constant transition overhead models typically do not distinguish between the voltage and frequency transition times and ignore the voltage transition energy overhead. The underlying assumption is that the PLL lock time is longer than the voltage transition time. In other words, frequency scaling is the time limiting part of the transition, which can be justified for old-fashioned analog PLL clock generators, and the PLL lock time is constant. Some models assume that the microprocessor halts during the entire transition period [18], [23], [24]. Later work used constant transition energy overhead on top of the constant transition time model [25]. Another type of model assumes a constant voltage and a constant frequency transition time aiming at digital PLL. The transition energy overhead is ignored insisting that the microprocessor halts during the transition period [26].

B. Variable Transition Overhead Models

One of the most frequently cited DVFS transition overhead models from [6] assumes a continuous mode DC–DC converter and a VCO. Unfortunately, most published works that
refer to this model do not specify whether a VCO or a PLL is used for the clock generator, and use an overhead value defined by the voltage transition. The notation for previous DVFS transition models in this section is given in Table I. This overhead model consists of time for transition $T_X$ and the energy overhead during the transition time $E_X$ that are given as

$$T_X = \frac{2C_D}{\max(I_D)} |V_i - V_f| \quad (3)$$

$$E_X = (1 - \eta)C_D |V_i^2 - V_f^2| \quad (4)$$

where factor of 2 is applied because the current is pulsed in a triangular waveform, and the efficiency of the DC–DC converter $\eta$ is assumed constant. One shortcoming of this model is overestimation of $\max(I_D)$. While [6] assumes $\max(I_D)$ is an order of magnitude bigger than the microprocessor current demand, in reality, designers do not overdesign the DC–DC converter in this way due to cost and volume consideration. Typical overdesign factor is within a factor of 3 from the average microprocessor current demand. Actually, the target Intel mainboard for E6850 uses a 130-A regulator while E6850 draws 44 A. So, the microprocessor current should be considered to determine $T_X$, that is

$$T_X = \frac{2C_D}{\max(I_D)} |I_f - I_0| \quad (5)$$

Because the microprocessor continues to operate even during the voltage transition, $I_f$ has a significant impact on $T_X$. Note that the transition time $T_X$ is not the actual overhead because the microprocessor may be operating during $T_X$. Only when the microprocessor is halted during the voltage transition period does $T_X$ become the delay overhead for the DVFS transition.

The energy overhead $E_X$ in symmetrical for voltage upscaling and downscaling, which is justified for continuous-mode DC–DC converters only. Unfortunately, $E_X$ equation in [6] gives the same expression for the energy dissipation for both upscaling and downscaling. The expression is twice what the correct value is per up or down transition. In particular, $E_X$ for a downscaling control command dumps the charge that is already stored in the bulk capacitor to the GND, and thus there is no additional current flow (and thus energy extraction) from the power source. In addition, the DC–DC converter efficiency should be considered as $1/\eta$ instead of $(1 - \eta)$. Once again, the bulk capacitor is charged adiabatically, and therefore, the correct $E_X$ for a continuous-mode DC–DC conversion with a VCO DVFS setup is as follows:

$$E_{X^{\text{up}}} = \begin{cases} \frac{1}{2\eta} C_D (V_i^2 - V_f^2) : \text{upscaling} \\ 0 : \text{downscaling.} \end{cases} \quad (6)$$

If voltage upscaling and downscaling occur evenly, the transition overhead may be distributed as follows. (This is similar to the calculation of CMOS logic gate dynamic energy.)

$$E_{X^{\text{up}}} = \frac{1}{4\eta} C_D |V_i^2 - V_f^2| \quad (7)$$

Another frequently cited DVFS transition overhead model is [7], which is basically the same as that of reference [6], but has additional consideration of the body bias.

### IV. FORMULATION OF THE DVFS TRANSITION OVERHEAD

This section presents a new and correct formulation of the DVFS transition overhead with modern DC–DC converters, which correctly accounts for both continuous- and discontinuous-modes of operation and a PLL clock generator. All the distinct sources of the overhead are taken into account including losses from the microprocessor and DC–DC converters as opposed to the use of constant efficiency model in the previous section. Although there are more advanced research-stage alternatives for DC–DC converters [27], [28], we leave modeling for such converters as a future work to provide more reliable results for the target setup. We restrict the scope of the modeling to PWM or PFM controlled buck-type DC–DC converters only, which is the most common setup for now.

The time and energy overheads $E_O$ and $T_O$ are denoted as follows:

$$T_O = T_{\text{trans}} - T_{\text{trans/id}} \quad (8a)$$

An ideal DVFS transition incurs no time and energy overhead as shown in Fig. 8.

$$E_O = E_{\text{trans}} - E_{\text{trans/id}} \quad (8c)$$

### A. DELAY OVERHEAD OF A DVFS TRANSITION

We divide the delay overhead of a DVFS transition into two parts: PLL-induced and underclocking-related delay overhead as shown in the following equation:

$$T_O = T_{\text{PLL}} + T_{\text{trans}} \quad (9)$$

1) **Underclocking-Related Delay Overhead:** We obtain $T_{\text{PLL}}$ by comparing the elapsed time between the real DVFS transition and the ideal DVFS transition. The microprocessor executes a certain amount of task during $T_X$. The time required for the microprocessor to execute the same number of cycles during an ideal transition is $T_{\text{ideal}} = \frac{1}{f} T_X$. Thus,
the underclocking-related delay overhead for an upscaling transition is

\[ T_{\text{up}} = T_s - \frac{f_s}{f_i} T_s + \frac{f_s}{f_i} T_s. \]  

(10)

For downscaling, the underclocking-related overhead is 0 since the processor operates at \( f_i \) immediately after a DVFS transition is initiated for both the ideal and real cases

\[ T_{\text{down}} = 0. \]  

(11)

2) PLL-Induced Delay Overhead: It is the delay overhead due to PLL lock time. Since the processor halts during the PLL lock time, \( T_{\text{PLL}} \) becomes the pure delay overhead of a DVFS transition. We denote \( T_{\text{PLL}} \) lock time as a constant without losing generality. Some recent PLL architectures have constant PLL lock time independent of \( f_s \) and \( f_i \), as described in [21]. Others have varying lock time, but we assume that the maximum value is given by the PLL vendors that can be used in our setup. The total delay overhead becomes

\[ T_d = \begin{cases} 
T_{\text{PLL}} \frac{f_s}{f_i} T_s : & \text{upsampling} \\
T_{\text{PLL}} : & \text{downscaling}.
\end{cases} \]  

(12)

B. Energy Overhead of a DVFS Transition

We divide the DVFS energy overhead into two parts: converter induced and microprocessor induced as shown in the following equation:

\[ E_d = E_{\text{core}} + E_{up}. \]  

(13)

1) Converter-Induced Energy Overhead: It is the energy overhead induced by the DC-DC converter. Switching loss and controller loss does not affect the DVFS transition overhead since the value does not differ much between a real DVFS transition and an ideal DVFS transition, and thus the effect of those components cancels out. However, the conduction loss differs quite a lot between those two cases. A large surge current flows into and out of the bulk capacitor via the inductor and MOSFETs as shown in Figs. 4 and 5 during a voltage transition. This causes additional IR losses in the inductor and MOSFETs.

Upscaling transfers additional charge to the bulk capacitor and increases the capacitor terminal voltage from \( V_l \) to \( V_U \). The amount of stored energy is \( E_{\text{cap}} = \frac{1}{2} C_d (V_U^2 - V_l^2) \). This energy is deposit, but not yet waste as discussed in Section I. The total amount of energy dissipation in the inductor due to this surge current during DVFS transition is shown in (14a).

Meanwhile, the amount of energy dissipation in the inductor in the case of the ideal DVFS transition during \( T_{\text{trans},id} \) is (14b), where \( I_{\text{DL}} \) is the current draw of the processor with \( V_i \) and \( f_s \):

\[ E_{\text{conv,up}} = \begin{cases} 
\tau_s \int_0^{T_s} R_L I_L(t)^2 dt : & \text{upsampling} \\
\tau_s \int_0^{T_{\text{trans},up}} R_L I_{\text{DL}}^2 dt : & \text{downscaling}.
\end{cases} \]  

(14a)

\[ E_{\text{conv,ideal}} = \int_0^{T_{\text{trans},id}} R_L I_{\text{DL}}^2 dt. \]  

(14b)

As \( T_{\text{trans},id} = \frac{f_i}{f_s} T_s \) during upscaling, the energy overhead due to additional inductor IR loss during upscaling is calculated as

\[ E_{\text{conv}} = E_{\text{conv,ideal}} - E_{\text{conv,up}} = \int_0^{T_s} R_L I_L(t)^2 dt - \int_0^{T_{\text{trans},up}} R_L I_{\text{DL}}^2 dt. \]  

(15)

The amount of charge drained to the ground from the bulk capacitor causes energy overhead during downscaling, \( E_{\text{conv,down}} \), is described as (16a). This becomes zero for discontinuous-mode upscaling:

\[ E_{\text{conv,down}} = \int_0^{T_{\text{trans},down}} R_L I_{\text{DL}}^2 dt. \]  

(16a)

The operation mode of the DC-DC converter, continuous- or discontinuous-modes, does not make difference to (15) and (17). It is implied in the term \( I_{\text{DL}}(t) \).

The total converter-induced energy overhead of a DVFS transition is given by

\[ E_{\text{con}} = \begin{cases} 
E_{\text{conv}} : & \text{upsampling} \\
E_{\text{conv,down}} : & \text{downscaling}.
\end{cases} \]  

(18)

2) Microprocessor-Induced Energy Overhead: As we have stated in the beginning of this section, the microprocessor-induced energy overhead, \( E_{\text{cpu}} \), consists of two factors, which are underclocking-related loss, \( E_{\text{cpu}} \), and PLL lock time loss, \( E_{\text{PLL}} \). We use a widely known processor power model as follows:

\[ P_{\text{cpu}} = P_{\text{cpu,s}} + P_{\text{cpu,u}} = \left( C_r V_{\text{CPU}}^2 f_{\text{CPU}} \right) + \left( \alpha_1 V_{\text{CPU}}^2 + \alpha_2 \right) \]  

(19)

where \( P_{\text{cpu,s}} \), \( P_{\text{cpu,u}} \), and \( P_{\text{cpu}} \) are the total power consumption, dynamic power consumption, and static power consumption of the target processor, respectively. The term \( C_r \) is the average switching capacitance per cycle, and \( V_{\text{CPU}} \) and \( f_{\text{CPU}} \) are the supply voltage and the clock frequency of the microprocessor. The parameter \( C_r \) is a strong function of circuit activity that differs from application to application. Processor utilization reflects the power consumption variation quite well among the various application-specific parameters once the clock frequency and supply voltage is fixed while the processor is in active mode [29]. We obtain sets of parameter values \( C_r \), \( \alpha_1 \), and \( \alpha_2 \) using offline regression analysis for each value of processor utilization. Static power is rather independent
of application-specific parameters and thus \( a_1 \) and \( a_2 \) remain almost the same. This results in the following relationship:

\[
C_v \propto \frac{\text{Vref} \times (\text{Vref} + \text{Vuser})}{\text{Vref} \times (\text{Vref} + \text{Vuser})}
\]

where \( \text{Vref} \), \( \text{Vuser} \), \( \text{Vref} \) are total utilization, utilization of the kernel process, and utilization of user process, respectively. Processor utilization is monitored at runtime, and the corresponding value of \( C_v \) is selected. For the very short period of time a DVFS transition takes place, we assume that \( C_v \) remains constant.

Underclocking phenomenon described in Section II-D makes the microprocessor consume additional dynamic and static power. We calculate the loss by (21) during \( T_k \)

\[
E_{\text{up}} = E_{\text{dual}} - E_{\text{dual}}
\]

\[
= \left( C_f f_c V_d(t) + a_1 V_c(t) + a_2 \right) dt
\]

\[
= \left( C_f f_c V_d(t) + a_1 V_c(t) + a_2 \right) dt
\]

\[
E_{\text{down}} = E_{\text{dual}} - E_{\text{dual}}
\]

\[
= \left( C_f f_c V_d(t) + a_1 V_c(t) + a_2 \right) dt
\]

\[
= \left( C_f f_c V_d(t) + a_1 V_c(t) + a_2 \right) dt
\]

Power consumption during the PLL lock time is caused by the static power consumption of the microprocessor during PLL lock time

\[
E_{\text{PLL}} = \int_{\text{PLL}} (a_1 V_c(t) + a_2) dt
\]

\[
E_{\text{PLL}} = \int_{\text{PLL}} (a_1 V_c(t) + a_2) dt
\]

PLL lock time is zero for an ideal transition, and thus \( E_{\text{PLL}} \) becomes a pure overhead.

The total microprocessor-induced energy overhead of a DVFS transition is given by

\[
E_\text{over} = E_{\text{up}} + E_{\text{down}} + E_{\text{PLL}}
\]

\[
E_\text{over} = E_{\text{up}} + E_{\text{down}} + E_{\text{PLL}} : \text{upscaling},
\]

\[
E_\text{over} = E_{\text{up}} + E_{\text{down}} + E_{\text{PLL}} : \text{downscaling}.
\]

V. MACROMODEL FOR DVFS TRANSITION OVERHEAD

Although the DVFS transition overhead is precisely formulated in Section IV, it is not easy to obtain the actual values of overhead. Evaluation of (12), (18), and (23) requires actual profiles of \( V_c(t) \) and \( I_{\text{dual}}(t) \) over time and the value of \( T_k \). Obtaining the profiles requires SPICE simulation. We thus provide an approximate, but a much simpler macromodel for DVFS transition overhead calculation, which consists of easy datasets parameters and RLC values of the DC-DC converter circuit only. The proposed macromodel provides convenience of evaluating the DVFS transition overhead at sacrificed accuracy when compared to SPICE simulation. There are variations of DC-DC converters that cannot be modeled by our macromodel as it uses small number of parameters obtained from datasheet. Modeling such variations might increase the complexity of the model too much, which is contradictory to its purpose. The quality of the modeling is only guaranteed for standard synchronous PWM or PFM controlled buck converters. The symbols used in macromodel are defined in Table I.

A. Macromodel for the Delay Overhead

Datasheets of DC-DC converters usually provide the worst-case value of \( T_k \) only. We propose to calculate the \( T_k \) by obtaining the slope of the initial voltage increase during \( T_1 \), as shown in Fig. 9. Generally, the controller in the DC-DC converter tries to drive the output voltage to the target voltage as fast as possible. The maximum output current of the DC-DC converter is determined by the peak current threshold constraint imposed on the DC-DC converter. We denote the peak current threshold as \( \text{max}(I_2) \), which is specified in the converter datasheet. The slope of the voltage increase is dependent on the current flowing into the bulk capacitor via the inductor \( \text{max}(I_2) \) and current drawn out of the bulk capacitor by the load (processor), \( I_0 \). The rate of output voltage change, \( \text{slope}_\text{up} \), during voltage upscaling is calculated as follows:

\[
\text{slope}_\text{up} = \frac{dV_2}{dt} = \frac{1}{C_2} \left( \text{max}(I_2) - I_0 \right).
\]

(24)

The change in \( I_0 \) during \( T_k \) is much smaller than \( I_2(t) \). Therefore, without losing much accuracy, we regard it as a constant value \( (I_{\text{max}} + I_{\text{max}})/2 \). We devise a heuristic to approximate \( T_k \) using \( \text{slope}_\text{up} \). The value of \( T_k \) is larger when the difference in \( V_0 \) and \( V_2 \) is larger. In addition, \( T_k \) shows correlation with the slope of voltage increase, \( \text{slope}_\text{up} \). We have found that linearizing the correlation between \( \text{slope}_\text{up} \) and \( T_k \) provides acceptable accuracy. We thus derive (25), which implies that \( T_k \) becomes nearly proportional to the rate of approaching the target voltage, \( \text{slope}_\text{up} \).

\[
T_k = T_1 + \text{slope}_\text{up} \cdot \beta.
\]

(25)

The value of \( \beta \) is calculated using the worst case settling time \( T_k \), which is again specified in the datasheet. The worst case \( T_k \) occurs when the difference between the initial and final voltages is the largest:

\[
T_k = (V_2 - V_0)/\text{slope}_\text{up}
\]

(26a)

\[
\beta = (T_k \cdot \text{max}(I_2)/\text{slope}_\text{up} \cdot \text{max}(I_2))
\]

(26b)

\[
T_1 \cdot \text{max}(I_2)/\text{slope}_\text{up}
\]

(26c)
We obtain the underlocking-related delay overhead by substituting (25) into (12).

B. Macromodel for the Converter-Induced Energy Overhead

The major hurdle is how to obtain the trace of $I_d(t)$ over time. We use a simple assumption for upscaling such that $I_d(t) = \max(I_d(t))$ during $T_1$. The value of $I_d(t)$ beyond $T_1$ becomes approximately the same as $I_d(t) = C_1V_d(I+\alpha_1+\alpha_2)/V_c$, derived from (19). The integral term including $I_d(t)$ in (15) and (17) then becomes

$$\int_0^{T_1} R_d I_d(t)^2 dt = R_d \max(I_d) \int_T + R_d \int_{T_1}^T (T_1 - T_1).$$

(27)

Substituting (27) into (15) and (17) gives the additional inductor IR loss for upscaling.

Continuous-mode downscaling makes the duty ratio of the lower MOSFET equal to 1 for fast transition. We derive the voltage curve during $T_1$ by solving the RLC circuit with a constant current source as shown in Fig. 10(a). The traces of $I_d(t)$ and $V_d(t)$ are determined by the passive components in the DC–DC converter, which are the MOSFET on-resistance, inductor, and bulk capacitor. The value of the current source is assumed to be $(I_{d,0} + I_{d,0})/2$ because its change is small during $T_1$. We denote the summation the MOSFET on-resistance and inductor resistance between the supply and the ground as $R$. We obtain the exact trace of node voltages and inductor current from the following system of nonhomogeneous differential equations:

$$\begin{pmatrix}
\frac{dV}{dt} \\
\frac{dI}{dt}
\end{pmatrix} =
\begin{pmatrix}
0 & \frac{1}{R} & \frac{1}{L} & \frac{1}{C} \\
\frac{1}{R} & \frac{1}{L} & \frac{1}{C} & \frac{1}{C}
\end{pmatrix}
\begin{pmatrix}
V \\\nI
\end{pmatrix}
+ \begin{pmatrix}
0 \\
\frac{1}{L}
\end{pmatrix},$$

(28)

Discontinuous-mode downscaling is as simple as in Fig. 10(b) such that

$$V_d(t) = V_c - \frac{I_{d,0}}{C} t.$$  

(29)

Commercial-grade discontinuous-mode DC–DC converters occasionally drain the bulk capacitor when the voltage difference between the output voltage and target voltage is too large. This continues until the error becomes smaller than a certain value, e.g., 0.1-V in case of the LTC3733 converter. We solve both (28) and (29) for such a case and set the appropriate boundary conditions. The obtained trace of $I_d(t)$ is substituted into (17) to calculate the additional inductor IR loss.

1) Macromodel for the Microprocessor-Induced Energy Overhead: We need the trace of $V_d(t)$ over time to calculate the underlocking-related energy overhead (21). For upscaling, we approximate the integral terms of \( \int V_d(t) dt \) and \( \int V_d(t)^2 dt \) in (21) by calculating the area of the two shaded triangles shown in Fig. 9. We assume that the integral values beyond $T_1$, \( \int_{T_1}^{T_1} (V_d(t) - V_c) dt \) and \( \int_{T_1}^{T_1} (V_d(t)^2 - V_c^2) dt \), add up to zero, and thus these terms are ignored

$$\int_0^{T_1} V_d(t) dt \approx T_1 V_c - \frac{1}{2} T_1 (V_c - V_0)$$

(30a)

$$\int_0^{T_1} V_d(t)^2 dt \approx T_1 V_c^2 - \frac{1}{2} T_1 (V_c^2 - V_0^2)$$

(30b)

We calculate $T_1$ from (28c). We calculate $V_0$ and $T_1$ like $T_1$ in (25). We linearize the variations $V_c$, and $T_2 - T_1$ according to the rate of approaching the target voltage slope. Thus, the following equations hold:

$$V_c = y \cdot \text{slope}_{\text{up}}.$$  

(31a)

$$T_2 = T_1 + \delta \cdot \text{slope}_{\text{up}}.$$  

(31b)

The values $y$ and $\delta$ determine the overshoot and settling time, which are device-dependent parameters. The selection of values does not affect the total DVFS transition overhead significantly since their effect is quite small as shown in Fig. 9. Taking (31) into account generally improves the accuracy of the DVFS transition overhead calculation.

For downscaling, we again use the solution of circuits Fig. 10(a) and (b) obtained from (28) and (29). We substitute the trace of $V_d(t)$ into (23) and obtain the underlocking-related energy loss during $T_1$. We assume that the voltage ripple beyond $T_1$ is small enough to cancel the integral terms in $E_{\text{uc,down}}$ in (21).

VI. EXPERIMENTAL RESULTS

In this section, we provide experimental results for the DVFS transition overhead of microprocessors exhibiting distinctive power consumption values as high as 60 W to as low as 10 mW. We show the overhead values obtained from the SPICE simulation and macromodel for three representative processors, which are Intel Core2 Duo, Samsung Exynos 4210, and TI MSP430.

A. Case 1: Intel Core2 Duo E6850 Processor

We choose a high-end DVFS-enabled microprocessor, i.e., Intel Core2 Duo E6850 processor, along with the LTC3733 three-phase synchronous step-down DC–DC converter that supports discontinuous mode, which is a representative setup of a modern high-performance DVFS-enabled microprocessor.
TABLE II
VOLTAGE \(V_{CPU}(V)\) AND CLOCK FREQUENCY \(f_{CPU}(GHz)\) LEVELS FOR THE INTEL CORE2 DUO E6850 PROCESSOR

<table>
<thead>
<tr>
<th>Level</th>
<th>(V_{CPU}) (V)</th>
<th>(f_{CPU}) (GHz)</th>
<th>(V_{CPU}) (V)</th>
<th>(f_{CPU}) (GHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.58</td>
<td>2.704</td>
<td>1.41</td>
<td>2.291</td>
</tr>
<tr>
<td>2</td>
<td>1.25</td>
<td>2.452</td>
<td>1.16</td>
<td>1.932</td>
</tr>
<tr>
<td>3</td>
<td>1.20</td>
<td>2.586</td>
<td>1.15</td>
<td>1.946</td>
</tr>
</tbody>
</table>

TABLE III
MEASURED AND ANALYTICAL MODELS OF INTEL CORE2 DUO E6850 POWER CONSUMPTION

<table>
<thead>
<tr>
<th>(V_{CPU}(V))</th>
<th>(f_{CPU}(GHz))</th>
<th>Measurement (W)</th>
<th>Analytical model (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.005</td>
<td>1.073</td>
<td>21.250</td>
<td>17.356</td>
</tr>
<tr>
<td>1.000</td>
<td>1.088</td>
<td>24.000</td>
<td>25.956</td>
</tr>
<tr>
<td>1.094</td>
<td>2.004</td>
<td>26.830</td>
<td>28.836</td>
</tr>
<tr>
<td>1.100</td>
<td>2.312</td>
<td>33.760</td>
<td>34.393</td>
</tr>
<tr>
<td>1.228</td>
<td>4.062</td>
<td>43.200</td>
<td>44.409</td>
</tr>
<tr>
<td>1.250</td>
<td>0.006</td>
<td>57.640</td>
<td>54.238</td>
</tr>
</tbody>
</table>

TABLE IV
DC–DC CONVERTER PARAMETERS OF LTC3733 THREE-PHASE CONVERTER FOR INTEL CORE2 DUO E6850

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(V_{in})</td>
<td>12-V</td>
<td>(f_{CPU})</td>
<td>100mHz</td>
</tr>
<tr>
<td>(C)</td>
<td>864nF</td>
<td>(V_{CPU}) in Table II</td>
<td></td>
</tr>
<tr>
<td>(L)</td>
<td>1µH/MHz phase</td>
<td>(I_{CPU})</td>
<td>75 A</td>
</tr>
<tr>
<td>(R_{P})</td>
<td>2.5mΩ</td>
<td>(I_{CPU})</td>
<td></td>
</tr>
<tr>
<td>(R_{IN}/L)</td>
<td>75 Ω</td>
<td>(f_{CPU})</td>
<td></td>
</tr>
</tbody>
</table>

We describe the microprocessor power consumption model in (19). We obtain the parameters \(C_{p}, a_{1}, a_{2}\) from real measurements. We insert a shunt monitor circuit in front of the DC–DC converter of the Intel Core2 Duo E6850 processor and measure the power supply current with an Agilent A34401 digital multimeter. We compensate the DC–DC converter efficiency from the measured current values, and characterize \(I_{CPU}\). We run PrimeZ benchmark and change \(V_{CPU}\) and \(f_{CPU}\) performing direct access to the basic input/output system (BIOS) as described in Table II because the Intel SpeedStep supports only two voltage levels. We finally derive the following power consumption model:

\[
P_{CPU} = 8.4503V_{CPU}V_{curr} – 3.39503 \quad \text{(32)}
\]

where the units of \(P_{CPU}\), \(V_{CPU}\), and \(f_{CPU}\) are W, V, and GHz, respectively. We use LTC3568 converter capable of supplying up to 1.8 A output current, of which the converter parameters are shown in Table IX. The difference between the analytical model and measurement results is less than 4.6% as shown in Table III. The DC–DC converter parameters are given in Table IV. The values are chosen according to guidelines in datasheet and reference designs offered by the vendor.

The delay overhead of DVFS transition is given in Table V and Fig. 11. The value of \(T_{DPU} \leq 5 \mu s\) is specified in the Intel Core2 Duo E6850 datasheet. The actual values are obtained from SPICE simulation results. We obtain \(T_{DPU}\) by observing the settling time of \(V_{CPU}(t)\) from SPICE results and substitute it into equations in Section IV to calculate the delay overhead. The estimated overhead from the proposed macromodel well matches the results obtained from the above SPICE simulation.

![Fig. 11. Delay overhead of a DVFS transition for Intel Core2 Duo E6850. (a) Upscaling (actual). (b) Upscaling (model).](image-url)

follows the trend of actual values. Unlike assumption of previous works, the underclocking-related overhead is the dominant factor for most cases as we have discussed in Section III. Fig. 11 graphically shows the trend of delay overhead according to different start and end voltage levels. The delay overhead value tends to be larger when the voltage difference is larger between the start and end voltage. The energy overhead values of a DVFS transition for continuous- and discontinuous-mode operations are given in Tables VI and VII, respectively. We obtain \(I_{CPU}(t), I_{CPU}(t),\) and \(V_{CPU}(t)\) from SPICE simulation and substitute them into (15), (17), and (21) to calculate the actual value. There is no \(E_{off}\) in the tables as it is implied in \(E_{on}\). The value of \(E_{on}\) for the case 1 \(-\) 6 is large because it drains significant amount of charge from the bulk capacitor to the ground. On the other hand, \(E_{on}\) for the same case in Table VII is much smaller because it uses most of the stored charge to supply the load. This result is very different from previous models such as [6] as they simply calculate the overhead based on the charge transfer to and from the bulk capacitor. The error ratio is quite large for some cases such as 3 \(-\) 4. However, we emphasize that the absolute value of error is within an acceptable range. Fig. 12 graphically shows the trend of energy overhead according to different start and end voltage values.
TABLE VI
DVFS Transition Energy Overhead of LTC3733 Operating in
Continuous Mode for Intel Core2 Duo E6850 Processor

<table>
<thead>
<tr>
<th>Level</th>
<th>Actual value (µJ)</th>
<th>Proposed model (µJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( E_0 )</td>
<td>( E_p )</td>
</tr>
<tr>
<td>1→2</td>
<td>2.5</td>
<td>62.8</td>
</tr>
<tr>
<td>1→3</td>
<td>39.7</td>
<td>98.9</td>
</tr>
<tr>
<td>1→4</td>
<td>72.5</td>
<td>177.5</td>
</tr>
<tr>
<td>1→5</td>
<td>240.0</td>
<td>398.5</td>
</tr>
<tr>
<td>2→3</td>
<td>115.3</td>
<td>31.3</td>
</tr>
<tr>
<td>2→4</td>
<td>185.9</td>
<td>62.4</td>
</tr>
<tr>
<td>2→5</td>
<td>167.1</td>
<td>99.3</td>
</tr>
<tr>
<td>3→4</td>
<td>320.2</td>
<td>96.7</td>
</tr>
<tr>
<td>3→5</td>
<td>14.8</td>
<td>64.2</td>
</tr>
<tr>
<td>4→3</td>
<td>85.3</td>
<td>110.1</td>
</tr>
<tr>
<td>4→5</td>
<td>141.1</td>
<td>253.4</td>
</tr>
<tr>
<td>5→4</td>
<td>59.8</td>
<td>23.6</td>
</tr>
<tr>
<td>5→6</td>
<td>102.1</td>
<td>136.6</td>
</tr>
<tr>
<td>6→5</td>
<td>12.7</td>
<td>87.9</td>
</tr>
<tr>
<td>6→4</td>
<td>91.0</td>
<td>196.4</td>
</tr>
<tr>
<td>6→3</td>
<td>67.1</td>
<td>73.4</td>
</tr>
<tr>
<td>7→2</td>
<td>43.2</td>
<td>173.5</td>
</tr>
<tr>
<td>7→3</td>
<td>53.3</td>
<td>104.5</td>
</tr>
<tr>
<td>7→4</td>
<td>42.1</td>
<td>707.8</td>
</tr>
<tr>
<td>7→5</td>
<td>49.3</td>
<td>340.5</td>
</tr>
<tr>
<td>8→5</td>
<td>55.6</td>
<td>132.1</td>
</tr>
<tr>
<td>9→5</td>
<td>140.6</td>
<td>1014.1</td>
</tr>
<tr>
<td>9→6</td>
<td>92.9</td>
<td>680.5</td>
</tr>
<tr>
<td>9→7</td>
<td>66.8</td>
<td>135.0</td>
</tr>
<tr>
<td>10→5</td>
<td>588.6</td>
<td>1635.9</td>
</tr>
<tr>
<td>10→6</td>
<td>511.5</td>
<td>1722.4</td>
</tr>
<tr>
<td>10→7</td>
<td>184.9</td>
<td>966.4</td>
</tr>
<tr>
<td>11→4</td>
<td>74.1</td>
<td>672.6</td>
</tr>
<tr>
<td>11→5</td>
<td>47.0</td>
<td>256.3</td>
</tr>
</tbody>
</table>

Table VI shows the energy overheads of transitioning from one level to another for the Intel Core2 Duo E6850 processor using the LTC3733. The actual values are compared with the proposed model to estimate the energy consumption accurately.

TABLE VII
DVFS Transition Energy Overhead of LTC3733 Operating in
Discontinuous Mode for the Intel Core2 Duo E6850 Processor

<table>
<thead>
<tr>
<th>Level</th>
<th>Start level (downscale)</th>
<th>Final level (upscale)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

TABLE VII continues the analysis of energy overhead for transitions in discontinuous mode with the Intel Core2 Duo E6850 processor.

TABLE VIII
Voltage \( V_{VFS} \) and Clock Frequency \( f_{CPU} \) (GHz) Levels for Samsung Exynos 4210

<table>
<thead>
<tr>
<th>Level</th>
<th>( V_{VFS} ) (V)</th>
<th>( f_{CPU} ) (GHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1</td>
<td>1.35</td>
<td>1.4</td>
</tr>
<tr>
<td>Level 2</td>
<td>1.15</td>
<td>1.324</td>
</tr>
<tr>
<td>Level 3</td>
<td>1.10</td>
<td>1.2218</td>
</tr>
</tbody>
</table>

The table shows voltage and clock frequency levels for different levels in the Samsung Exynos 4210 processor.

TABLE IX
DC – DC CONVERTER PARAMETERS OF THE LTC3566 CONVERTER FOR SAMSUNG EXynos 4210

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_{DC} )</td>
<td>5.5 V</td>
<td>( I_{E} )</td>
<td>1 A</td>
</tr>
<tr>
<td>( V_{OUT} )</td>
<td>4 V</td>
<td>( f_{CPU} )</td>
<td>2.0 GHz</td>
</tr>
</tbody>
</table>

TABLE IX provides the DC-DC converter parameters for the LTC3566 used in the Samsung Exynos 4210 processor.

The energy overhead tends to be larger when the voltage difference is larger. For downsampling in discontinuous mode, the overhead is smaller than the value in continuous mode as the charge in the bulk capacitor is supplied to the processor rather than being discharged to the ground.

B. Samsung Exynos 4210 Processor

The second target DVFS system is the Samsung Exynos 4210 processor based on ARM Cortex-A9 core (Table IX).
Table X

<table>
<thead>
<tr>
<th>Level</th>
<th>Actual</th>
<th>Model (μs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>tc</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>±φc</td>
<td>4.75</td>
</tr>
<tr>
<td></td>
<td>±φd</td>
<td>4.75</td>
</tr>
<tr>
<td></td>
<td>±φs</td>
<td>4.75</td>
</tr>
<tr>
<td></td>
<td>±φd</td>
<td>4.75</td>
</tr>
<tr>
<td></td>
<td>±φs</td>
<td>4.75</td>
</tr>
<tr>
<td></td>
<td>±φd</td>
<td>4.75</td>
</tr>
<tr>
<td></td>
<td>±φs</td>
<td>4.75</td>
</tr>
<tr>
<td></td>
<td>±φd</td>
<td>4.75</td>
</tr>
<tr>
<td></td>
<td>±φs</td>
<td>4.75</td>
</tr>
</tbody>
</table>

Fig. 13. Delay overhead of a DVFS transition for Samsung Exynos 4210 (a) Upscaling (actual). (b) Upscaling (model).

The third target system is the TI MSP430 microcontroller. TI MSP430 is a microcontroller used for ultralow-power embedded systems such as wireless sensor nodes. The power consumption of the TI MSP430 microcontroller is at most 10.1 mW. A procedure similar to that for Samsung Exynos 4210 is performed to obtain the following power model:

\[ P_{\text{proc}} = 0.1128 V_{\text{ref}} V_{\text{CPU}}^2 f_{\text{CPU}} + (0.1738 V_{\text{ref}} - 0.2832) \]

where the units of \( P_{\text{proc}}, V_{\text{CPU}}, \) and \( f_{\text{CPU}} \) are mW, V, and MHz, respectively. We use the LTC3632 converter to power the target processor. LTC3632 is designed for low-power applications. It is PWM controlled and operates in discontinuous mode only. The parameters for the DC-DC converter are reported in Table XIII.

Table XIV and Fig. 15 show the DVFS transition delay overhead for the target system. There is no overhead due to PLL lock time \( T_{\text{PLL}} \) because TI MSP430 uses digitally controlled oscillator (DCO) instead of PLL, which is an improved variation of VCO. The underclocking-related overhead \( T_{\text{clk}} \) is the only delay overhead for the TI MSP430 microcontroller.

Table XV and Fig. 16 show the DVFS transition energy overhead for the target system. Upscaling transitions have large \( E_{\text{u}} \) while downsampling transitions have large \( E_{\text{d}} \). The bulk capacitor is discharged slowly with light load current during a downsampling transition.

VII. IMPACT OF DVFS TRANSITION OVERHEAD: DYNAMIC THERMAL MANAGEMENT EXAMPLE

In this section, we show how much DVFS transition overhead impacts on overall system performance and energy consumption when we perform DTM. DVFS is a very useful
TABLE XI
DVFS Transition Energy Overhead for Samsung Exynos 4210
With the LTC3632 Converter

<table>
<thead>
<tr>
<th>Level</th>
<th>Active Value</th>
<th>Proposed model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$E_a$ (µJ)</td>
<td>$E_p$ (µJ)</td>
</tr>
<tr>
<td>1–5</td>
<td>1.95</td>
<td>0.75</td>
</tr>
</tbody>
</table>

TABLE XII
Voltage ($V_{DC}$) and Clock Frequency ($f_{cpu}$ (MHz)) Levels for the TI MSP430 Microcontroller

<table>
<thead>
<tr>
<th>DVFS level</th>
<th>$V_{DC}$</th>
<th>$f_{cpu}$</th>
<th>$E_{total}$ (µJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1</td>
<td>3.3</td>
<td>8</td>
<td>2.175</td>
</tr>
<tr>
<td>Level 2</td>
<td>2.55</td>
<td>6</td>
<td>1.18</td>
</tr>
</tbody>
</table>

TABLE XIII
DC–DC Converter Parameters of the LTC3632 Converter for the TI MSP430 Microcontroller

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{DC}$</td>
<td>3.3V</td>
<td>$L$</td>
<td>5 µH</td>
</tr>
<tr>
<td>$I_{L}$</td>
<td>1:6:16</td>
<td>$I_{max}$</td>
<td>Variable</td>
</tr>
<tr>
<td>$P_{max}$</td>
<td>100 mW</td>
<td>$T_a$</td>
<td>Delay (µs)</td>
</tr>
</tbody>
</table>

Fig. 15. Delay overhead of a DVFS transition for TI MSP430. (a) Upscaling (actual). (b) Upscaling (model).

control knob for DTMs [30], [31]. DTM techniques based on PID control method usually use the time quantum of the operating system as the minimum time granularity. The time quantum of operating system is in the range of a few milliseconds. On the contrary, the thermal RC time constant of a processor is much larger than the time quantum of operating systems. Although the two time constants differ in magnitude, the DVFS transition occurs much more frequently than the thermal RC time constant when the chip temperature is near the target temperature.

We implement a PID control-based DTM scheme in MATLAB/Simulink environment. Parameters of the PID controller are determined by a tuner embedded in MATLAB/Simulink. The thermal resistance from the chip to the ambient is $R = 0.7$ kW and thermal capacitance of the chip is $C = 140.3$ J/K, which is the same as [30]. The thermal RC constant is 98.21 s. Fig. 17 shows the delay and energy overhead of DVFS accord-
Dynamic voltage and frequency scaling (DVFS) is widely used for energy saving and thermal management nowadays. Understanding correct DVFS transition overhead is crucial in achieving the maximum power gain and thermal stability. In fact, DVFS transition overhead is comparable to context switching overhead in modern microprocessors. However, DVFS transition overhead has not been properly dealt with so far due to an absence of accurate models.

This is the first paper that introduced accurate DVFS transition overhead models. We showed that energy to charge and discharge the bulk capacitor in the DC-DC converter, which was regarded as the major source of overhead, is not true overhead. Instead, we introduced energy and delay overhead caused by microprocessor underclocking and additional current through the inductor. This paper provided comprehensive solutions for the models, but the derived model is somewhat complicated for system engineers. We also provided succinct macromodels while maintaining reasonable accuracy. Finally, we summarize DVFS transition overhead values of three representative microprocessors for high-end, embedded, and ultra-low-power applications so that some software programmers may simply use the numbers.

REFERENCES

Qing Xie (S’12) received the B.S. degree in physics from Fudan University, Shanghai, China, in 2007, and the M.S. degree in physics from Northeastern University, Boston, MA, in 2009, respectively. He is currently working toward the Ph.D. degree in System Power Optimization and Regulation Technologies (SPORT) Laboratory, University of Southern California, Los Angeles, under the supervision of Prof. M. Pedram.

His research interests include the area of system-level power management, thermal management, and low-power systems design.

Massoud Pedram (F’01) received the B.S. degree in electrical engineering from the California Institute of Technology, Pasadena, in 1986, and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1989 and 1991, respectively.

He then joined the Department of Electrical Engineering—Systems, the University of Southern California, Los Angeles, where he is currently a Professor and a Chair of the computer engineering. He has published four books and more than 400 journal and conference papers. His current research includes energy-efficient computing, energy storage, low-power electronics and design, computer-aided design of VLSI circuits, and quantum computing.

Dr. Pedram has served on the technical program committee of a number of conferences, including the Design automation Conference; Design and Test in Europe Conference, and International Conference on Computer-Aided Design. He co-founded and served as the Technical Co-Chair and General Co-Chair of the International Symposium on Low Power Electronics and Design in 1996 and 1997, respectively. He was the Technical Program Chair and the General Chair of the 2002 and 2003 International Symposium on Physical Design. His research has received a number of Best Paper awards including two from IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN and one on VLSI SYSTEMS.

He is a recipient of the NSF’s Young Investigators Award in 1994 and the Presidential Faculty Fellows Award (a.k.a. PECASE Award) in 1996. He is an ACM Distinguished Scientist. He currently serves as the Editor-in-Chief of the ACM Transactions on Design Automation of Electronic Systems and the IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS.

Nasbeysh Chang (F’12) 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 the Department of Computer Engineering, Seoul National University, in 1997, where he is currently a Professor with the Department of Electrical Engineering and Computer Science and the Vice Dean of College of Engineering. His current research interests include low-power embedded systems, hybrid electrical energy storage systems, next-generation energy sources. He has served on the technical program committees in many EDA conferences, including DAC, International Conference on Computer-Aided Design (ICCAD), International Symposium on Low Power Electronics and Design (ISLPED), DATE, CODES+ISSS, and ASP-DAC. He was a TPC Co-Chair of International Conference on Embodied and Real-Time Computing Systems and Applications 2007, ISLPED 2009, ESTIMedia 2009 and 2010, and CODES+ISSS 2012, and will serve as the TPC Chair of ICCD 2014, and the ASP-DAC 2015. He was the General Vice-Chair of ISLPED 2010, General Chair of ISLPED 2011, and ESTIMedia 2011. He was an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN, ACM Transactions on Design Automation of Electronic Systems, Springer DAES, and was a Guest Editor of ACM Transactions on Design Automation of Electronic Systems in 2010, and ACM Transactions on Embedded Computing Systems in 2010 and 2011. He is the ACM SIGDA Chair and an ACM Distinguished Scientist.