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*, Naehyuck Chang *Fellow, IEEE*, and Massoud Pedram *Fellow, IEEE*

***Abstract*—Dynamic voltage and frequency scaling (DVFS) has been studied for well over a decade. The state-of-the-art DVFS technologies and architectures are advanced enough such that they are employed in most commercial systems today. Nevertheless, existing DVFS transition overhead models suffer from significant inaccuracies, for example, by correctly accounting for the effect of DC-DC converters, frequency synthesizers, and 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. Through detailed analysis of modern DVFS setups and voltage and frequency change policies provided by commercial vendors, this paper introduces accurate DVFS transition overhead models for both energy consumption and delay. In particular, we identify new contributors to 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 three representative processors: Intel Core2Duo E6850, ARM Cortex-A8, and TI MSP430. Finally, we present a compact, yet accurate, DVFS transition overhead macro model for use by high-level DVFS schedulers.**

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 from ultra low-power microprocessors for embedded applications such as the TI MSP430 to high-performance microprocessors for desktops and servers such as the Intel's SpeedStep Technology [1] and the AMD equivalent PowerNow!. DVFS is enabled by 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.

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. For such policies, the transition overhead is a major deterrent to wider and more effective adoption of the DVFS. 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. Perhaps due to this reason, among the 120 DVFS-related papers published in 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], [4], [5]. Among the 17% of DVFS papers, 75% of papers are based on the analytical transition overhead models introduced in [6], [7]. Some of the previous work (e.g. [6], [8], [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, majority of the prior art papers consider a DVFS transition overhead model based on incorrect assumptions. A recent work has raised this problem and suggested the correct definition of DVFS transitions [11]. 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 will be 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 macro model for DVFS transition overhead. This paper takes into account all the major power and performance loss components in the modern DVFS setups as follows:

- Conventional DVFS transition models consider the PLL lock time as the major delay (latency) overhead, and the energy required to charge and discharge the bulk

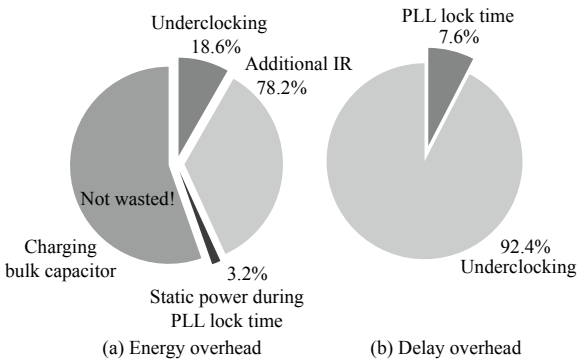

Fig. 1. Breakdown of DVFS transition overhead in energy and time (upscaling).

capacitor as the energy overhead. Both assumptions are incorrect (delay overhead due to PLL lock time accounts for only 7.6% of the total delay overhead as shown in Fig. 1). Energy consumed for charging and discharging the bulk capacitor does not fully account for the energy overhead, especially for the discontinuous mode DC–DC converters since they discharge the bulk capacitor by the load current. Fig. 1(a) shows that more than half of the energy is used to charge the bulk capacitor. However, a significant portion of this energy will be used by the load device again during voltage downscaling.

- During the DVFS transition, the microprocessor operates at a higher supply voltage level than what is strictly necessary. This results in energy waste. We call this phenomenon the *underclocking-related loss*, which is a significant source of energy overhead during the mode change (energy overhead due to under clocking accounts for 18.6% of total energy overhead as shown in Fig. 1(a)). In addition, the underclocking causes the microprocessor to operate at a lower clock frequency than what is allowed during the voltage-frequency upscaling, which is a major source of the delay overhead (92.4% of total delay overhead as shown in Fig. 1(b)).
- Voltage upscaling in a conventional DC–DC converter requires more current to be fed through the inductor to increase the bulk capacitor voltage. This in turn results in additional IR loss from the inductor (78.2% of total energy overhead as shown in Fig. 1(a)).
- During the PLL lock time, although the microprocessor halts, it continues to consume static power. 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 E680, ARM Cortex A-8, and TI MSP430

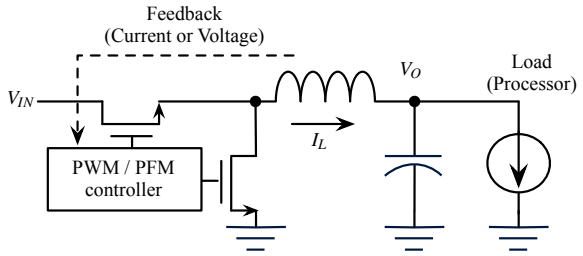

Fig. 2. A DVFS enabled microprocessor model with a buck type DC–DC converter.

Microcontroller. Some programmers who have no hardware knowledge may use the numbers. We finally emphasize the importance of considering the DVFS transition overhead for a dynamic thermal management (DTM) example.

II. Background

A. DC–DC Converters for DVFS

In addition to a DVFS-enabled microprocessor, DVFS setups require a voltage regulator and a clock generator with programmable output voltage and frequency, respectively.

A switching-mode DC–DC converter typically exhibits much higher conversion efficiency compared to a linear regulator (that has a dropout voltage caused by the input and output voltage difference). A microprocessor is generally powered by a buck type switching-mode DC–DC converter as shown in Fig. 2. In this type of design, the inductor current increases when the upper MOSFET is turned on. This current in turn charges the bulk capacitor. The inductor current continuously decreases when the lower MOSFET is turned on, but the inductor still keeps supplying current to the bulk capacitor, dissipating the stored electromagnetic energy. More importantly, 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. The primary sources of losses for the bulk capacitor charge and discharge are the conduction loss of the MOSFET, the IR loss of the inductor, and the MOSFET gate drive loss.

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 are usually used in modern switching regulator designs to overcome the disadvantages of voltage-mode control [12]. The key difference between the current- and voltage-mode control is the current feedback loop. A fixed frequency clock periodically turns on the upper-side MOSFET. The output error and the signal derived from the inductor current determine when to turn off the MOSFET. In other words, the error voltage directly controls the peak switching current. Fast response time is achieved by direct inductor

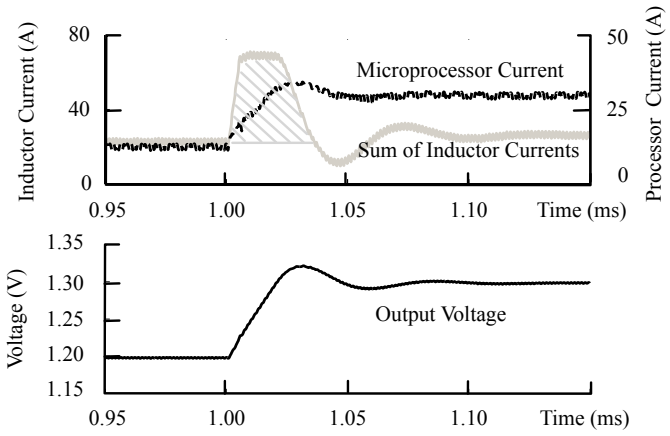

Fig. 3. SPICE simulation result of an upscaling transition (Level 3 → Level 1).

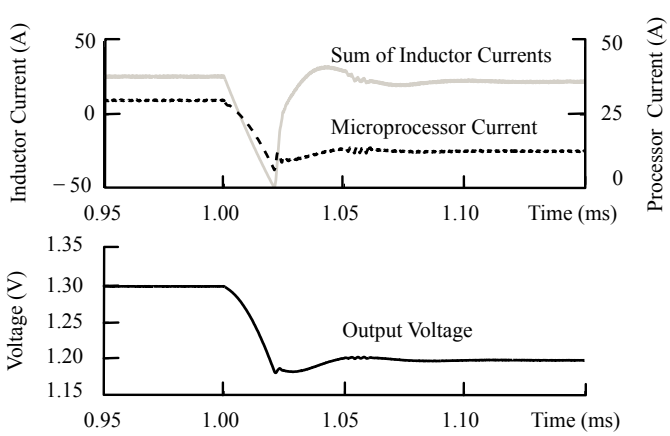

Fig. 4. SPICE simulation result of an continuous-mode downscaling transition (Level 1 → Level 3).

current sensing. However, modeling the behavior of current-mode controlled DC–DC converters is not a trivial task since this type of converter exhibits highly non-linear characteristics. Modeling the behavior itself is a demanding task, and many papers only consider small-signal modeling [13].

DC–DC converters for low-power applications usually adopt pulse-frequency modulation (PFM) since PFM method exhibit higher efficiency with light load. We assume a general setup for microprocessor systems throughout the paper; a buck type DC–DC converter with peak current-mode PWM control for high-power processors and PFM for low-power processors.

C. Voltage Transition Sequences in Continuous and Discontinuous Modes

In this section, we observe and characterize the DVFS transition sequence for various cases. We characterize the upscaling and downscaling sequences using the continuous- and discontinuous-mode DC–DC converters separately because their behaviors are quite different.

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 VID (voltage identifier) code to make the DC-DC converter generate a higher output voltage. The voltage comparator recognizes that the DC–DC converter output voltage (i.e., the bulk capacitor terminal voltage) is lower than the VID and increases the duty ratio of the upper MOSFET. This increases the inductor current, and the charging current of the bulk capacitor becomes larger than the discharging current (which is the current consumed by the microprocessor). This eventually increases the bulk capacitor voltage. There is no difference in the voltage transition sequence between the continuous- and discontinuous-mode during voltage upscaling.

Voltage upscaling pumps more charge into the bulk capacitor by increasing $I_L(t)$. Fig. 3 illustrates an SPICE simulation of an upscaling transition of Intel Core2 Duo E6850 processor using LTSPICE [14]. The shaded area denotes the amount of additional energy 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) Downscaling Transition Sequence using Continuous Mode DC–DC Converters: Downscaling stands for decreasing the supply voltage and clock frequency. Continuous-mode discharges the bulk capacitor to GND by the microprocessor power supply current together with the inductor current. Such active discharging operation to GND results in significant energy loss. On the plus side, the voltage transition time will be shorter in this case. Modern DVFS setups prefer to use discontinuous-mode for more efficient use of the stored energy in the bulk capacitor.

Fig. 4 shows how continuous-mode DC–DC conversion performs voltage downscaling. The downscaling transition stabilizes in 40 μs, during which the bulk capacitor is actively discharged to GND (by flow of negative inductor current). This helps reduce the DVFS voltage transition time, but unfortunately, it increases the transition energy overhead. Voltage upscaling is generally slower than the voltage downscaling due to the limited capacity of the power source due to its internal resistance, heavily loaded long wire from the positive power supply, delay overhead of the boost-up gate drive for the high-side MOSFET in the DC–DC converter circuit, etc.

3) Downscaling Transition Sequence using Discontinuous Mode DC–DC Converters: Fig. 5 shows how discontinuous-mode works. As soon as the inductor current becomes negative, the bottom transistor is turned off, which prevents the bulk capacitor from discharging further. Instead, $I_O(t)$ discharges the bulk capacitor and makes the DC–DC converter output voltage converge to V_e. Downscaling takes longer to stabilize in the discontinuous mode compared to the continuous mode because only $I_O(t)$ discharges the bulk capacitor. On the positive side, the actual energy overhead of the discontinuous mode is much smaller than that of the continuous mode. To shorten the settling time, some DC–DC converters operating in the discontinuous mode allow the inductor current to become

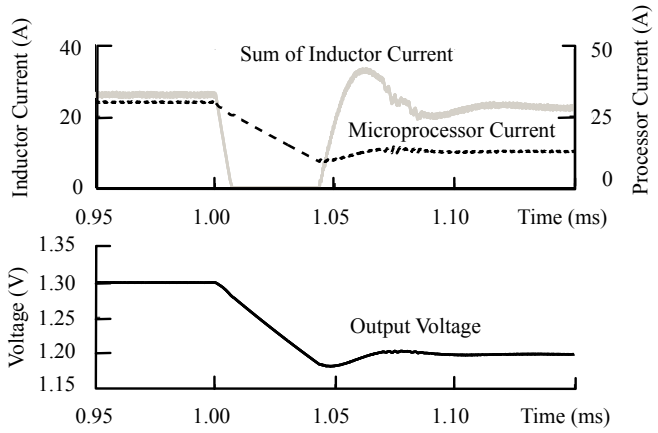

Fig. 5. SPICE simulation result of a discontinuous-mode downscaling transition (Level 1 → Level 3).

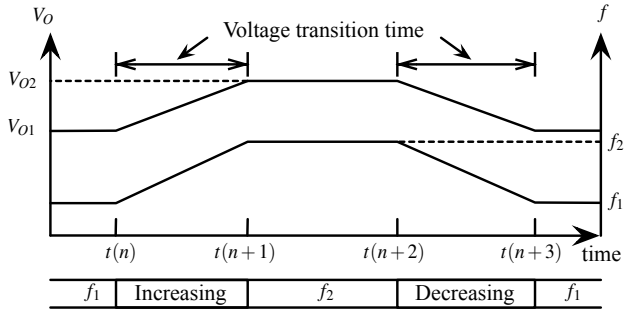

Fig. 6. DVFS upscaling and downscaling with a VCO.

negative until the output voltage is close enough (e.g. 0.1 V) to the target voltage V_e. However, discontinuous mode gives rise to more underclocking of the microprocessor due to longer transition time, which is an additional energy overhead as shown in Fig. 7. We will describe the underclocking energy overhead in Section IV-C in detail. Discontinuous mode plays an important role in modern DC–DC converter design by maintaining high conversion efficiency even when the load current is light.

D. Clock Frequency Transition

The relationship between the supply voltage and clock frequency is approximately explained by the Alpha Power Law [15]. 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. As Fig. 6 illustrates, 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. Some low-performance microcontrollers running at around a few MHz, such as TI MSP430, use a VCO though.

On the other hand, PLLs are widely used for the programmable clock generators thanks to the accuracy of the

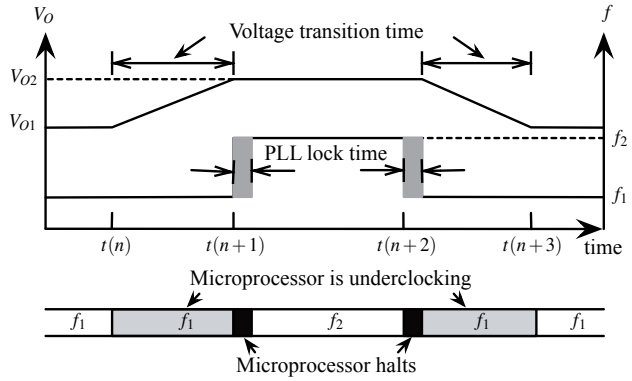

Fig. 7. DVFS up and downscaling for a PLL.

frequency setting.

As illustrated in Fig. 7, upscaling first attempts voltage change and waits until the voltage is stabilized. Once the voltage is stabilized, the microprocessor changes the PLL setting. 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 [16]. 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. We have identified this situation as one of the dominant sources of the voltage transition energy overhead, and quantify it through detailed analysis in Section IV.

PLL lock time takes typically tens of micro-seconds for a modern digital PLL [17]. Modern processors such as the Intel's Nehalem architecture typically have PLLs with several micro-seconds of lock time [1], [18]. A StrongARM 1100 processor measurement result shows that the PLL lock time is insensitive to the difference between the present and target frequencies [19]. As a side note, the IBM's PowerTune technology is able to make a frequency transition in one cycle using multiple pre-generated clocks and selecting one by a multiplexer [20]. Although our proposed model focuses on conventional digital PLLs, it is also applicable to this technology except for the PLL lock time in Section IV-B. It is obvious that the PLL lock time of this technology is one clock cycle.

III. Previous DVFS Transition Overhead Models

This section introduces previous DVFS transition overhead models. Once again, these models cannot be applied to modern DVFS setup as discussed before.

A. Constant Transition Overhead Models

Constant transition overhead models typically do not distinguish between the voltage and frequency transition times and

TABLE I
NOTATION FOR DVFS TRANSITION OVERHEAD DEFINITION AND MODELING.

T_X	Time to complete a voltage transition
T_O	Total delay overhead of a DVFS transition
E_O	Total energy overhead of a DVFS transition
V_s/V_e	Output voltage before/after a DVFS transition
f_s/f_e	Clock frequency before/after a DVFS transition
η	Converter efficiency (constant value) used for previous DVFS transition models
C_b	Output capacitance of a DC–DC converter
$max(I_L)$	Maximum output current of a DC–DC converter specified in the datasheet
T_{uc}	Underclocking-related delay overhead
T_{PLL}	Delay overhead due to PLL lock time
E_{conv}	Converter-induced energy overhead of a DVFS transition
$E_{\mu p}$	Microprocessor-induced energy overhead of a DVFS transition
E_{uc}	Energy overhead due to underclocking
E_{pll}	Energy overhead due to processor energy consumption during the PLL lock time
$V_O(t)$	Transient output voltage of a DC–DC converter
$I_L(t)$	Sum of transient current of inductors
$I_O(t)$	Transient load current, i.e., the microprocessor current
T_{trans}	Total time for a DVFS transition to finish
$T_{trans,id}$	Time to execute equivalent number of instructions when an ideal transition takes place
E_{trans}	Energy consumption of all components during T_{trans}
$E_{trans,id}$	Energy consumption of all components during $T_{trans,id}$ when an ideal transition takes place
T_1	The first crossing between V_O and V_e during upscaling
T_2	The second crossing between V_O and V_e during upscaling
$slope_{up}$	Average slope of increasing V_O during T_1
V_{ov}	Voltage overshoot when upscaling
β	Coefficient for $slope_{up}$ and T_X relationship
γ	Coefficient for $slope_{up}$ and V_{ov} relationship
δ	Coefficient for $slope_{up}$ and T_2 relationship

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. These models assume that the microprocessor halts during the entire transition period [17], [21], [22]. Later work used constant transition energy overhead on top of the constant transition time model [23]. Another type of model considered the voltage transition time and frequency transition time separately, accounting for a digital PLL whose lock time is shorter than the voltage transition time, i.e., voltage transition is the time limiting part of the transition. However, this work assumes a constant voltage transition time. The transition energy overhead is ignored insisting on that the microprocessor halts during the transition period [24].

B. Variable Transition Overhead Models

One of the most frequently-referred 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. This overhead model consists of time for transition, T_X, and the energy overhead during the transition time, E_X. The notation for previous DVFS transition models in this section is given in Table I

$$T_X = \frac{2C_b}{max(I_L)}|V_e - V_s|, \tag{1}$$

$$E_X = (1-\eta)C_b|V_e^2 - V_s^2|, \tag{2}$$

where factor of 2 is applied because the current is pulsed in a triangular waveform, and the efficiency of the DC–DC converter η is assumed constant. One shortcoming of this model is overestimation of $max(I_L)$. While [6] assumes $max(I_L)$ is much 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 an 130A regulator while E6850 draws 44A. So, the microprocessor current should be considered to determine T_X i.e.,

$$T_X = \frac{2C_b}{max(I_L) - I_O}|V_e - V_s|. \tag{3}$$

Because the microprocessor continues to operate even during the voltage transition, I_O has a significant impact on T_X. Notice that the transition time, T_X, is not the actual overhead because the microprocessor may be operating during T_X. Only if the microprocessor is halted during the voltage transition period, T_X becomes the delay overhead for the DVFS transition.

The energy overhead, E_X, is 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 up 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.

$$E_X^* = \begin{cases} \frac{1}{2\eta}C_b(V_e^2 - V_s^2) & : \text{upscaling,} \\ 0 & : \text{downscaling.} \end{cases} \tag{4}$$

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

$$E_X^{**} = \frac{1}{4\eta}C_b|V_e^2 - V_s^2|. \tag{5}$$

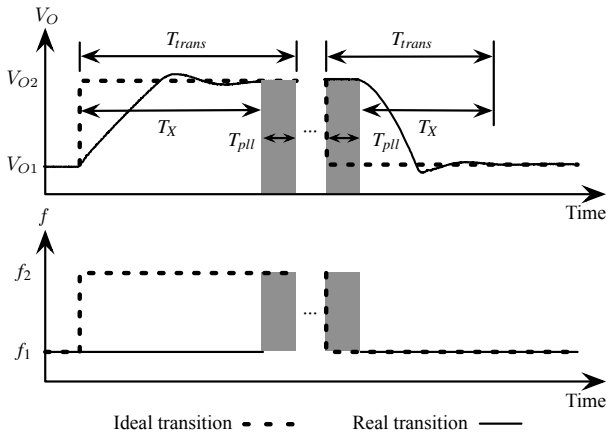

Fig. 8. Definition of an ideal DVFS transition.

Another frequently referred 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 fof operation and a PLL clock generator. Previous works on DVFS scheduling have typically assumed that charging and discharging the bulk capacitor constitute the major portion of DVFS energy overhead. As explained before, this is not a correct assumption, i.e., a significant portion of charge moved into the bulk capacitor can be used to supply power to the processor (load device). The underclocking-related energy loss, which is an energy overhead coming from conservative voltage setting during the voltage transition, as well as an energy loss during the PLL lock time contribute to the total DVFS energy overhead. Furthermore, previous works have assumed the delay overhead to be a constant value or a value that is proportional to the input-output voltage difference. However, not only the PLL lock time, but also the underclocking phenomenon contributes to the DVFS delay overhead. In our proposed formulation, we take all the distinct sources of the overhead into account including losses from the microprocessor and DC–DC converters.

A. Real vs. Ideal Transitions

We define the delay and energy overheads by comparing a real DVFS transition with an ideal one. An ideal transition incurs no time and energy overhead, i.e., the DC–DC converter output voltage and microprocessor frequency are changed instantaneously as shown in Fig. 8. The time and energy overheads, E_O and T_O, are defined as follows.

$$T_O = T_{trans} - T_{trans,id}, \tag{6a}$$

$$E_O = E_{trans} - E_{trans,id}, \tag{6b}$$

where each term is defined in Table I and Fig. 8. Note that T_{trans} is different from the delay overhead since the processor can execute instructions during T_{trans}. This is a simple and obvious way to define the overhead, which leads to following equations.

$$T_{real} = T_{ideal} + \sum T_O, \tag{7a}$$

$$E_{real} = E_{ideal} + \sum E_O, \tag{7b}$$

where T_{real} and T_{ideal} denote the elapsed times to execute a set tasks for the real transition case and the ideal transition case, respectively. Similarly, E_{real} and E_{ideal} denote energy consumptions to execute a set tasks for the real transition case and the ideal transition case, respectively. In other words, real delay and energy consumption values are obtained by summing up their ideal values plus the sum of delay and energy overheads, respectively. In the following sections, we provide the detailed model of the DVFS transition overhead, described by parameters that can be obtained from device datasheets.

B. 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.

$$T_O = T_{uc} + T_{PLL}. \tag{8}$$

Underclocking-related delay overhead: It is the delay overhead due to *underclocking phenomenon* discussed in Section I. For upscaling, the first step for determining the delay overhead is to calculate the T_X value. Parameter T_X is defined as the settling time of the output voltage of the DC–DC converter, where the output settles down into a certain percentage, e.g. 1%, of the target voltage The value of T_{uc} is obtained by comparing the elapsed time between the real DVFS transition and the ideal DVFS transition as defined in (6a). Consider an upscaling DVFS transition from f_s to f_e. In case of a real DVFS transition, the microprocessor operates at a lower operating frequency, f_s, throughout T_X, to guarantee safe operation of the microprocessor. In case of an ideal DVFS transition, the microprocessor would operate at f_e immediately after the transition since there is no delay overhead. The time required for the microprocessor to execute the same number of cycles during an ideal transition is, $T_{trans,id} = \frac{f_s}{f_e} T_X$. Thus, the underclocking-related delay overhead for an upscaling transition is

$$T_{uc,up} = T_X - \frac{f_s}{f_e} T_X = \frac{f_e - f_s}{f_e} T_X. \tag{9}$$

For downscaling, the underclocking-related overhead is 0 since the processor operates at f_e immediately after a DVFS transition is initiated for both the ideal and real case.

$$T_{uc,down} = 0 \tag{10}$$

PLL-induced delay overhead: It is the delay overhead due to PLL lock time. Since the processor halts during the PLL

lock time, T_{PLL} becomes the pure delay overhead of a DVFS transition. From modern literature, we derive the PLL lock time T_{PLL} as a constant which is independent of the present and next clock frequencies, f_s and f_e, as described in [19].

The total delay overhead becomes

$$T_O = \begin{cases} T_{PLL} + \dfrac{f_e - f_s}{f_e} T_X & : \text{upscaling}, \\ T_{PLL} & : \text{downscaling}. \end{cases} \tag{11}$$

C. 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.

$$E_O = E_{conv} + E_{\mu p}. \tag{12}$$

Converter-induced energy overhead: It is the energy overhead induced by the DC–DC converter. During the voltage transition, a large surge current flows into and out of the bulk capacitor via the inductor and MOSFETs as shown in Figs 3 and 4. This causes additional IR losses in the inductor and MOSFETs.

In case of upscaling, additional charge is transferred to the bulk capacitor, and it increases the terminal voltage from V_s to V_e. The amount of energy is described by $E_{cap} = \frac{1}{2} C_b ({V_e}^2 - {V_s}^2)$. The energy drawn from the power supply during an upscaling transition is not yet wasted since it is stored in the bulk capacitor as discussed in Section I. The amount of loss due to this surge current is shown in (13a). Meanwhile, the amount of loss in the DC–DC converter in presence of an ideal transition during $T_{trans,id}$ is (13b), where $I_{O,e}$ is the current draw of the processor with V_e and f_e.

$$E_{conv,up,real} = \int_0^{T_X} R_L I_L(t)^2 dt, \tag{13a}$$

$$E_{conv,,up,ideal} = \int_0^{T_{trans,id}} R_L I_{O,e}^2 dt. \tag{13b}$$

During upscaling, $T_{trans,id}$ is equal to $\frac{f_s}{f_e} T_X$. Thus, the additional inductor IR loss during upscaling is defined as

$$\begin{aligned} E_{conv,up} &= E_{conv,up,real} - E_{conv,up,ideal} \\ &= \int_0^{T_X} R_L I_L(t)^2 dt - \int_0^{\frac{f_s}{f_e} T_X} R_L I_{O,e}^2 dt. \end{aligned} \tag{14}$$

In case of downscaling, the charge drained to the ground from the bulk capacitor causes the energy overhead. All the energy of the drained charge is dissipated as heat in the inductor, and the value $E_{conv,real,down}$ is described as (13a). If the converter operates in discontinuous mode and if the inductor current is zero, $E_{conv,real,down}$ will be zero. Once again, the energy loss of DC–DC converter in presence of an ideal transition should be subtracted.

$$E_{conv,real,down} = \int_0^{T_X} R_L I_L(t)^2 dt, \tag{15a}$$

$$E_{conv,ideal,down} = \int_0^{T_{trans,id}} R_L I_{O,e}^2 dt. \tag{15b}$$

During downscaling, $T_{trans,id}$ is equal to T_X. Thus, the additional inductor IR loss during downscaling is defined as

$$\begin{aligned} E_{conv,down} &= E_{conv,real,down} - E_{conv,ideal,down} \\ &= \int_0^{T_X} R_L (I_L(t)^2 - I_{O,e}^2) dt. \end{aligned} \tag{16}$$

The operation mode of the DC–DC converter, continuous- or discontinous-mode, does not make difference to (14) and (16). It is implied in the term $I_L(t)$.

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

$$E_{conv} = \begin{cases} E_{conv,up} & : \text{upscaling}, \\ E_{conv,down} & : \text{downscaling}. \end{cases} \tag{17}$$

The E_{cap} term used in previous DVFS works is implied in the equations.

Microprocessor-induced energy overhead: As we have stated in the beginning of this section, the microprocessor-induced energy overhead, $E_{\mu p}$, consists of two factors, which are underclocking-related loss, E_{uc}, and PLL lock time loss, E_{pll}. Before we move on to the detailed definition and modeling, we state that a widely known processor power model is used.

$$P_{cpu} = P_{dyn} + P_{sta} = (C_e V_{cpu}^2 f_{cpu}) + (\alpha_1 V_{cpu} + \alpha_2), \tag{18}$$

where P_{cpu}, P_{dyn}, and P_{sta} is the total power consumption, dynamic power consumption, and static power consumption of the target processor, respectively. The term C_e is the average switching capacitance per cycle, and V_{cpu} and f_{cpu} are the supply voltage and the clock frequency of the microprocessor.

Microprocessor underclocking-related loss is caused by underclocking the microprocessor (i.e., applying a conservative clock frequency below the maximum frequency that the supply voltage can safely support) during the transition period as shown in Fig. 7. Because of underclocking, the microprocessor consumes additional dynamic and static power. The underclocking-related loss is calculated by (19) during voltage transition time T_X.

$$\begin{aligned} E_{uc,up} = \; & E_{real} - E_{ideal} \\ = \; & \int_0^{T_X} \left(C_e f_s V_O(t)^2 + \alpha_1 V_O(t) + \alpha_2 \right) dt \\ & - \int_0^{\frac{f_s}{f_e} T_X} \left(C_e f_e V_e^2 + \alpha_1 V_e + \alpha_2 \right) dt, \\ E_{uc,down} = \; & E_{real} - E_{ideal} \\ = \; & \int_0^{T_X} \left(C_e f_e V_O(t)^2 + \alpha_1 V_O(t) + \alpha_2 \right) dt \\ & - \int_0^{T_X} \left(C_e f_e V_e^2 + \alpha_1 V_e + \alpha_2 \right) dt, \end{aligned} \tag{19}$$

Power consumption during the PLL lock time is caused by the static power consumption of the microprocessor during PLL lock time. In general, clock and/or power gating cannot be ideal (without overhead losses), i.e., there is non-zero amount of static power consumption from the microprocessor

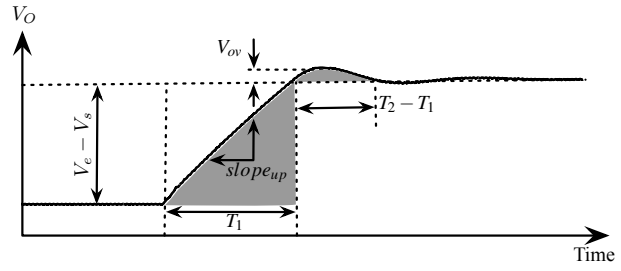

Fig. 9. Approximation of underclocking-related loss.

during the PLL lock time, which is given by

$$\begin{aligned} E_{PLL,up} &= \int_0^{T_{PLL}} (\alpha_1 V_e + \alpha_2)\, dt, \\ E_{PLL,down} &= \int_0^{T_{PLL}} (\alpha_1 V_s + \alpha_2)\, dt. \end{aligned} \tag{20}$$

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

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

$$E_{\mu p} = \begin{cases} E_{uc,up} + E_{PLL,up} & : \text{upscaling}, \\ E_{uc,down} + E_{PLL,down} & : \text{downscaling}. \end{cases} \tag{21}$$

V. Macro Model for DVFS Transition Overhead

A compact macro model, which can be evaluated using datasheet parameters, is crucial for its use in high-level scheduling problems. Although the DVFS transition overhead is precisely formulated in Section IV, it is not easy to obtain the actual values of overhead. The profile of $V_O(t)$ and $I_O(t)$ over time and the value of T_X is required to calculate the delay overhead (11), converter-induced energy overhead (17), and microprocessor-induced energy overhead (21). However, obtaining the exact values of these parameters is not a trivial task. Rather it requires elaborate modeling of the non-linear peak current-mode control of DC–DC converters as we discussed in Section II-B. The details of control parameters are often omitted in the datasheets of DC–DC converters, which further prohibit using such elaborate DVFS transition overhead models. We thus provide an approximate, but much simpler macro model for DVFS transition overhead calculation, which consists of datasheet parameters and RLC values of the DC–DC converter circuit. The symbols used in macro model are defined in Table I.

Macro model for the delay overhead: The value of T_X is required for calculating the value of delay overhead defined in Section IV-B. However, the datasheets of DC–DC converters usually provide the worst-case value of T_X only. Thus, we devise a method to calculate T_X by reprocessing the datasheet parameters. The first step for calculating the T_X is to obtain the slope of the initial voltage increase during T_1, which is 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 $max(I_L)$. Note that this value 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, $max(I_L)$, and current drawn out of the bulk capacitor by the load (processor), I_O. The rate of output voltage change, $slope_{up}$, during voltage upscaling is calculated as follows.

$$slope_{up} = \frac{dV_O}{dt} = \frac{1}{C}(max(I_L) - I_O). \tag{22}$$

The change in I_O during T_X is much smaller than $I_L(t)$. Therefore, without losing much accuracy, we regard it as a constant value $(I_{O,s} + I_{O,e})/2$. We devise a heuristic to approximate T_X using $slope_{up}$. The value of T_X is larger when the difference in V_s and V_e is larger. In addition, T_X shows correlation with the slope of voltage increase, $slope_{up}$. We have found that linearizing the correlation between $slope_{up}$ and T_X provides acceptable accuracy. We thus come up with (23), which implies that $T_X - T_1$ is nearly proportional to the rate of approaching the target voltage, $slope_{up}$.

$$T_X = T_1 + slope_{up} * \beta. \tag{23}$$

The value of β is calculated using the worst case settling time T_X, which is again specified in the datasheet. The worst case T_X occurs when the difference between the initial and final voltages is the largest.

$$T_1 = (V_e - V_s)/slope_{up}, \tag{24a}$$

$$\beta = (T_{X,worst} - T_{1,worst})/slope_{up,worst}, \tag{24b}$$

$$T_{1,worst} = (V_{max} - V_{min})/slope_{up}. \tag{24c}$$

We obtain the underclocking-related delay overhead by substituting (23) into (11).

Macro model for the converter-induced energy overhead: The major hurdle for calculating the converter-induced energy overhead defined in Section IV-C is that of obtaining the trace of $I_L(t)$ over time. For upscaling, we use a similar assumption that the DC–DC converter tries to drive the output to the target voltage as fast as possible, which means that $I_L(t) = max(I_L)$ during T_1. The value of $I_L(t)$ beyond T_1 becomes approximately the same as $I_{O,e} = C_e V_e f_e + \alpha_1 + \alpha_2/V_e$ derived from (18). The integral term including $I_L(t)$ in (14) and (16) then becomes

$$\int_0^{T_X} R_L I_L(t)^2 dt = R_L max(I_L)^2 T_1 + R_L I_{O,e}^2 (T_X - T_1). \tag{25}$$

Substituting (25) into (14) and (16) gives the additional inductor IR loss for upscaling.

For downscaling using continuous mode converter, the controller in the DC–DC converter tries to drive the output voltage to the target value as fast as possible by making the duty ratio of the lower MOSFET equal to 1. We can use this feature to derive the voltage curve during T_1 by solving the RLC circuit with a constant current source as shown in Fig. 10(a).

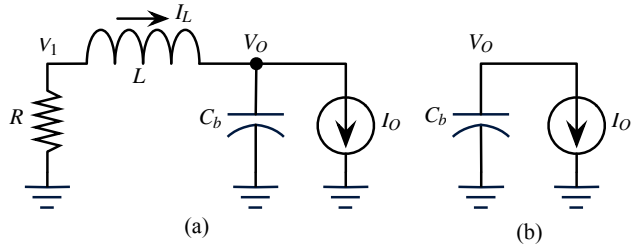

Fig. 10. (a) Circuit for continuous mode downscaling. (b) Circuit for discontinuous mode downscaling.

The traces of $I_L(t)$ and $V_O(t)$ 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 $(I_{O,s}+I_{O,e})/2$ because its change is not dramatic during T_1. R is the summation of whatever resistance exists between the supply and the ground, which consists of the MOSFET on-resistance and inductor resistance. The exact trace of node voltages and inductor current can be obtained by solving the following system of non-homogeneous differential equations.

$$\begin{pmatrix} I_L' \\ V_1' \\ V_O' \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{L} & -\frac{1}{L} \\ 0 & -\frac{R}{L} & \frac{R}{L} \\ \frac{1}{C_b} & 0 & 0 \end{pmatrix} \begin{pmatrix} I_L \\ V_1 \\ V_O \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -\frac{I_O}{C_b} \end{pmatrix}. \tag{26}$$

The closed form solution of $I_L(t)$ and $V_O(t)$ can be obtained by any standard method for solving a system of ordinary differential equations.

For downscaling using discontinuous mode converter, the inductor current is constrained to be non-negative during normal operation. The DC–DC converter controller generally waits for the microprocessor to discharge the bulk capacitor. This involves solving the circuit given in Fig. 10(b).

$$V_O(t) = V_s - \frac{I_O}{C}t. \tag{27}$$

Sometimes the DC–DC converter drains the charge from the bulk capacitor when the voltage difference between the output voltage and target voltage is large. This continues until the error becomes smaller than a certain value, e.g. 0.1 V in case of the LTC3733 converter. For such a case, we solve both (26) and (27), and set the appropriate boundary conditions. The obtained trace of $I_L(t)$ is substituted into (16) to calculate the additional inductor IR loss.

Macro model for the microprocessor-induced energy overhead: The trace of $V_O(t)$ over time is required to calculate the underclocking-related energy overhead (19). However, it is very difficult to calculate the exact value as we discussed in Section II-B. We thus make an approximation as detailed below.

For upscaling, we approximate the integral terms of $\int V_O(t)dt$ and $\int V_O(t)^2 dt$ in (19) by calculating the area of the two shaded triangles shown in Fig. 9. We assume that the integral values beyond T_2, $\int_{T_2}^{T_x}(V_O(t)-V_e)\,dt$ and $\int_{T_2}^{T_x}\left(V_O(t)^2-V_e^2\right)dt$, add up to zero, and thus these terms are ignored.

$$\int_0^{T_X} V_O(t)dt \approx T_X V_e - \frac{1}{2}T_1(V_e - V_s) + \frac{1}{2}(T_2 - T_1)V_{ov}, \tag{28a}$$

$$\int_0^{T_X} V_O(t)^2 dt \approx T_X V_e^2 - \frac{1}{2}T_1(V_e^2 - V_s^2) + \frac{1}{2}(T_2 - T_1)\left((V_e + V_{ov})^2 - V_e^2\right). \tag{28b}$$

T_1 is calculated by (24c). V_{ov}, and T_2 is calculated in a similar way to that of calculating T_X in (23). We linearize the variations V_{ov}, and $T_2 - T_1$ according to the rate of approaching the target voltage $slope_{up}$. Thus, the following equations hold.

$$V_{ov} = \gamma slope_{up}, \tag{29a}$$

$$T_2 = T_1 + \delta slope_{up}. \tag{29b}$$

The values γ and δ are device-dependent parameters, which determine the overshoot and settling time. 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 (29) 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 (26) and (27). The trace of output voltage, $V_O(t)$, is substituted into (19) to obtain the underclocking-related energy loss during T_1. We assume that the voltage ripple beyond T_1 is small enough to cancel the integral terms in $E_{uc,down}$ in (19).

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 emphasize that our macro model and the experimental results are not restricted to specific types of microprocessors, and they are applicable to any other microprocessors exhibiting similar amount of power consumption, supply voltage levels, and clock frequencies. The three representative processors we chose are Intel Core2 Duo, ARM Cortex-A8, and TI MSP430. The Intel Core2 Duo processor shows power consumption higher than 50 W while MSP430 shows power consumption of a few mW. We show the accuracy and generality of the macro model by showing the results for a wide range of processors.

A. Case 1: Intel Core2 Duo E6850 Processor

A high-fidelity DVFS transition overhead model requires detailed microprocessor power consumption information that

TABLE II
VOLTAGE (V_{cpu} (V)) AND CLOCK FREQUENCY (f_{CPU} (GHz)) LEVELS FOR INTEL CORE2 DUO E6850 PROCESSOR.

DVFS level	V_{cpu}	f_{cpu}	DVFS level	V_{cpu}	f_{cpu}
Level 1	1.30	3.074	Level 4	1.15	2.281
Level 2	1.25	2.852	Level 5	1.10	1.932
Level 3	1.20	2.588	Level 6	1.05	1.540

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

$V_{cpu}(V)$	$f_{cpu}(GHz)$	Measurement (W)	Analytical model (W)
1.056	1.776	21.520	21.212
1.080	1.888	24.000	23.956
1.104	2.004	26.320	26.856
1.160	2.338	33.760	34.838
1.224	2.672	43.200	44.409
1.280	3.006	55.440	54.236

reflects the supply voltage and frequency changes. We choose a high-end DVFS-enabled microprocessor, i.e., Intel Core2 Duo E6850 processor, along with the LTC3733 3-phase synchronous step-down DC–DC converter that supports discontinuous mode, which is a representative setup of a modern high-performance DVFS-enabled microprocessor.

The microprocessor power consumption model is described in (18). The parameters C_e, α_1, and α_2 is obtained from actual measurements. We insert a shunt monitor circuit right 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_O. We run PrimeZ benchmark and change V_{cpu} and f_{cpu} performing direct access to the BIOS (basic input/output system) as described in Table II because the Intel SpeedStep supports only two voltage levels. We finally derive the following power consumption model:

$$P_{cpu} = 8.4503 V_{cpu}^2 f_{cpu} + (36.3851 V_{cpu} - 33.9503), \quad (30)$$

where the units of P_{cpu}, V_{cpu}, and f_{cpu} are W, V, and GHz, respectively. 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.

TABLE IV
DC–DC CONVERTER PARAMETERS OF LTC3733 3-PHASE CONVERTER FOR INTEL CORE2 DUO E6850.

Parameter	Value	Parameter	Value
V_{IN}	12 (V)	V_{OUT}	V_O in Table II
C	8840 (μF)	L	1 (μH) per phase
R_L	2.3 (mΩ)	f_{DC}	530 (kHz) per phase
$max(I_L)$	75 (A)		

TABLE V
DVFS TRANSITION DELAY OVERHEAD FOR INTEL CORE2 DUO E6850 PROCESSOR WITH LTC3733 CONVERTER. (E_{pll}=5 μs AND THE NUMBER OF CYCLES ARE CALCULATED AT $f = 3.074\ GHz$).

Level	Actual value (μs)			Proposed model (μs)		
	T_{uc}	Total	Cycles	T_{uc}	Total	Cycles
2→1	4.77	9.77	30018	4.11	9.11	28011
3→1	12.29	17.29	53141	12.21	17.21	52890
3→2	5.95	10.95	33672	5.72	10.72	32950
4→1	22.29	27.29	83894	24.24	29.24	89894
4→2	14.69	19.69	60531	16.21	21.21	65201
4→3	7.33	12.33	37921	8.06	13.06	40150
5→1	34.81	39.81	122389	40.37	45.37	139457
5→2	26.47	31.47	96733	31.44	36.44	112025
5→3	27.33	32.33	99383	21.87	26.87	82606
5→4	9.43	14.43	44361	11.49	16.49	50694
6→1	57.68	62.68	192684	60.90	65.90	202590
6→2	49.50	54.50	167525	51.65	56.65	174157
6→3	31.89	36.89	113409	41.52	46.52	142994
6→4	28.35	33.35	102531	30.14	35.14	108006
6→5	12.14	17.14	52688	16.84	21.84	67141
Downscale	0	5	15370	0	5	15370

The value of T_{pll}, 5 μs, is specified in the Intel Core2 Duo E6850 datasheet. The actual values are obtained from SPICE simulation results. We obtain T_X by observing the settling time of $V_O(t)$ from SPICE results, and substitute it into equations in Section IV to calculate the delay overhead. The estimated overhead from the proposed macro model well follows the trend of actual values. For upscaling, the delay overhead is sum of underclocking-related overhead T_{uc} and PLL lock time loss T_{pll}. Unlike assumption of previous works, the underclocking-related overhead is the dominant factor for most cases as we have discussed in Section III. For downscaling, PLL lock time is the only delay overhead, and thus the overhead values are the same for all cases.

The energy overhead values of a DVFS transition for continuous- and discontinuous-mode operations are given in Tables VI and VII, respectively. For the actual value, we obtain $I_L(t)$, $I_O(t)$, and $V_O(t)$ from SPICE simulation and substitute them into (14), (16), and (19). There is no E_{cap} in the Tables as it is implied in E_{ir}. The value of E_{ir} for the case $1 \rightarrow 6$ in Table VI is large because it drains significant amount of charge from bulk capacitor to the ground. On the other hand, E_{ir} 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.

B. Case 2: ARM Cortex-A8 Processor

The second target DVFS system is the ARM Cortex-A8 processor with LTC3446 converter. ARM Cortex-A8 processor is an application processor targeting high-end mobile products such as smartphones, tablets, and netbooks. It exhibits power consumption of 600 mW at full speed. We perform a procedure

TABLE VI
DVFS TRANSITION ENERGY OVERHEAD OF LTC3733 OPERATING IN CONTINUOUS MODE FOR INTEL CORE2 DUO E6850 PROCESSOR.

Level	Actual value (μJ)			Proposed model (μJ)		
	E_{uc}	E_{ir}	Total	E_{uc}	E_{ir}	Total
1→2	-2.5	-62.8	-51.9	35.4	-14.9	33.9
1→3	57.7	-7.1	64.0	112.7	4.2	130.2
1→4	152.5	177.5	343.4	202.0	119.0	335.3
1→5	246.0	336.8	596.2	293.7	274.2	581.3
1→6	329.3	680.2	1022.8	371.7	467.9	852.9
2→3	-11.5	-31.3	-31.3	33.0	-12.6	31.9
2→4	64.9	41.2	117.6	104.2	18.9	134.6
2→5	146.7	178.1	336.4	185.2	90.5	287.3
2→6	229.2	436.9	677.6	262.1	265.0	538.6
3→4	-1.4	-4.1	4.2	34.9	-6.7	37.9
3→5	65.3	110.1	185.1	94.6	28.0	132.3
3→6	141.1	273.4	424.3	165.7	131.3	306.7
4→5	3.0	22.6	33.5	32.5	-3.6	36.8
4→6	62.1	178.6	248.6	82.3	30.3	120.4
5→6	12.7	59.7	78.5	28.9	-0.3	34.7
2→1	29.0	378.6	420.9	16.3	352.9	382.5
3→1	47.1	734.3	794.7	32.4	671.2	716.9
3→2	43.2	373.5	428.2	40.9	335.6	388.0
4→1	83.1	1054.4	1150.8	91.7	951.8	1056.9
4→2	82.1	707.8	801.4	91.9	634.6	738.0
4→3	49.3	340.5	399.5	64.0	317.3	391.0
5→1	155.6	1352.1	1521.1	216.7	1192.6	1422.6
5→2	140.6	1014.1	1166.2	190.0	894.4	1095.9
5→3	192.9	689.5	892.1	144.9	596.3	750.9
5→4	58.6	315.0	381.5	85.1	298.1	391.1
6→1	388.6	1635.5	2037.5	423.1	1391.4	1827.8
6→2	331.5	1312.4	1655.4	354.9	1113.1	1479.5
6→3	184.9	966.4	1161.0	276.8	834.8	1121.4
6→4	174.1	672.6	854.6	191.9	556.6	756.3
6→5	61.7	276.3	344.1	103.7	278.3	388.1

End level (upscale) Start level (downscale)	1	2	3	4	5
E_{pll} (μJ)	66.8	57.7	48.6	39.5	30.4

TABLE VII
DVFS TRANSITION ENERGY OVERHEAD OF LTC3733 OPERATING IN DISCONTINUOUS MODE FOR INTEL CORE2 DUO E6850 PROCESSOR.

Level	Actual Value (μJ)			Proposed model (μJ)		
	E_{uc}	E_{ir}	Total	E_{uc}	E_{ir}	Total
1→2	-10.6	-88.7	-85.9	0.5	-252.5	-238.7
1→3	74.0	-179.5	-92.1	116.3	-268.4	-138.8
1→4	237.9	-268.5	-17.3	268.2	-325.3	-43.8
1→5	376.3	-230.0	159.7	386.5	-248.2	151.7
1→6	478.0	-32.1	459.2	509.7	17.1	540.1
2→3	-14.5	-195.0	-197.9	0.5	-287.2	-275.2
2→4	94.3	-159.3	-53.5	124.9	-231.6	-95.2
2→5	248.4	-217.7	42.2	275.0	-260.4	26.2
2→6	375.8	-125.8	261.5	383.6	-139.1	256.0
3→4	1.9	-55.7	-44.1	0.5	-215.8	-205.6
3→5	106.0	-132.5	-16.7	138.2	-193.0	-45.2
3→6	266.8	-158.3	118.2	284.7	-187.0	107.4
4→5	10.1	-47.9	-29.9	0.5	-177.3	-168.9
4→6	126.0	-106.6	27.2	161.1	-153.1	15.9
5→6	21.5	-39.5	-11.9	0.5	-137.4	-130.8
Upscale	The same as Table VI.					

TABLE VIII
VOLTAGE (V_{cpu} (V)) AND CLOCK FREQUENCY (f_{CPU} (MHZ)) LEVELS FOR ARM CORTEX-A8 PROCESSOR.

DVFS level	V_{cpu}	f_{cpu}	DVFS level	V_{cpu}	f_{cpu}
Level 1	1.35	600	Level 4	1.10	250
Level 2	1.25	550	Level 5	1.00	125
Level 3	1.20	500			

TABLE IX
DC–DC CONVERTER PARAMETERS OF LTC3446 CONVERTER FOR ARM CORTEX-A8.

Parameter	Value	Parameter	Value
V_{IN}	5 (V)	V_{OUT}	V_O in Table VIII
C	22 (μF)	L	1 (μH)
R_L	1 (mΩ)	f_{DC}	2.25 (MHz)
$max(I_L)$	1 (A)		

similar to that for Intel Core2 Duo E6850 processor to model the power consumption of ARM Cortex-A8 processor. The resulting equation is as follows.

$$P_{cpu} = 0.4913 V_{cpu}^2 f_{cpu} + (0.09614 V_{cpu} - 0.08187), \quad (31)$$

where the units of P_{cpu}, V_{cpu}, and f_{cpu} are W, V, and GHz, respectively. The parameters for DC–DC converters are shown in Table IX. The values are taken from datasheet and the reference design provided by the vendor.

Table X shows the DVFS transition delay overhead for the target system. The value of T_{pll}, 10 μs, is obtained from TI OMAP3530 datasheet which is based on ARM Cortex-A8. The underclocking-related overhead is higher when the change in voltage is large. Table XI shows the DVFS transition energy overhead for the target system. Unlike LTC3733, LTC3446 operates in discontinuous-mode only. There is energy gain (minus overhead) in E_{ir} for downscaling because the inductor current is fixed to 0, and no IR loss occurs when compared with the ideal transition.

TABLE X
DVFS TRANSITION DELAY OVERHEAD OF ARM CORTEX-A8 WITH LTC3446 CONVERTER. ($T_{pll} = 10\mu s$ AND THE NUMBER OF CYCLES IS CALCULATED AT $f = 600\,MHz$)

Level	Actual (μs)			Model (μs)		
	T_{uc}	Total	Cycles	T_{uc}	Total	Cycles
2→1	4.3	14.3	8586	5.5	15.5	9286
3→1	11.5	21.5	12912	12.2	22.2	13303
3→2	4.5	14.5	8685	6.4	16.4	9842
4→1	60.2	70.2	42124	54.6	64.6	38748
4→2	46.3	56.3	33765	49.9	59.9	35946
4→3	39.0	49.0	29378	44.8	54.8	32909
5→1	119.8	129.8	77890	81.0	91.0	54581
5→2	106.9	116.9	70117	77.6	87.6	52539
5→3	104.0	114.0	68402	74.0	84.0	50423
5→4	23.1	33.1	19873	47.6	57.6	34549

TABLE XI
DVFS TRANSITION ENERGY OVERHEAD FOR ARM CORTEX-A8 WITH LTC3446 CONVERTER.

Level	Actual Value			Proposed model		
	E_{uc} (nJ)	E_{ir} (nJ)	Total (nJ)	E_{uc} (nJ)	E_{ir} (nJ)	Total (nJ)
1→2	187	-939	-350	145	-683	-136
1→3	297	-1296	-664	509	-1140	-296
1→4	2056	-1184	1062	2138	-1158	1171
1→5	3960	-804	3274	3684	-802	3000
2→3	73	-675	-266	113	-514	-66
2→4	1090	-836	445	1179	-814	556
2→5	2452	-602	1968	2369	-599	1889
3→4	429	-548	72	561	-534	218
3→5	1385	-435	1069	1430	-434	1115
4→5	113	-131	101	175	-128	166
2→1	242	2285	3006	256	2422	3157
3→1	435	4403	5317	410	4369	5258
3→2	223	1827	2452	287	2039	2729
4→1	2385	7943	10808	2243	7579	10300
4→2	1637	5392	7432	1847	5558	7807
4→3	1168	3153	4656	1468	3789	5592
5→1	5396	9278	15154	3540	8853	12873
5→2	4014	6580	10996	2935	6965	10302
5→3	3239	4399	7973	2397	5312	8044
5→4	3309	2529	6029	943	1771	2905

End level (upscale) / Start level (downscale)	1	2	3	4
E_{pll} (nJ)	402	335	191	119

TABLE XII
VOLTAGE (V_{cpu} (V)) AND CLOCK FREQUENCY (f_{CPU} (MHz)) LEVELS FOR TI MSP430 MICROCONTROLLER.

DVFS level	V_{cpu}	f_{cpu}	DVFS level	V_{cpu}	f_{cpu}
Level 1	3.3	8	Level 4	2.175	5
Level 2	2.925	7	Level 5	1.8	4
Level 3	2.55	6			

C. Case 3: TI MSP430 Microcontroller

The third target system is the TI MSP430 microcontroller. TI MSP430 is a microcontroller used for ultra low-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 ARM Cortex-A8 is performed to obtain the following power model.

$$P_{cpu} = 0.1128 V_{cpu}^2 f_{cpu} + (0.1738 V_{cpu} - 0.2832), \qquad (32)$$

TABLE XIII
DC–DC CONVERTER PARAMETERS OF LTC3620 CONVERTER FOR TI MSP430 MICROCONTROLLER.

Parameter	Value	Parameter	Value
V_{IN}	3.6 (V)	V_{OUT}	V_O in Table XII
C	1 (μF)	L	22 (μH)
R_L	1 (mΩ)	f_{DC}	Variable (PFM)
$max(I_L)$	15 (mA)		

TABLE XIV
DVFS DELAY OVERHEAD OF DC–DC CONVERTERS FOR TI MSP430 MICROCONTROLLER WITH LTC3620. (THE NUMBER OF CYCLES IS CALCULATED AT $f = 8\ MHz$)

Level	Actual value		Proposed model	
	T_{uc} (μs)	Cycles	T_{uc} (μs)	Cycles
2→1	79.4	635	214	1712
3→1	382.3	3058	478.4	3827
3→2	103.4	827	254.3	2035
4→1	706	5648	786.2	6290
4→2	477.7	3822	562.3	4499
4→3	139	1112	306.7	2453
5→1	1116.1	8929	1132	9056
5→2	879.1	7033	917	7336
5→3	623.8	4990	671.8	5374
5→4	278.5	2228	378.2	3026

where the units of P_{cpu}, V_{cpu}, and f_{cpu} are mW, V, and MHz, respectively. We use LTC3620 converter to power the target processor. The parameters for the DC–DC converter are reported in Table XIII.

Table XIV shows the DVFS transition delay overhead for the target system. There is no overhead due to PLL lock time T_{pll} because TI MSP430 uses digitally controlled oscillator (DCO) instead of PLL, which is a improved variation of VCO. The underclocking-related overhead T_{uc} is the only delay overhead for TI MSP430 microcontroller. Table XV shows the DVFS transition energy overhead for the target system. LTC3620 is designed for low-power applications. It is PFM controlled and operates in discontinuous mode only. The IR loss for TI MSP430 processor is very small due to small load current. The underclocking-related overhead is relatively large because of the long voltage settling time due to simple control method.

The result of the proposed model is inaccurate for some cases when the difference in the initial and final voltage levels is small because our model does not take into account all the details during the voltage transition period. However, the result of the proposed model follows the general trend well.

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 dynamic thermal management (DTM). DVFS is a very useful control knob for dynamic thermal management (DTM) [25], [26]. 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.

TABLE XV
DVFS TRANSITION ENERGY OVERHEAD OF TI MSP430 MICROCONTROLLER WITH LTC3620.

Level	Actual Value			Proposed model		
	E_{uc} (nJ)	E_{ir} (nJ)	Total (nJ)	E_{uc} (nJ)	E_{ir} (nJ)	Total (nJ)
1→2	720.9	-4.1	716.8	649.2	-4.5	644.7
1→3	2565.6	-6.8	2558.8	2547.8	-7.3	2540.5
1→4	5413.9	-8.3	5405.6	5619.8	-8.5	5611.3
1→5	9116.3	-8.4	9107.9	9802.4	-8.6	9793.8
2→3	701.0	-2.4	698.6	652.1	-3.4	648.7
2→4	2515.6	-5.0	2510.6	2562.1	-5.3	2556.8
2→5	5371.7	-5.8	5365.9	5674.2	-5.9	5668.3
3→4	694.0	-2.3	691.7	659.9	-2.4	657.5
3→5	2530.7	-3.5	2527.2	2610.0	-3.6	2606.4
4→5	689.0	-1.5	687.5	680.4	-1.6	678.8
2→1	60.9	75.6	136.5	418.2	56.3	474.5
3→1	300.8	151.8	452.6	188.6	108.2	296.8
3→2	29.2	72.1	101.3	374.6	54.1	428.7
4→1	177.7	197.0	374.7	-57.8	157.8	100
4→2	245.6	140.5	386.1	209.4	104.5	313.9
4→3	39.4	67.0	106.4	313.2	52.3	365.5
5→1	-18.1	232.0	213.9	-232.1	203.0	-29.1
5→2	108.4	178.1	286.5	47.7	152.2	199.9
5→3	204.0	125.4	329.4	195.8	101.5	297.3
5→4	75.9	65.3	141.2	240.0	50.7	290.7

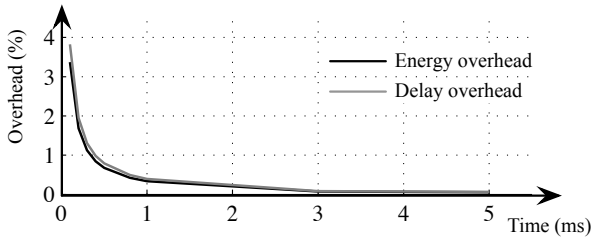

Fig. 11. Energy and delay overhead of PID control based DTM for Intel Core2 Duo E6850 processor according to time granularity of DTM.

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 $R = 0.7\ K/W$ and thermal capacitance of the chip is $C = 140.3\ J/K$, which is the same as [25]. The thermal RC constant is 98.21 seconds. Fig. 11 shows the delay and energy overhead of DVFS according to the time granularity of DTM for Intel Core2 Duo E6850 processor. The results show that we should avoid using time quantum value below 1 ms for performance and energy efficiency. The energy and delay overhead is comparable to the scheduling overhead and context switching overhead of operating systems, which take about 0.4% to 1.6% in general purpose operating systems [27].

VIII. CONCLUSIONS

Dynamic voltage and frequency (DVFS) scaling 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 so far due to absence of correct and accurate models.

This paper is the first paper that introduces correct and accurate DVFS transition overhead models. We show 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 introduce energy and delay overhead caused by microprocessor underclocking and additional current through the inductor. This paper provides comprehensive solutions for the models, but the derived model is somewhat complicated for system engineers. We also provide 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, such as Intel Core2 Duo E6850, Cortex-A8 and TI MSP430 so that some software programmers may simply use the numbers.

REFERENCES

[1] *Intel Core2 Extreme Processor QX9000 and Intel Core2 Quad Processor Q9000, Q9000S, Q8000 and Q8000S Series Datasheet*, 2009.

[2] Y.-H. Lu and G. De Micheli, "Comparing system level power management policies," *Design Test of Computers, IEEE*, vol. 18, no. 2, pp. 10 –19, mar/apr 2001.

[3] T. Ishihara and H. Yasuura, "Voltage scheduling problem for dynamically variable voltage processors," in *Proceedings of the 1998 international symposium on Low power electronics and design*, ser. ISLPED '98. New York, NY, USA: ACM, 1998, pp. 197–202. [Online]. Available: http://doi.acm.org/10.1145/280756.280894

[4] W. Kim, J. Kim, and S. L. Min, "Preemption-aware dynamic voltage scaling in hard real-time systems," in *Low Power Electronics and Design, 2004. ISLPED '04. Proceedings of the 2004 International Symposium on*, aug. 2004, pp. 393 – 398.

[5] Z. Cao, B. Foo, L. He, and M. van der Schaar, "Optimality and improvement of dynamic voltage scaling algorithms for multimedia applications," in *Proceedings of the 45th annual Design Automation Conference*, ser. DAC '08. New York, NY, USA: ACM, 2008, pp. 179–184. [Online]. Available: http://doi.acm.org/10.1145/1391469.1391516

[6] T. D. Burd and R. W. Brodersen, "Design issues for dynamic voltage scaling," in *Proceedings of the 2000 international symposium on Low power electronics and design*, ser. ISLPED '00. New York, NY, USA: ACM, 2000, pp. 9–14. [Online]. Available: http://doi.acm.org/10.1145/344166.344181

[7] S. M. Martin, K. Flautner, T. Mudge, and D. Blaauw, "Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads," in *Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design*, ser. ICCAD '02. New York, NY, USA: ACM, 2002, pp. 721–725. [Online]. Available: http://doi.acm.org/10.1145/774572.774678

[8] X. Zhang, A. Bermak, and F. Boussaid, "Dynamic voltage and frequency scaling for low-power multi-precision reconfigurable multiplier," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, 30 2010-june 2 2010, pp. 45 –48.

[9] M. Schmitz, B. Al-Hashimi, and P. Eles, "Energy-efficient mapping and scheduling for dvs enabled distributed embedded systems," in *Proceedings of the conference on Design, automation and test in Europe*, ser. DATE '02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 514–. [Online]. Available: http://dl.acm.org/citation.cfm?id=882452.874328

[10] *Voltage Regulator Module (VRM) and Enterprise Voltage Regulator-Down (EVRD) 11.1*, 2009.

[11] J. Park, D. Shin, M. Pedram, and N. Chang, "Accurate modeling and calculation of delay and energy overheads of dynamic voltage scaling in modern high-performance microprocessors," in *Proceedings of Proceedings of IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, August 2010, pp. 419–424.

[12] R. Ridley, "A new, continuous-time model for current-mode control [power convertors]," *Power Electronics, IEEE Transactions on*, vol. 6, no. 2, pp. 271 –280, apr 1991.

[13] B. Bryant and M. Kazimierczuk, "Modeling the closed-current loop of pwm boost dc-dc converters operating in ccm with peak current-mode control," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 52, no. 11, pp. 2404 – 2412, nov. 2005.

[14] *LTSPICE. www.linear.com*.

[15] T. Sakurai and A. Newton, "Alpha-power law mosfet model and its applications to cmos inverter delay and other formulas," *Solid-State Circuits, IEEE Journal of*, vol. 25, no. 2, pp. 584 –594, apr 1990.

[16] "Introduction to intel core duo processor architecture," in *Intel Technology Journal*, vol. 10, 2006, pp. 89 –97.

[17] S. Lee and T. Sakurai, "Run-time voltage hopping for low-power real-time systems," in *Design Automation Conference, 2000. Proceedings 2000. 37th*, 2000, pp. 806 –809.

[18] A. Bashir, J. Li, K. Ivatury, N. Khan, N. Gala, N. Familia, and Z. Mohammed, "Fast lock scheme for phase-locked loops," in *Custom Integrated Circuits Conference, 2009. CICC '09. IEEE*, sept. 2009, pp. 319 –322.

[19] J. Pouwelse, K. Langendoen, and H. Sips, "Dynamic voltage scaling on a low-power microprocessor," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, ser. MobiCom '01. New York, NY, USA: ACM, 2001, pp. 251–259. [Online]. Available: http://doi.acm.org/10.1145/381677.381701

[20] C. Lichtenau, M. Ringler, T. Pfluger, S. Geissler, R. Hilgendorf, J. Heaslip, U. Weiss, P. Sandon, N. Rohrer, E. Cohen, and M. Canada, "Powertune: advanced frequency and power scaling on 64b powerpc microprocessor," in *Solid-State Circuits Conference, 2004. Digest of Technical Papers. ISSCC. 2004 IEEE International*, feb. 2004, pp. 356 – 357 Vol.1.

[21] B. Mochocki, X. S. Hu, and G. Quan, "A realistic variable voltage scheduling model for real-time applications," in *Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design*, ser. ICCAD '02. New York, NY, USA: ACM, 2002, pp. 726–731. [Online]. Available: http://doi.acm.org/10.1145/774572.774679

[22] P. Schaumont, B.-C. C. Lai, W. Qin, and I. Verbauwhede, "Cooperative multithreading on embedded multiprocessor architectures enables energy-scalable design," in *Design Automation Conference, 2005. Proceedings. 42nd*, june 2005, pp. 27 – 30.

[23] D. Shin and J. Kim, "Optimizing intratask voltage scheduling using profile and data-flow information," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, no. 2, pp. 369 – 385, feb. 2007.

[24] P. Pillai and K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," in *Proceedings of the eighteenth ACM symposium on Operating systems principles*, ser. SOSP '01. New York, NY, USA: ACM, 2001, pp. 89–102. [Online]. Available: http://doi.acm.org/10.1145/502034.502044

[25] S. Zhang and K. Chatha, "Approximation algorithm for the temperature-aware scheduling problem," in *Computer-Aided Design, 2007. ICCAD 2007. IEEE/ACM International Conference on*, nov. 2007, pp. 281 –288.

[26] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, "Temperature-aware microarchitecture: Modeling and implementation," *ACM Trans. Archit. Code Optim.*, vol. 1, no. 1, pp. 94–125, Mar. 2004. [Online]. Available: http://doi.acm.org/10.1145/980152.980157

[27] D. Tsafrir, "The context-switch overhead inflicted by hardware interrupts (and the enigma of do-nothing loops)," in *Proceedings of the 2007 workshop on Experimental computer science*, ser. ExpCS '07. New York, NY, USA: ACM, 2007. [Online]. Available: http://doi.acm.org/10.1145/1281700.1281704