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.

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

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 20120005640. The ICT at Seoul National University provided research facilities for this paper. Preliminary version of this paper have been presented in part in the International Symposium on Lower Power Electronics and Design (ISLPED) 2010 [1]. This paper was recommended by Associate Editor J. Cortadella.

S. Park, J. Park, and N. Chang are with Seoul National University, Seoul 151-744, Korea (e-mail: sypark@elpl.snu.ac.kr; jhpark@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: yanzhiwa@usc.edu; xqing@usc.edu; pedram@usc.edu).

Digital Object Identifier 10.1109/TCAD.2012.2235126

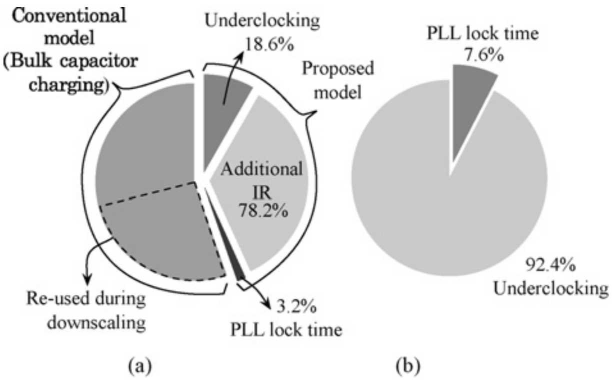


Fig. 1. Breakdown of an upscaling DVFS transition overhead in energy and delay. (Intel Core2 Duo E6850, upscaling from 1.05 to 1.3-V). (a) Energy overhead. (b) Delay overhead.

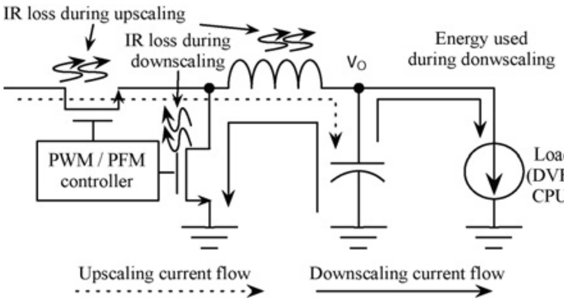


Fig. 2. Upscaling and downscaling current flows.

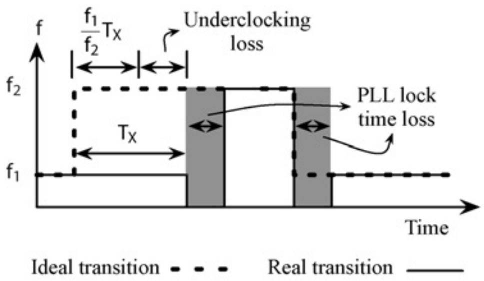


Fig. 3. Concept of underclocking loss and PLL lock time loss in a DVFS transition.

follows.

1) 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 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).
2) 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 as shown in Fig. 2. The additionally stored energy to the bulk capacitor during the upscaling is eventually used by the load during the next downscaling.
3) Voltage upscaling in a conventional DC–DC converter requires more current to be fed through the inductor to increase the bulk capacitor voltage as shown in Fig. 2. This in turn results in additional IR loss from the inductor (78.2% of total energy overhead as shown in Fig. 1).
4) 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 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 E680, 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

$$P_{\mathrm{cdt}} = {I_O}^2 \cdot \Big(D \cdot R_{\mathrm{SW1}} + (1-D) \cdot R_{\mathrm{SW2}} + R_L\Big) + \frac{1}{3} \cdot \left(\frac{\Delta I_L}{2}\right)^2 \cdot \Big(D \cdot R_{\mathrm{SW1}} + (1-D) \cdot R_{\mathrm{SW2}} + R_L + R_C\Big) \quad (1)$$

where I_O, 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

$$P_{\mathrm{sw}} = V_I \cdot f_s \cdot (Q_{\mathrm{SW1}} + Q_{\mathrm{SW2}}) \quad (2)$$

where V_I, f_s, Q_{SW1}, and Q_{SW2} are input voltage, switching frequency, gate charges of the upper and lower MOSFET,

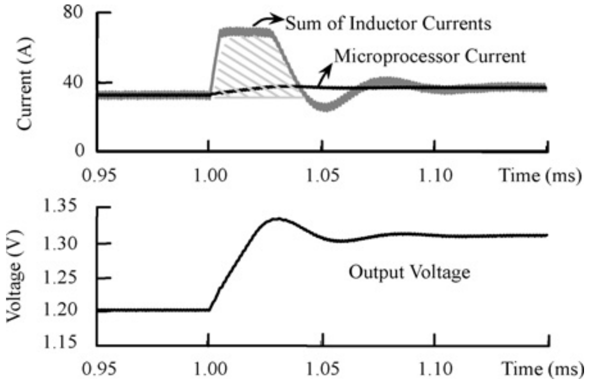


Fig. 4. SPICE simulation result of an upscaling transition (Level 3 → Level 1).

respectively. Controller loss is power loss in the PWM/PFM 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_c 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) VID 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.

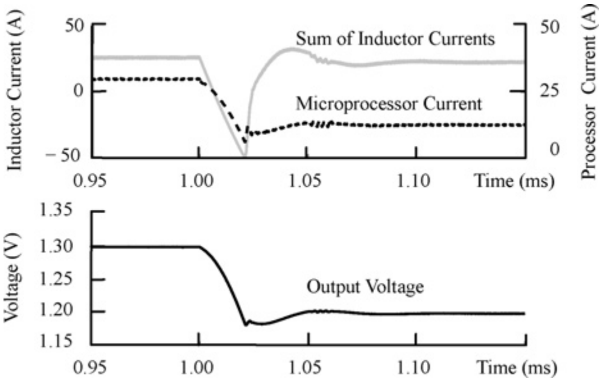


Fig. 5. SPICE simulation result of an continuous-mode downscaling transition (Level 1 → Level 3).

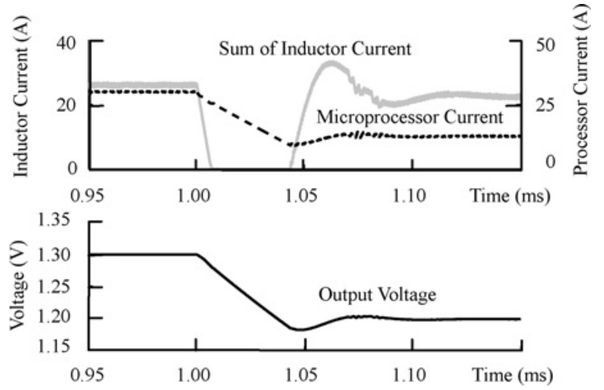


Fig. 6. SPICE simulation result of a discontinuous-mode downscaling transition (Level 1 → Level 3).

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 through the lower MOSFET. Such active discharging operation results in significant energy loss but a faster voltage transition time. Fig. 5 demonstrates that a continuous-mode downscaling transition stabilizes in 40 μs, during which the bulk capacitor is actively discharged to GND (by flow of negative inductor current).

3) *Downscaling Transition Sequence Using Discontinuous Mode DC–DC Converters:* Fig. 6 illustrates that a discontinuous-mode downscaling turns off the lower MOSEFT as soon as the inductor current becomes negative, which prevents the bulk capacitor from further discharging. 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. 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.

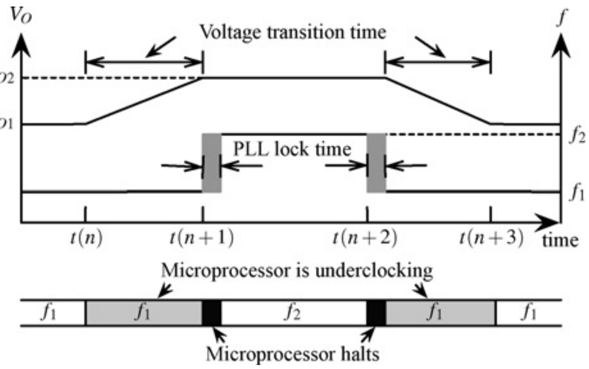


Fig. 7. DVFS upscaling and downscaling for a PLL.

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].

TABLE I
NOTATION FOR DVFS TRANSITION OVERHEAD DEFINITION AND MODELING

T_X	Time to complete a voltage transition defined by settling time of output voltage
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}	Delay overhead due to underclocking
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
$T_{\text{trans,id}}$	Time to execute equivalent number of instructions when an ideal transition takes place
E_{trans}	Energy consumption of all components during T_{trans}
$E_{\text{trans,id}}$	Energy consumption of all components during $T_{\text{trans,id}}$ 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

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 on 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

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

$$E_X = (1-\eta)C_b|{V_e}^2 - {V_s}^2| \tag{4}$$

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 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

$$T_X{}^* = \frac{2C_b}{\max(I_L) - I_O}|V_e - V_s|. \tag{5}$$

Because the microprocessor continues to operate even during the voltage transition, I_O 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 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 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:

$$E_X{}^* = \begin{cases} \dfrac{1}{2\eta}C_b({V_e}^2 - {V_s}^2) & : \text{upscaling} \\ 0 & : \text{downscaling.} \end{cases} \tag{6}$$

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.)

$$E_X{}^{**} = \frac{1}{4\eta}C_b|{V_e}^2 - {V_s}^2|. \tag{7}$$

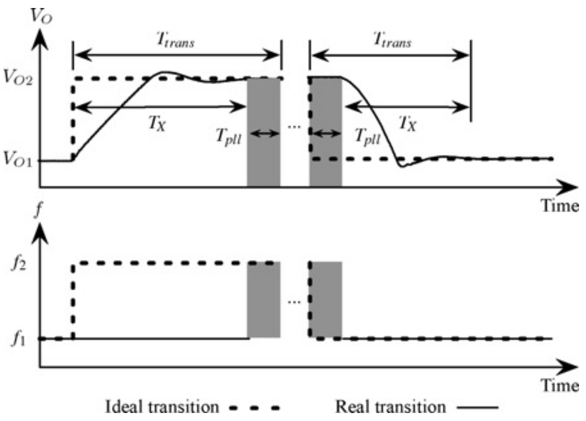


Fig. 8. Definition of an ideal DVFS transition.

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:

$$T_O = T_{\text{trans}} - T_{\text{trans,id}}. \tag{8a}$$

$$\tag{8b}$$

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

$$E_O = E_{\text{trans}} - E_{\text{trans,id}}. \tag{8c}$$

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:

$$T_O = T_{\text{uc}} + T_{\text{PLL}}. \tag{9}$$

1) *Underclocking-Related Delay Overhead:* We obtain T_{uc} 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 $T_{\text{trans,id}} = \dfrac{f_s}{f_e}T_X$. Thus,

the underclocking-related delay overhead for an upscaling transition is

$$T_{\mathrm{uc,up}} = T_X - \frac{f_s}{f_e}T_X = \frac{f_e - f_s}{f_e}T_X. \tag{10}$$

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 cases

$$T_{\mathrm{uc,down}} = 0. \tag{11}$$

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_{PLL} becomes the pure delay overhead of a DVFS transition. We denote T_{PLL} lock time as a constant without losing generality. Some recent PLL architectures have constant PLL lock time independent of f_s and f_e, 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

$$T_O = \begin{cases} T_{\mathrm{PLL}} + \dfrac{f_e - f_s}{f_e}T_X & : \text{ upscaling} \\ T_{\mathrm{PLL}} & : \text{ downscaling.} \end{cases} \tag{12}$$

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:

$$E_O = E_{\mathrm{conv}} + E_{\mu p}. \tag{13}$$

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_s to V_e. The amount of stored energy is $E_{\mathrm{cap}} = \frac{1}{2}C_b({V_e}^2 - {V_s}^2)$. 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 $T_{\mathrm{trans,id}}$ is (14b), where $I_{O,e}$ is the current draw of the processor with V_e and f_e:

$$E_{\mathrm{conv,up,real}} = \int_0^{T_X} R_L I_L(t)^2 dt, \tag{14a}$$

$$E_{\mathrm{conv,up,ideal}} = \int_0^{T_{\mathrm{trans,id}}} R_L {I_{O,e}}^2 dt. \tag{14b}$$

As $T_{\mathrm{trans,id}} = \dfrac{f_s}{f_e}T_X$ during upscaling, the energy overhead due to additional inductor IR loss during upscaling is calculated as

$$\begin{aligned} E_{\mathrm{conv,up}} &= E_{\mathrm{conv,up,real}} - E_{\mathrm{conv,up,ideal}} \\ &= \int_0^{T_X} R_L I_L(t)^2 dt - \int_0^{\frac{f_s}{f_e}T_X} R_L {I_{O,e}}^2 dt. \end{aligned} \tag{15}$$

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

$$E_{\mathrm{conv,real,down}} = \int_0^{T_X} R_L I_L(t)^2 dt, \tag{16a}$$

$$E_{\mathrm{conv,ideal,down}} = \int_0^{T_{\mathrm{trans,id}}} R_L {I_{O,e}}^2 dt. \tag{16b}$$

As $T_{\mathrm{trans,id}} = T_X$ during downscaling, the additional inductor IR loss during downscaling is defined as

$$\begin{aligned} E_{\mathrm{conv,down}} &= E_{\mathrm{conv,real,down}} - E_{\mathrm{conv,ideal,down}} \\ &= \int_0^{T_X} R_L (I_L(t)^2 - {I_{O,e}}^2) dt. \end{aligned} \tag{17}$$

The operation mode of the DC–DC converter, continuous- or discontinous-modes, does not make difference to (15) and (17). It is implied in the term $I_L(t)$.

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

$$E_{\mathrm{conv}} = \begin{cases} E_{\mathrm{conv,up}} & : \text{ upscaling} \\ E_{\mathrm{conv,down}} & : \text{ downscaling.} \end{cases} \tag{18}$$

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

2) *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}. We use a widely known processor power model as follows:

$$P_{\mathrm{cpu}} = P_{\mathrm{dyn}} + P_{\mathrm{sta}} = (C_e {V_{\mathrm{cpu}}}^2 f_{\mathrm{cpu}}) + (\alpha_1 V_{\mathrm{cpu}} + \alpha_2) \tag{19}$$

where P_{cpu}, P_{dyn}, and P_{sta} are 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. The parameter C_e 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_e, α_1, and α_2 using offline regression analysis for each value of processor utilization. Static power is rather independent of application-specific parameters and thus α_1 and α_2 remain almost the same. This results in the following relationship:

$$C_e \propto \upsilon_{\text{proc}} = \upsilon_{\text{kernel}} + \upsilon_{\text{user}} \tag{20}$$

where υ_{proc}, υ_{kernel}, and υ_{user} 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_e is selected. For the very short period of time a DVFS transition takes place, we assume that C_e 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_X

$$\begin{aligned} E_{\text{uc,up}} = {} & E_{\text{real}} - E_{\text{ideal}} \\ = {} & \int_0^{T_X} \left(C_e f_s V_O(t)^2 + \alpha_1 V_O(t) + \alpha_2\right) dt \\ & - \int_0^{\frac{f_s}{f_e} T_X} \left(C_e f_e {V_e}^2 + \alpha_1 V_e + \alpha_2\right) dt \\ E_{\text{uc,down}} = {} & E_{\text{real}} - E_{\text{ideal}} \\ = {} & \int_0^{T_X} \left(C_e f_e V_O(t)^2 + \alpha_1 V_O(t) + \alpha_2\right) dt \\ & - \int_0^{T_X} \left(C_e f_e {V_e}^2 + \alpha_1 V_e + \alpha_2\right) dt \end{aligned} \tag{21}$$

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

$$\begin{aligned} E_{\text{PLL,up}} &= \int_0^{T_{\text{PLL}}} (\alpha_1 V_e + \alpha_2)\, dt \\ E_{\text{PLL,down}} &= \int_0^{T_{\text{PLL}}} (\alpha_1 V_s + \alpha_2)\, dt. \end{aligned} \tag{22}$$

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

$$E_{\mu p} = \begin{cases} E_{\text{uc,up}} + E_{\text{PLL,up}} & : \text{ upscaling,} \\ E_{\text{uc,down}} + E_{\text{PLL,down}} & : \text{ downscaling.} \end{cases} \tag{23}$$

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 $V_O(t)$ and $I_O(t)$ over time and the value of T_X. 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 datasheet 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.

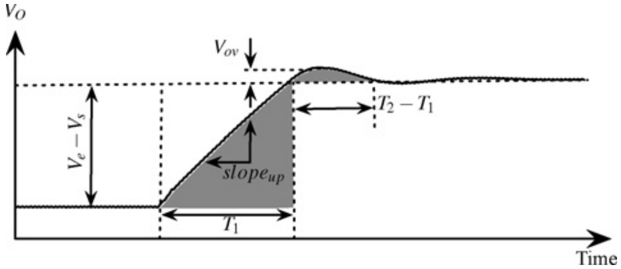


Fig. 9. Approximation of underclocking-related loss.

A. Macromodel for the Delay Overhead

Datasheets of DC–DC converters usually provide the worst-case value of T_X only. We propose to calculate the T_X 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 $\max(I_L)$, 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 $\max(I_L)$ 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:

$$\text{slope}_{\text{up}} = \frac{dV_O}{dt} = \frac{1}{C}(\max(I_L) - I_O). \tag{24}$$

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 derive (25), which implies that $T_X - T_1$ is nearly proportional to the rate of approaching the target voltage, slope_{up}

$$T_X = T_1 + \text{slope}_{\text{up}} \cdot \beta. \tag{25}$$

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:

$$T_1 = (V_e - V_s)/\text{slope}_{\text{up}} \tag{26a}$$

$$\beta = (T_{X,\text{worst}} - T_{1,\text{worst}})/\text{slope}_{\text{up,worst}} \tag{26b}$$

$$T_{1,\text{worst}} = (V_{\max} - V_{\min})/\text{slope}_{\text{up}}. \tag{26c}$$

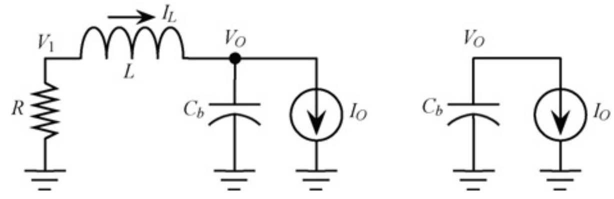


Fig. 10. Equivalent circuit for a downscale DVFS transition. (a) Continuous mode downscaling. (b) Discontinuous mode downscaling.

We obtain the underclocking-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 $I_L(t)$ over time. We use a similar assumption for upscaling such 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 (19). The integral term including $I_L(t)$ in (15) and (17) then becomes

$$\int_0^{T_X} R_L I_L(t)^2 dt = R_L \max(I_L)^2 T_1 + R_L I_{O,e}{}^2 (T_X - T_1). \quad (27)$$

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 $I_L(t)$ 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 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:

$$\begin{pmatrix} I_L' \\ V_1' \\ V_O' \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{L} & -\frac{1}{L} \\ 0 & -\frac{R}{L} & \frac{R}{L} \\ \frac{1}{C_b} & 0 & 0 \end{pmatrix} \begin{pmatrix} I_L \\ V_1 \\ V_O \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -\frac{I_O}{C_b} \end{pmatrix}. \quad (28)$$

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

$$V_O(t) = V_s - \frac{I_O}{C} t. \quad (29)$$

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 $I_L(t)$ is substituted into (17) to calculate the additional inductor IR loss.

1) *Macromodel for the Microprocessor-Induced Energy Overhead::* We need the trace of $V_O(t)$ over time to calculate the underclocking-related energy overhead (21). For upscaling, we approximate the integral terms of $\int V_O(t)dt$ and $\int V_O(t)^2 dt$ in (21) by calculating the area of the two shaded triangles shown in Fig. 9. We assume that the integral values beyond T_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

$$\int_0^{T_X} V_O(t)dt \approx T_X V_e - \frac{1}{2} T_1 (V_e - V_s) + \frac{1}{2}(T_2 - T_1) V_{\text{ov}} \quad (30a)$$

$$\int_0^{T_X} V_O(t)^2 dt \approx T_X V_e{}^2 - \frac{1}{2} T_1 (V_e{}^2 - V_s{}^2) + \frac{1}{2}(T_2 - T_1)\Big((V_e + V_{\text{ov}})^2 - V_e{}^2\Big). \quad (30b)$$

We calculate T_1 from (26c). We calculate V_{ov} and T_2 like T_X in (25). 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:

$$V_{\text{ov}} = \gamma \cdot \text{slope}_{\text{up}}, \quad (31a)$$

$$T_2 = T_1 + \delta \cdot \text{slope}_{\text{up}}. \quad (31b)$$

The values γ and δ 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 $V_O(t)$ into (21) and 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 $E_{\text{uc,down}}$ 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

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

$V_{\mathrm{cpu}}(V)$	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

TABLE IV

DC–DC CONVERTER PARAMETERS OF LTC3733 THREE-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		

We describe the microprocessor power consumption model in (19). We obtain the parameters C_e, α_1, and α_2 from real 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 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:

$$P_{\mathrm{cpu}} = 8.4503 \upsilon_{\mathrm{proc}} V_{\mathrm{cpu}}{}^2 f_{\mathrm{cpu}} + (36.3851 V_{\mathrm{cpu}} - 33.9503) \quad (32)$$

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_{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 macromodel well 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.

TABLE V

DVFS TRANSITION DELAY OVERHEAD FOR INTEL CORE2 DUO E6850 PROCESSOR WITH LTC3733 CONVERTER

Level	Actual value (μs)			Proposed model (μs)		
	T_{uc}	Total	Cycles	T_{uc}	Total	Cycles
2→1	4.77	9.77	30 018	4.11	9.11	28 011
3→1	12.29	17.29	53 141	12.21	17.21	52 890
3→2	5.95	10.95	33 672	5.72	10.72	32 950
4→1	22.29	27.29	83 894	24.24	29.24	89 894
4→2	14.69	19.69	60 531	16.21	21.21	65 201
4→3	7.33	12.33	37 921	8.06	13.06	40 150
5→1	34.81	39.81	122 389	40.37	45.37	139 457
5→2	26.47	31.47	96 733	31.44	36.44	112 025
5→3	27.33	32.33	99 383	21.87	26.87	82 606
5→4	9.43	14.43	44 361	11.49	16.49	50 694
6→1	57.68	62.68	192 684	60.90	65.90	202 590
6→2	49.50	54.50	167 525	51.65	56.65	174 157
6→3	31.89	36.89	113 409	41.52	46.52	142 994
6→4	28.35	33.35	102 531	30.14	35.14	108 006
6→5	12.14	17.14	52 688	16.84	21.84	67 141
Downscale	0	5	15 370	0	5	15 370

T_{PLL}=5 μs and the number of cycles are calculated at f = 3.074 GHz.

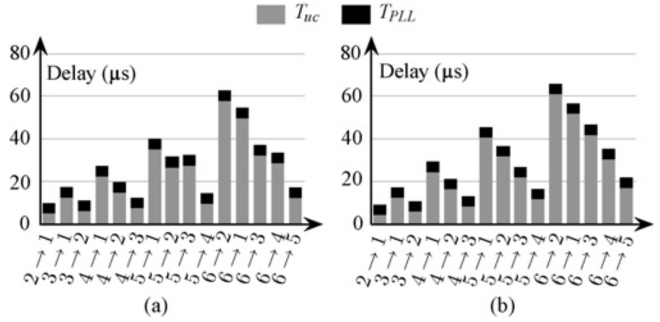


Fig. 11. Delay overhead of a DVFS transition for Intel Core2 Duo E6850. (a) Upscaling (actual). (b) Upscaling (model).

The energy overhead values of a DVFS transition for continuous- and discontinuous-mode operations are given in Tables VI and VII, respectively. We obtain $I_L(t)$, $I_O(t)$, and $V_O(t)$ from SPICE simulation and substitute them into (15), (17), and (21) to calculate the actual value. There is no E_{cap} in the tables as it is implied in E_{ir}. The value of E_{ir} for the case 1 → 6 in Table VI is large because it drains significant amount of charge from the 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. 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

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 THE 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.					

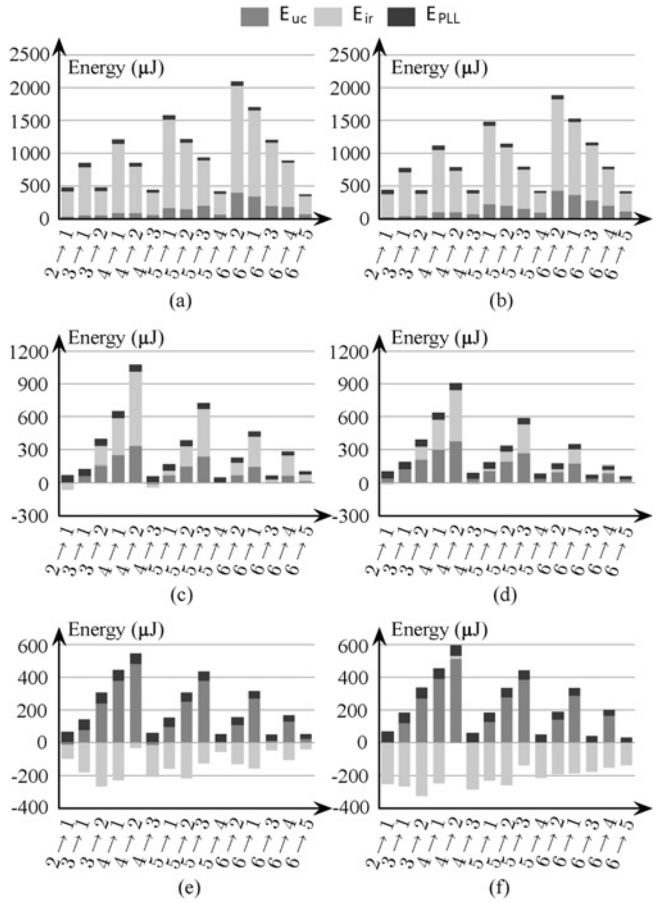


Fig. 12. Energy overhead of a DVFS transition for Intel Core2 Duo E6850. (a) Upscaling (actual). (b) Upscaling (model). (c) Downscaling continuous mode (actual). (d) Downscaling continuous mode (model). (e) Downscaling discontinuous mode (actual). (f) Downscaling discontinuous mode (model).

TABLE VIII

VOLTAGE [V_{CPU} (V)] AND CLOCK FREQUENCY [f_{CPU} (GHZ)] LEVELS FOR SAMSUNG EXYNOS 4210

DVFS level	V_{cpu}	f_{cpu}	DVFS level	V_{cpu}	f_{cpu}
Level 1	1.2	1.4	Level 4	1.05	1.12870
Level 2	1.15	1.3122	Level 5	1.00	1.0327
Level 3	1.10	1.2218			

TABLE IX

DC–DC CONVERTER PARAMETERS OF THE LTC3568 CONVERTER FOR SAMSUNG EXYNOS 4210

Parameter	Value	Parameter	Value
V_{IN}	5-V	V_{OUT}	V_O in Table VIII
C	100 μF	L	6.8 μH
R	0.11 Ω	f_{DC}	4 MHz
max(I_L)	1.8 A		

The energy overhead tends to be larger when the voltage difference is larger. For downscaling in discontinuous mode, the overhead is smaller than the value in continuous mode as 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
DVFS TRANSITION DELAY OVERHEAD OF SAMSUNG EXYNOS 4210 WITH LTC3568 CONVERTER

Level	Actual (μs)			Model (μs)		
	T_{uc}	Total	Cycles	T_{uc}	Total	Cycles
2→1	1.27	11.27	8051	0.56	10.56	7542
3→1	2.95	12.95	9254	2.11	12.11	8652
3→2	1.25	11.25	8039	0.39	10.39	7425
4→1	5.30	15.30	10 928	4.24	14.24	10 175
4→2	2.83	12.83	9165	1.76	11.76	8402
4→3	1.24	11.24	8027	0.25	10.25	7323
5→1	8.24	18.24	13 029	6.77	16.77	11 977
5→2	5.17	15.27	10 836	3.76	13.76	9826
5→3	2.72	12.72	9088	1.47	11.47	8191
5→4	1.21	11.21	8011	0.11	10.11	7223

$T_{\text{PLL}} = 10\,\mu$s and the number of cycles is calculated at $f = 1.4.$ GHz.

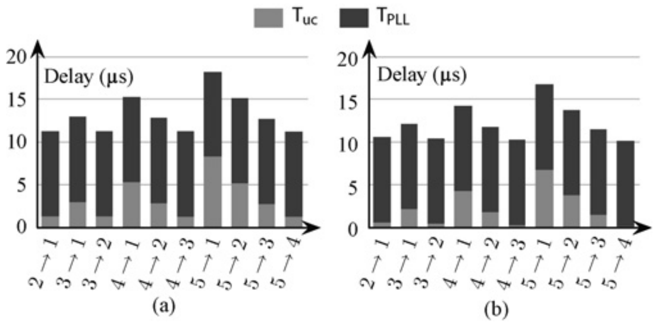


Fig. 13. Delay overhead of a DVFS transition for Samsung Exynos 4210. (a) Upscaling (actual). (b) Upscaling (model).

Exynos 4210 processor is a state-of-the-art application processor utilized in high-end mobile products such as Samsung Galaxy Note, Galaxy S II, Meizu MX, etc. It exhibits power consumption of 1.2 W at 100% processor utilization. We perform a power characterization procedure similar to that for Intel Core2 Duo E6850 processor. The resulting equation is as follows:

$$P_{\text{cpu}} = 0.446\upsilon_{\text{proc}} {V_{\text{cpu}}}^2 f_{\text{cpu}} + 0.1793V_{\text{cpu}} - 0.1527 \qquad (33)$$

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.

Table X shows the DVFS transition delay overhead for the target system. The value of T_{PLL}, 10 μs, is obtained from device datasheet. 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, LTC3568 operates in discontinuous-mode only. There is energy gain (minus overhead) in E_{ir} for downscaling because of discontinuous-mode operation.

Figs. 13 and 14 graphically show the overhead. Upscaling transitions have the large E_{ir} while downscaling transitions have large E_{uc}. This is because the LTC3632 converter charges the bulk capacitor with surge current during an upscaling transition that causes large IR loss, and it discharges the bulk capacitor by the processor current which leads to long transition time and thus large underclocking loss.

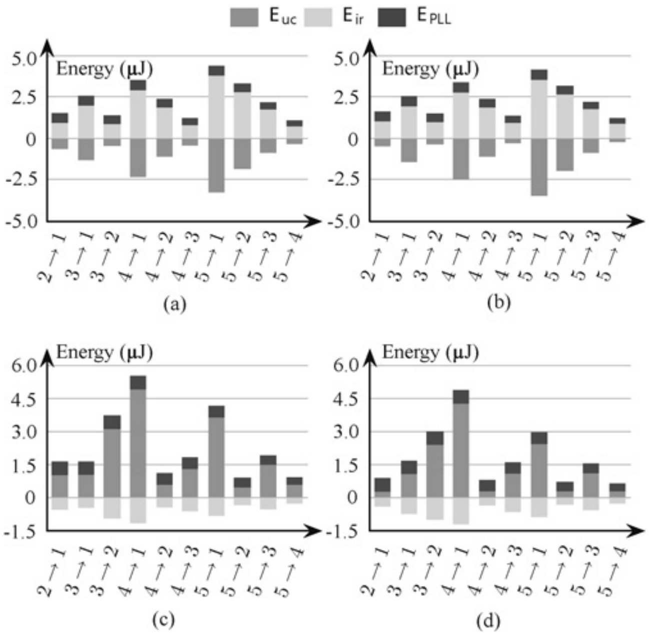


Fig. 14. Energy overhead of a DVFS transition for Samsung Exynos 4210. (a) Upscaling (actual). (b) Upscaling (model). (c) Downscaling (actual). (d) Downscaling (model).

C. Case 3: TI MSP430 Microcontroller

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:

$$P_{\text{cpu}} = 0.1128\upsilon_{\text{proc}} {V_{\text{cpu}}}^2 f_{\text{cpu}} + (0.1738V_{\text{cpu}} - 0.2832) \qquad (34)$$

where the units of P_{cpu}, V_{cpu}, and f_{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 PFM 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_{PLL} because TI MSP430 uses digitally controlled oscillator (DCO) instead of PLL, which is an improved variation of VCO. The underclocking-related overhead T_{uc} 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_{ir} while downscaling transitions have large E_{uc}. The trend in MSP430 is more distinctive than in Exynos 4210. The bulk capacitor is discharged slowly with light load current during a downscaling 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 LTC3568 Converter

Level	Actual Value			Proposed model		
	E_{uc} (μJ)	E_{ir} (μJ)	Total (μJ)	E_{uc} (μJ)	E_{ir} (μJ)	Total (μJ)
1→2	1.00	−0.545	1.085	−0.26	−0.41	0.48
1→3	1.02	−0.47	1.175	1.05	−0.74	0.93
1→4	3.09	−0.95	2.774	2.37	−1.01	1.99
1→5	4.89	−1.16	4.35	4.25	−1.20	3.67
2→3	0.57	−0.44	0.66	0.27	−0.36	0.44
2→4	1.28	−0.61	1.21	1.07	−0.65	0.95
2→5	3.63	−0.83	3.34	2.42	−0.88	2.08
3→4	0.45	−0.35	0.55	0.27	−0.32	0.40
3→5	1.47	−0.52	1.39	1.09	−0.57	0.97
4→5	0.57	−0.26	0.66	0.28	−0.28	0.36
2→1	−0.63	0.92	0.91	−0.48	1.01	1.16
3→1	−1.32	1.96	1.27	−1.42	1.93	1.13
3→2	−0.45	0.85	0.94	−0.36	0.97	1.14
4→1	−2.33	2.90	1.19	−2.48	2.77	0.91
4→2	−1.10	1.85	1.29	−1.10	1.85	1.28
4→3	−0.42	0.78	0.81	−0.28	0.92	1.09
5→1	−3.26	3.78	1.14	−3.47	3.52	0.68
5→2	−1.83	2.78	1.48	−1.95	2.64	1.23
5→3	−0.86	1.75	1.33	−0.87	1.76	1.34
5→4	−0.34	0.72	0.74	−0.23	0.88	1.01

End level (upscale) Start level (downscale)	1	2	3	4
E_{PLL} (nJ)	624.6	535.0	445.3	355.7

TABLE XII

Voltage [V_{CPU} (V)] and Clock Frequency [f_{CPU} (MHz)] Levels for the 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			

TABLE XIII

DC−DC Converter Parameters of the LTC3632 Converter for the TI MSP430 Microcontroller

Parameter	Value	Parameter	Value
V_{IN}	5-V	V_{OUT}	V_O in Table XII
C	5 μF	L	220 μH
R_L	1 mΩ	f_{DC}	Variable (PFM)
max(I_L)	20 mA		

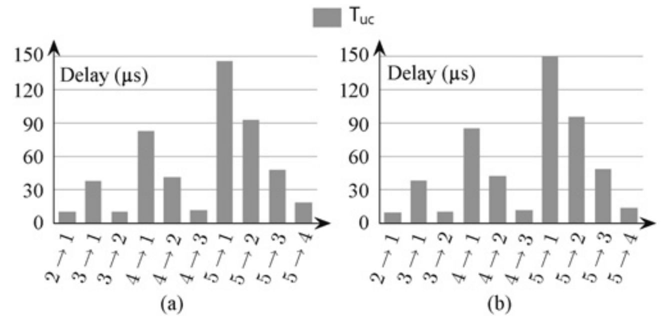


Fig. 15. Delay overhead of a DVFS transition for TI MSP430. (a) Upscaling (actual). (b) Upscaling (model).

TABLE XIV

DVFS Delay Overhead of DC−DC Converters for TI MSP430 Microcontroller With LTC3632

Level	Actual value		Proposed model	
	T_{uc} (μs)	Cycles	T_{uc} (μs)	Cycles
2→1	10.0	81	9.2	74
3→1	37.7	302	37.9	304
3→2	10.0	81	10.1	81
4→1	82.4	660	85.1	681
4→2	41.2	330	42.3	339
4→3	11.5	93	11.5	92
5→1	145.6	1165	150.0	1200
5→2	92.4	740	95.3	763
5→3	47.6	381	48.3	386
5→4	18.2	146	13.5	108

The number of cycles is calculated at $f = 8$ MHz.

TABLE XV

DVFS Transition Energy Overhead of the TI MSP430 Microcontroller With LTC3632

Level	Actual Value			Proposed model		
	E_{uc} (nJ)	E_{ir} (nJ)	Total (nJ)	E_{uc} (nJ)	E_{ir} (nJ)	Total (nJ)
1→2	1394.7	−23.8	1370.9	690.6	−23.8	666.8
1→3	3720.9	−40.0	3680.9	2710.4	−38.6	2671.8
1→4	6515.3	−45.4	6469.9	5978.5	−45.4	5933.1
1→5	10 101.3	−45.3	10056.0	10 428.1	−45.5	10 382.6
2→3	1001.5	−19.3	982.2	693.7	−18.0	675.7
2→4	3059.3	−28.9	3030.4	2725.7	−28.1	2697.6
2→5	6401.3	−31.7	6369.6	6036.4	−31.5	6004.9
3→4	1218.3	−13.8	1204.5	702.0	−13.0	689.0
3→5	3212.0	−20.2	3191.8	2776.6	−19.2	2757.4
4→5	967.7	−9.8	957.9	723.9	−13.0	710.9
2→1	−85.5	320.5	235.0	−76.5	343.2	266.7
3→1	−233.6	628.9	395.3	−239.4	670.7	431.3
3→2	−52.5	275.6	223.0	−56.5	335.4	278.9
4→1	−422.9	947.1	524.2	−410.2	986.0	575.8
4→2	−173.1	614.3	441.2	−171.2	657.3	486.1
4→3	−39.8	271.2	231.4	−40.2	328.7	288.4
5→1	−561.7	1272.4	710.7	−532.4	1292.0	759.6
5→2	−289.3	934.6	645.4	−280.4	969.0	688.7
5→3	−117.1	604.9	487.8	−116.6	646.0	529.4
5→4	−24.9	353.1	328.1	−27.3	323.0	295.7

control knob for DTM [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 K/W 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-

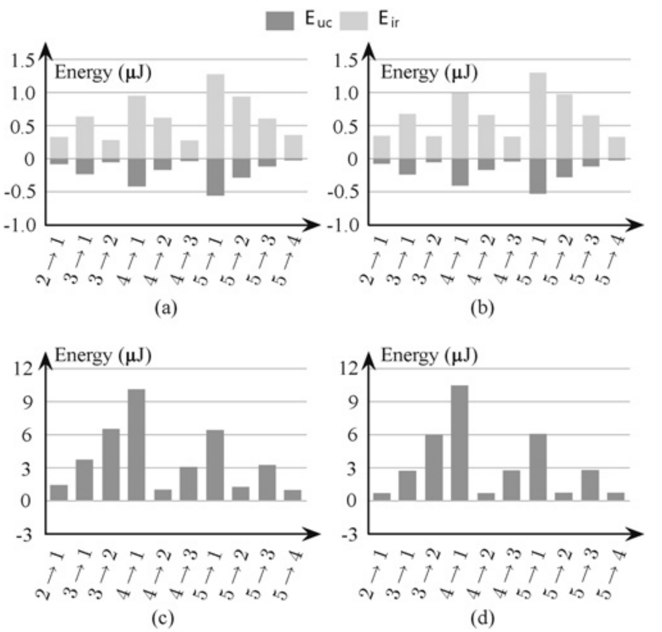


Fig. 16. Energy overhead of a DVFS transition for TI MSP430. (a) Upscaling (actual). (b) Upscaling (model). (c) Downscaling (actual). (d) Downscaling (model).

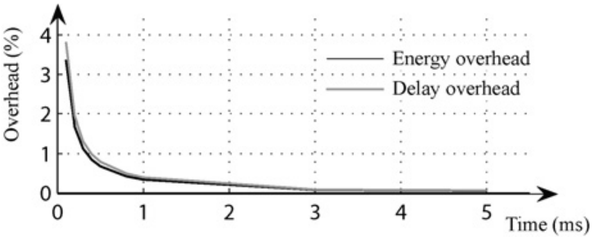


Fig. 17. Energy and delay overhead of PID control based DTM for Intel Core2 Duo E6850 processor according to time granularity of DTM.

ing to the time granularity of DTM for the 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 [32].

VIII. Conclusion

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

[1] 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 *Proc. ISLPED*, Aug. 2010, pp. 419–424.

[2] Y.-H. Lu and G. De Micheli, “Comparing system level power management policies,” *IEEE Design Test Comput.*, 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 *Proc. ISLPED*, Aug. 1998, pp. 197–202.

[4] W. Kim, J. Kim, and S. L. Min, “Preemption-aware dynamic voltage scaling in hard real-time systems,” in *Proc. ISLPED*, 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 *Proc. DAC*, 2008, pp. 179–184.

[6] T. D. Burd and R. W. Brodersen, “Design issues for dynamic voltage scaling,” in *Proc. ISLPED*, 2000, pp. 9–14.

[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 *Proc. ICCAD*, Nov. 2002, pp. 721–725.

[8] X. Zhang, A. Bermak, and F. Boussaid, “Dynamic voltage and frequency scaling for low-power multi-precision reconfigurable multiplier,” in *Proc. ISCAS*, May–Jun. 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 *Proc. DATE*, 2002, pp. 514–521.

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

[11] Y. Choi, N. Chang, and T. Kim, “Dc-dc converter-aware power management for low-power embedded systems,” *IEEE Trans. Comput. Aaided Design*, vol. 26, no. 8, pp. 1367–1381, Aug. 2007.

[12] R. Ridley, “A new, continuous-time model for current-mode control [power convertors],” *IEEE Trans. Power Electron.*, vol. 6, no. 2, pp. 271–280, Apr. 1991.

[13] Y.-F. Liu and P. Sen, “A general unified large signal model for current programmed dc-to-dc converters,” *IEEE Trans. Power Electron.*, vol. 9, no. 4, pp. 414–424, Jul. 1994.

[14] E. Van Dijk, J. Spruijt, D. O’Sullivan, and J. Klaassens, “PWM-switch modeling of dc–dc converters,” *IEEE Trans. Power Electron.*, vol. 10, no. 6, pp. 659–665, Nov. 1995.

[15] Linear Technology. (2012). *LTSPICE* [Online]. Available: www.linear.com

[16] T. Sakurai and A. Newton, “Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas,” *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, Apr. 1990.

[17] S. Gochman, A. Mendelson, A. Naveh, and E. Rotem, “Introduction to Intel core duo processor architecture,” *Intel Technology J.*, vol. 10, 2006, pp. 89–97.

[18] S. Lee and T. Sakurai, “Run-time voltage hopping for low-power real-time systems,” in *Proc. DAC*, 2000, pp. 806–809.

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

[20] A. Bashir, J. Li, K. Ivatury, N. Khan, N. Gala, N. Familia, and Z. Mohammed, “Fast lock scheme for phase-locked loops,” in *Proc. IEEE Custom Integr. Cicuit Conf.*, Sep. 2009, pp. 319–322.

[21] J. Pouwelse, K. Langendoen, and H. Sips, “Dynamic voltage scaling on a low-power microprocessor,” in *Proc. MobiCom*, 2001, pp. 251–259.

[22] 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 *Proc. ISSCC*, vol. 1. Feb. 2004, pp. 356–357.

[23] B. Mochocki, X. S. Hu, and G. Quan, “A realistic variable voltage scheduling model for real-time applications,” in *Proc. ICCAD*, Nov. 2002, pp. 726–731.

[24] P. Schaumont, B.-C. C. Lai, W. Qin, and I. Verbauwhede, “Cooperative multithreading on embedded multiprocessor architectures enables energy-scalable design,” in *Proc. DAC*, Jun. 2005, pp. 27–30.

[25] D. Shin and J. Kim, "Optimizing intratask voltage scheduling using profile and data-flow information," *IEEE Trans. Comput. Aided Design*, vol. 26, no. 2, pp. 369–385, Feb. 2007.

[26] P. Pillai and K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," in *Proc. SOSP*, 2001, pp. 89–102.

[27] R. Redl and J. Sun, "Ripple-based control of switching regulators—An overview," *IEEE Trans. Power Electron.*, vol. 24, no. 12, pp. 2669–2680, Dec. 2009.

[28] V. Yousefzadeh, A. Babazadeh, B. Ramachandran, E. Alarcon, L. Pao, and D. Maksimovic, "Proximate time-optimal digital control for synchronous buck dc–dc converters," *IEEE Trans. Power Electron.*, vol. 23, no. 4, pp. 2018–2026, Jul. 2008.

[29] A. Shye, B. Scholbrock, and G. Memik, "Into the wild: Studying real user activity patterns to guide power optimizations for mobile architectures," in *Proc. MICRO*, 2009, pp. 168–178.

[30] S. Zhang and K. Chatha, "Approximation algorithm for the temperature-aware scheduling problem," in *Proc. ICCAD*, Nov. 2007, pp. 281–288.

[31] 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.

[32] D. Tsafrir, "The context-switch overhead inflicted by hardware interrupts (and the enigma of do-nothing loops)," in *Proc. ExpCS*, 2007, p. 3.

Sangyoung Park (S'08) received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 2008, where he is currently pursuing the Ph.D. degree in electrical engineering and computer science.

He was a Visiting Scholar with the University of Southern California, Los Angeles, in 2010. His current research interests include embedded system design, system-level power management, next-generation power sources, and hybrid electrical energy storage systems.

Mr. Park was a recipient of the IEEE SSCS Seoul Chapter Best Paper Award in ISOCC 2009 and was the International Symposium on Low Power Electronics and Design Low Power Design Contest Winner in 2012.

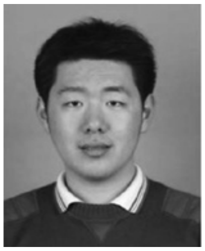
Jaehyun Park (S'08) received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 2006, where he is currently pursuing the Ph.D. degree in electrical engineering and computer science.

He was a Visiting Scholar with the University of Southern California, Los Angeles, from 2009 to 2010. His current research interests include low-power-embedded system design and nonvolatile memory system design.

Mr. Park was a recipient of the International Symposium on Low Power Electronics and Design Low Power Design Contest Winner in 2007 and 2012.

Donghwa Shin (M'12) received the B.S. degree in computer engineering, the M.S. degree in computer science, and the Ph.D. degree in electrical engineering all from Seoul National University, Seoul, Korea, in 2005, 2007, and 2012, respectively.

He is currently joining Politecnico di Torino, Turin, Italy, as a Research Assistant. His current research interests include system-level low-power techniques for embedded system, low-power techniques for display, and portable electronic system design using hybrid power sources.



Yanzhi Wang (S'11) received the B.S. degree (with distinction) in electronic engineering from Tsinghua University, Beijing, China, in 2009. He is currently pursuing the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, under the supervision of Prof. M. Pedram.

His current research interests include system-level power management, next-generation energy sources, hybrid electrical energy storage systems, and the smart grid. He has published more than 30 papers in these areas.

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 ON VLSI SYSTEMS. He is a recipient of the NSF's Young Investigator 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.

Naehyuck 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 Embedded 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, and ACM Transactions on Embedded Computing 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.