

Accurate Modeling and Calculation of Delay and Energy Overheads of Dynamic Voltage Scaling in Modern High-Performance Microprocessors^{*}

Jaehyun Park, Donghwa Shin, and
Naehyuck Chang[†]
Seoul National University, Korea
{jhpark, dhshin,
naehyuck}@elpl.snu.ac.kr

Massoud Pedram
University of Southern California, CA, USA
pedram@usc.edu

ABSTRACT

Dynamic voltage and frequency scaling (DVS) has been studied for well over a decade, and even commercial systems widely support DVS nowadays. Nevertheless, existing DVS transition overhead models do not accurately reflect modern DVS architectures including modern DC–DC converters, PLL (Phase Lock Loop), and voltage and frequency change policies. Incorrect DVS overhead models prevent one from achieving the maximum energy gain, by misleading the DVS control policies. This paper introduces an accurate DVS overhead model, in terms of both energy consumption and time penalty, through detailed observation of modern DVS setups and voltage and frequency change guidelines from vendors. We introduce new major contributors to the DVS overhead including the performance underdrive loss of the DVS-enabled microprocessor, additional inductor IR loss, and so on, as well as consideration of power efficiency from discontinuous-mode DC–DC conversion. Our DVS overhead model enhances the DVS overhead model accuracy from 86% to 238% for Intel Core2 Duo E6850 and LTC3733.

Categories and Subject Descriptors

B.8.2 [Performance Analysis and Design Aids]; C.4 [Performance of Systems]: Modeling techniques

General Terms

Design, Experimentation

Keywords

DVFS, DVS overhead model, PLL, DC–DC converter

^{*}This work is supported by the Brain Korea 21 Project, IC Design Education Center (IDEC), and Mid-career Researcher Program through NRF grant funded by the MEST (No. 2010-0017680). The ICT at Seoul National University provides research facilities for this study.

[†]Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISLPED'10, August 18–20, 2010, Austin, Texas, USA.

Copyright 2010 ACM 978-1-4503-0146-6/10/08 ...\$10.00.

1. INTRODUCTION

Dynamic voltage and frequency scaling (DVS) has proved itself as one of the most successful energy saving techniques. Although DVS primarily aims at dynamic power savings, it also has positive impacts on leakage power saving. As of today, even high-performance microprocessors support DVS e.g., Intel's SpeedStep Technology [1] (or the AMD equivalent PowerNow!). DVS requires an (output voltage) programmable DC–DC converter and a programmable clock generator. It is natural to incur some overhead whenever the system changes its voltage and frequency setting. The break-even time of DVS (minimum duration of time that the target system should stay in a particular voltage-frequency state for the DVS to produce a positive energy gain) is strongly dependent on this overhead. Consequently, it is obvious that correct overhead estimation is crucial to realizing the maximum DVS benefit.

The DVS overhead appears as two major factors: time and energy. Most modern microprocessors have a phase lock loop (PLL) as a frequency synthesizer for the clock signal. As the output clock is subject to jitter and/or glitches during the PLL lock time, the microprocessor should stop operation, which generally determines the time overhead of DVS. Recent DVS setup allows the microprocessor to continue its operation during the voltage transition period, but then the microprocessor may not be able to operate at the maximum feasible speed during the voltage transition period, which is another source of the time overhead.

Energy overhead of DVS is caused by four major factors. First, the DC–DC converter may consume additional energy during the voltage transition period. Second, the microprocessor can be supplied a higher voltage than the necessary minimum voltage during the voltage transition period. This results in additional dynamic and static energy dissipation. Third, even during the halt period, i.e., during the PLL lock time, the microprocessor may consume static power. Fourth, upscaling in a conventional DC–DC converter requires more current to be fed through the inductor to increase the bulk capacitor voltage, which results in additional IR loss due to the inductor.

Correct modeling of the voltage transition overhead is not a trivial undertaking. It requires analysis of the DC–DC converter, frequency synthesizer, voltage and frequency transition policies, and so forth. Unfortunately, majority of DVS studies simply ignore the transition overhead [2, 3, 4]. We surveyed 120 DVS-related papers published in the past 10 years, and found that only 17% of DVS papers, considered the transition overhead. Among the 17% of the papers, 75% of papers are based on the analytical transition overhead model introduced in [5, 6]. Other variations are summarized in Section 3.

The transition overhead can be negligible or significant depending on how often we change the DVS setting. However, generally

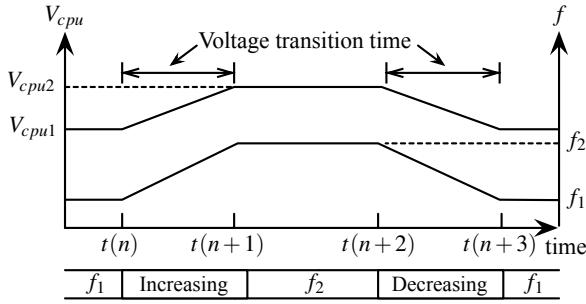


Figure 1: DVS up and downscaling for a VCO clock.

aggressive DVS policies may have more chance to realize the DVS potential. The transition overhead is a major deterrent to wider and more successful use of the DVS. Thus, it is crucial to accurately estimate the DVS transition overhead. Moreover, existing DVS overhead models have limitations and cannot be utilized in modern DVS setups. They are significantly simplified, contain technical fallacies, or are limited to uncommon setups as will be summarized in Section 3.

This paper takes into account all the major lossy components for the modern DVS setups introducing performance underdrive loss that occurs during the voltage transition period. We formulate a user friendly analytical model with easy parameters that can be acquired from the data sheets and/or passive component values such as R, L and C.

2. DVS TRANSITION SEQUENCES

DVS setups require a voltage regulator and a clock generator with programmable output voltage and frequency, respectively, in addition to a DVS-enabled microprocessor. Output programming is typically done by configuration register access from the microprocessor. The configuration registers are often mapped to special registers of the microprocessor such as MSRs (Model Specific Registers) of Intel Core Duo [1].

A high-efficiency voltage regulator is a switching-mode DC–DC converter because the additional dropout of a linear regulator by reducing the output voltage simply results in power loss. The inductor current increases when the upper MOSFET is turned on, and in turn the bulk capacitor is charged. The inductor current continuously decreases when the lower MOSFET is turned off, but the inductor still keeps supplying current to the bulk capacitor, dissipating the stored electromagnetic energy. Most importantly, the inductor current never changes abruptly, which results in adiabatic charging and discharging to and from the bulk capacitor. Therefore, the primary losses of 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.

For upscaling (raising) the supply voltage, the microprocessor sets a new VID (voltage identifier) code to have a higher output voltage than the current one. The voltage comparator recognizes the DC–DC converter (bulk capacitor) output voltage is lower than the VID and increases the duty ratio of the upper MOSFET. This increases the inductor current, and the charge current of the bulk capacitor becomes larger than the discharge current (current consumed by the microprocessor). This eventually increases the bulk capacitor voltage. There is no difference in the voltage transition sequence between a continuous-mode and discontinuous-mode DC–DC conversions during voltage upscaling.

For downscaling (lowering) the supply voltage, discontinuous-mode DC–DC conversion makes the DC–DC converter output voltage decrease by discharging the bulk capacitor with the microprocessor

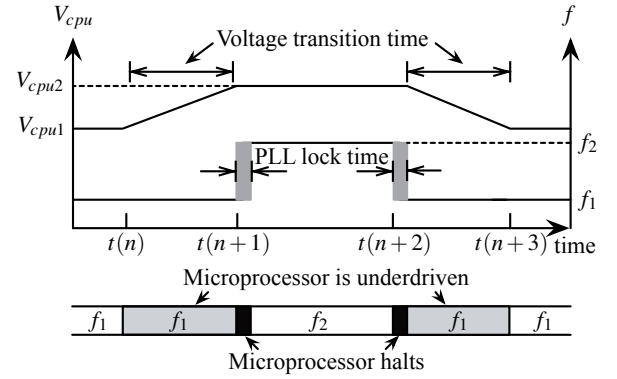


Figure 2: DVS up and downscaling for a PLL clock.

power supply current only. Thus, there is no wasted energy from the bulk capacitor during downscaling although the voltage transition time is generally longer. In contrast, continuous-mode DC–DC conversion discharges the bulk capacitor to GND as well as the microprocessor power supply current. The discharge to GND results in significant energy loss while the voltage transition time is shorter. Modern DVS setups prefer to use discontinuous-mode DC–DC conversion for downscaling for more efficient use of the stored energy in the bulk capacitor.

2.1 Clock Frequency Transition

The relation between the DVS supply voltage and clock frequency can be approximately defined by the Alpha Power Law [7]. Early DVS work assumes a VCO (voltage controlled oscillator) for the clock generator [5]. The VCO performs automatic and continuous frequency change according to the transient voltage. As Fig. 1 illustrates, the gradual frequency changes allow the microprocessor keep operating during the entire voltage transition period. However, VCOs are not commonly used in typical microprocessor-based systems due to the unstable and imprecise clock frequency.

On the other hand, PLLs are widely used for the programmable clock generators thanks to the accuracy of the frequency setting. PLL setting is independent from the DC–DC converter setting. Operating at a lower frequency than maximum allowable clock frequency does not incur stability issue, but wastes valuable energy and degrades the DVS gain. Fig. 2 shows that the microprocessor should halt during the PLL lock time. PLL lock time takes typically tens of microseconds for a modern digital PLL [8]. Modern processors such as the Nehalem architecture of Intel typically have several micro-seconds lock time [9, 1]. A StrongARM 1100 processor measurement result shows that the PLL lock time is insensitive to the difference between the current and target frequencies [10].

As illustrated in Fig. 2, 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 operation during the PLL lock time. Therefore, the microprocessor is supplied an unnecessarily high voltage during the voltage transition period, i.e., the microprocessor is underdriven in terms of its clock frequency. The microprocessor consumes unnecessarily high dynamic and static power due to performance underdriving. We pay attention to this situation as one of the most dominant sources of the voltage transition energy overhead, and quantify it through detailed analysis in Section 4. 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 [11]. IBM's PowerTune technology is able to make a fre-

quency transition in one cycle using multiple pre-generated clocks and selecting one by a multiplexor [12]. Although our proposed model focuses on conventional digital PLLs, it is applicable to this technology as well except for the PLL lock time in Section 4.3.1. It is obvious that the PLL lock time of this technology is one clock cycle.

3. DVS OVERHEAD MODELS: OVERVIEW

3.1 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 (i.e., frequency scaling is the time limiting part of the transition), an assumption which is justified for old-fashioned analog PLL clock generators, and that the PLL lock time is constant. These models assume the microprocessor is halted during the entire transition period [8, 13, 14]. Later work uses constant transition energy overhead on top of the constant transition time model [15]. Another type of model considers 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 scaling is the time limiting part of the transition). This work, however, goes on to assume a constant voltage transition time [16]. Transition energy overhead is ignored insisting on that the microprocessor is halted during the transition period.

3.2 Variable Transition Overhead Models

One of the most frequently-referred DVS overhead models from [5] assumes a continuous-mode DC–DC conversion and a voltage-controlled oscillator (VCO) clock generator. Published works that refer to this model do not specify whether a VCO or a PLL is used for the clock generator, and use a 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 .

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

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

where I_{max} is the maximum output current of the DC–DC converter, factor of 2 is applied because the current is pulsed in a triangular waveform, η is the efficiency of the DC–DC converter, and V_s and V_e are voltages before and after the transition, respectively. One shortcoming of this model is overestimation of I_{max} . While [5] assumes I_{max} 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.,

$$T_X = \frac{2C_b}{I_{max} - I_{cpu}}|V_e - V_s|, \quad (3)$$

where I_{cpu} is the microprocessor power supply current during the DVS transition. Because the microprocessor continues to operate even during the voltage transition, I_{cpu} has a significant effect 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 operating during the voltage transition period, T_X becomes the time overhead for the DVS 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 [5] 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 DVS setup is as follows

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

If voltage up and downscaling happens evenly, we may distribute the transition overhead as follows (this is similar to calculation of CMOS logic gate dynamic energy).

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

Another frequently referred DVS overhead model is [6], which is basically the same as that of reference [5], but has additional consideration of body bias.

4. PROPOSED DVS OVERHEAD MODEL

This section introduces a correct and highly accurate DVS overhead model for modern DC–DC converters that include both continuous- and discontinuous-modes of operation and a PLL clock generator for practical modern DVS systems. From modern literature, we derive the PLL lock time as a constant, independent of the present and next clock frequencies. On the other hand, the voltage transition overhead primarily determines the DVS transition overhead. We take all the distinct sources of this overhead into account including the microprocessor and DC–DC converter losses. Actual microprocessors consume noticeable amount of power while they halt. We additionally consider static power during the PLL lock time as the secondary source of the overhead, which has not been mentioned in prior work.

4.1 Target DVS System Setup

A high-fidelity DVS transition overhead model requires detailed microprocessor power consumption information which reflects the supply voltage and frequency changes. We choose a high-end DVS-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 [17], which is a representative setup of a modern high-performance DVS-enabled microprocessor.

The microprocessor power consumption model is based on the following equation:

$$P_{cpu} = P_{dyn} + P_{sta} = \left(C_e V_{cpu}^2 f_{cpu} \right) + \left(\alpha_1 V_{cpu} + \alpha_2 \right), \quad (6)$$

where P_{dyn} and P_{sta} denote dynamic and static power consumptions, respectively, C_e is the average switched capacitance per cycle, and V_{cpu} and f_{cpu} are the supply voltage and the clock frequency of the microprocessor.

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_{cpu} . We run PrimeZ benchmark and

Table 1: Voltage (V_{cpu} (V)) and clock frequency (f_{CPU} (GHz)).

DVS level	V_{cpu}	f_{cpu}	DVS 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 2: 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

change V_{cpu} and f_{cpu} performing direct access to the BIOS (basic input/output system) as described in Table 1 because the Intel Speed-Step supports only two voltage levels. We finally derive the following power consumption model:

$$P_{cpu} = 8.4503V_{cpu}^2f_{cpu} + (36.3851V_{cpu} - 33.9503), \quad (7)$$

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 2.

4.2 Energy Cost of a Voltage Transition

4.2.1 Time duration of a voltage transition

We perform LTSPICE [18] simulation to observe the DVS transition overhead. We configure passive components and switching frequency of the DC–DC converter as reported in Table 3. The simulation environment is designed to change the digital input of the DC–DC converter using a 5-bit VID controller. We characterize the DVS voltage transition time overhead with the variables as reported in Table 4. Notice that we characterize the up and downscaling overheads separately because, as also noted before, their actual behaviors are quite different.

Voltage upscaling pumps more charge into the bulk capacitor increasing the inductor current, $I_L(t)$. Fig. 3 illustrates an upscaling transition from Level 3 to Level 1 (cf. Table 1). We achieve settling time of about a 70 μ s. The difference between the approximately 30A normal operating $I_L(t)$ and the temporarily higher transient $I_L(t)$ level of more than 60A is the additional energy overhead for the upscaling transition. For illustration purpose, the shaded area in this figure roughly represents the upscaling overhead.

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

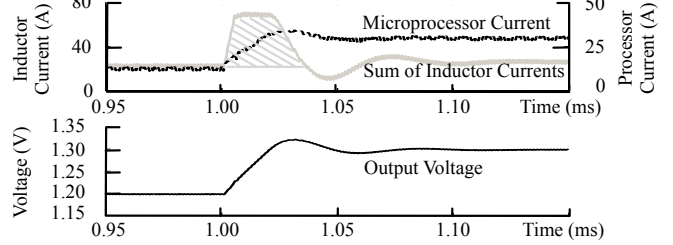
Fig. 5 shows how discontinuous-mode DC–DC conversion works. As soon as the inductor current becomes negative, the bottom tran-

Table 3: DC–DC converter setup for LTSPICE simulations.

Parameter	Value	Parameter	Value
V_{IN}	12 (V)	V_{OUT}	V_{cpu} in Table 1
η	0.9	C	8840 (μ F)
L	1 (μ H)	R_L	2.3 (m Ω)
f_{DC}	530 (kHz)	I_{max}	60 (A)

Table 4: Notations for voltage transition time model.

T_X^\dagger	The amount of time needed for voltage transition
C_b	Output capacitance of a DC–DC converter
V_s	Initial supply voltage for a transition
V_e	Final supply voltage for a transition
$V_{cpu}(t)$	Transient output voltage
$I_L(t)$	Sum of transient current of inductors
$I_{cpu}(t)$	Power supply current of a microprocessor

**Figure 3: Upscaling (Level 3 \rightarrow Level 1).**

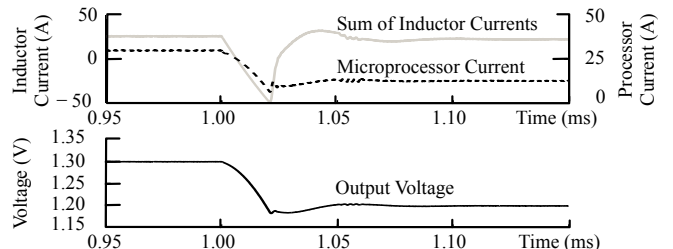
sistor is turned off, which prevents the bulk capacitor from discharging further. Instead, I_{cpu} discharges the bulk capacitor and makes the DC–DC converter output voltage converge to V_e . It goes without saying that downscaling takes longer to stabilize in the discontinuous mode compared to the continuous-mode DC–DC conversion because in the former case the only current discharging the bulk capacitor is I_{cpu} whereas in the latter the bulk capacitor is actively discharged to the GND through the bottom transistor. On the positive side, the actual energy overhead of the discontinuous-mode DC–DC conversion is generally smaller than that of the continuous-mode DC–DC conversion because the bulk capacitor discharge current is the sum of I_{cpu} and low-side MOSFET current. However, we emphasize that, due to longer transition time, discontinuous-mode DC–DC conversion gives rise to more performance *underdrive* of the microprocessor, which is an additional energy overhead as shown in Fig. 2. We describe the underdrive energy overhead in Section 4.2.2.

Discontinuous-mode plays an important role in modern DC–DC converter design by maintaining high conversion efficiency even when the load current is light. Since the microprocessor I_{cpu} fluctuates significantly during runtime, discontinuous-mode effectively prevents negative inductor current flow, i.e., avoids discharging the bulk capacitor to GND. Voltage downscaling is a case that incurs negative inductor current. Discontinuous-mode is thus crucial for efficient voltage downscaling like the regenerative brake systems for hybrid vehicles [19].

In summary, the actual DVS voltage transition time T_X^\dagger should satisfy the following equation:

$$\int_0^{T_X^\dagger} (I_L(t) - I_{cpu}(t)) dt = C_b(V_e - V_s). \quad (8)$$

T_X^\dagger is an important variable to determine the amount of E_{up} , E_{down} and E_{ud} . Note that we take $I_{cpu}(t)$ into consideration because $\max(I_L(t))$ is not large enough to allow us ignore I_{cpu} during voltage transition. However, I_{cpu} was ignored in [5] as explained Section 3.

**Figure 4: Continuous-mode downscaling (Level 1 \rightarrow Level 3).**

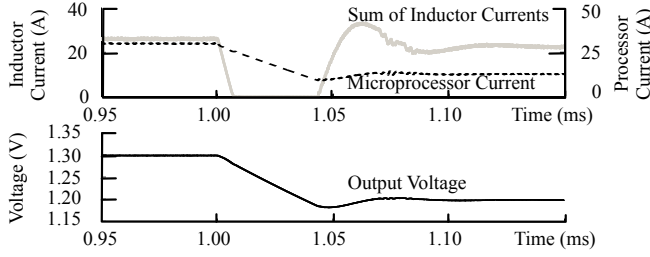


Figure 5: Discontinuous-mode downscaling (Level 1 → Level 3).

Table 5: Notation for DVS transition energy overhead model.

E_C^\dagger	Voltage transition energy cost
E_{cap}	Charge transfer to and from the bulk capacitor
E_{up}	Additional inductor IR loss
E_{down}	Bulk capacitor charge loss due to the continuous mode
E_p	Energy penalty during the DVS transition
E_{ud}	Underdrive loss
E_{PLL}	Microprocessor energy consumption during the PLL lock time
f_s	Initial clock frequency for a transition
f_e	Final clock frequency for a transition
T_{lock}^\dagger	PLL lock time

4.2.2 Energy cost of a voltage transition

Conventional DVS transition energy overhead accounts for the charge transfer to and from the bulk capacitor as described in Section 3. We perform LTSPICE simulation and confirm the presence of: i) additional inductor IR loss during voltage upscaling and ii) energy loss due to continuous-mode DC–DC conversion.

Charge transfer to and from the bulk capacitor: The DC–DC converter output voltage is set by the bulk capacitor terminal voltage, which is in turn proportional to this charge stored in the capacitor. Voltage upscaling transfers additional charge to the bulk capacitor and increases the terminal voltage from V_s to V_e . The amount of energy for the upscaling is calculated as

$$E_{cap} = \frac{1}{2} C_b (V_e^2 - V_s^2), \quad (9)$$

Notice that $E_{cap} > 0$ for voltage upscaling, which denotes energy loss.

For downscaling, $E_{cap} \leq 0$ because the upper MOSFET in the DC–DC converter is open and stops supplying current to the bulk capacitor, but the bulk capacitor still supplies power to the microprocessor until the voltage converges to V_e . This contributes as a source of negative energy overhead (i.e., energy gain) during voltage downscaling.

Additional inductor IR losses: The additional charge transfer to the bulk capacitor also incurs IR loss in the inductor. This loss is not symmetrical because voltage downscaling does not involve the inductor.

$$E_{up} = \int_0^{T_X^\dagger} R_L (I_L(t)^2 - I_{cpu}(t)^2) dt. \quad (10)$$

Notice that the above equation does not account for the inductor IR losses due to microprocessor current that goes thru the inductor. This is necessary in order to avoid double counting on the energy losses during upscaling. Instead it only counts the additional current needed to increase the bulk capacitor voltage.

Energy loss due to continuous-mode DC–DC conversion: Unfortunately, continuous-mode DC–DC conversion mostly wastes the potential energy gain from E_{cap} during the voltage downscaling because the lower MOSFET discharges the bulk capacitor to GND.

The energy loss, E_{down} is given by

$$E_{down} = \int_0^{T_X^\dagger} V_{cpu}(t) |I_L^*(t)| dt, \quad (11)$$

where

$$I_L^*(t) = \begin{cases} I_L(t) & \text{when } I_L(t) < 0, \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

On the other hand, discontinuous-mode DC–DC conversion effectively blocks the negative inductor current, i.e., $E_{down} = 0$.

Total voltage transition energy cost: The total voltage transition energy cost, which is not symmetrical for voltage up and downscaling, and is given by

$$E_C^\dagger = \begin{cases} E_{cap} + E_{up} & \text{: upscaling,} \\ E_{cap} + E_{down} & \text{: downscaling.} \end{cases} \quad (13)$$

4.3 Time and Energy Overhead Models

4.3.1 Time overhead of a DVS transition

Once again, we model the PLL lock time as a constant time penalty regardless of f_s and f_e as described in [10]. Most previous work regards the PLL lock time as the only source of the time penalty that causes the microprocessor performance degradation. However, we address another time overhead factor especially for upscaling. Therefore, we first present the penalty of the microprocessor during the DVS transition in cycles, and then derive time penalty. The microprocessor operates at f_s during the voltage upscaling time to guarantee safe operation of the microprocessor. This is similar to the performance underdrive loss but only voltage upscaling is subject to the additional time overhead (Fig. 2). Thus, the cycle penalty, C_P is given by

$$C_P = \begin{cases} f_e \cdot T_{lock}^\dagger + (f_e - f_s) T_X^\dagger & \text{: upscaling,} \\ f_e \cdot T_{lock}^\dagger & \text{: downscaling,} \end{cases} \quad (14)$$

and then we present time penalty using the cycle penalty as follows

$$T_P^\dagger = \begin{cases} T_{lock}^\dagger + \frac{f_e - f_s}{f_e} T_X^\dagger & \text{: upscaling,} \\ T_{lock}^\dagger & \text{: downscaling.} \end{cases} \quad (15)$$

The time penalty is the only overhead for a DVS transition because the the microprocessor is not halted for T_X^\dagger as described in Section 2.1. Therefore, the time overhead of a DVS transition is the same as the time penalty as follows:

$$T_O = T_P^\dagger. \quad (16)$$

4.3.2 Energy overhead of a DVS transition

There is a energy penalty when the microprocessor changes its voltage and the frequency. We perform LTSPICE simulation and evaluate the microprocessor performance underdrive energy loss. We find that I_{cpu} during the PLL lock time is an important factor for calculating the energy loss and thus, should not be ignored.

Microprocessor performance underdrive loss: One of the most dominant energy loss sources is caused by underdriving 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. 2. Because of underdrive, the microprocessor consumes additional dynamic and static power, which is given by

$$E_{ud} = C_e \min(f_s, f_e) \times \int_0^{T_X^\dagger} (V_{cpu}(t)^2 - \min(V_s, V_e)^2 + \alpha_1 (V_{cpu}(t) - \min(V_s, V_e))) dt. \quad (17)$$

Table 6: Comparison of the DVS transition overhead with a discontinuous-mode DC-DC converter.

Level	Proposed model					Burd <i>et al.</i> [5]			
	T_X^\dagger (μ s)	E_C^\dagger (mJ)	E_P^\dagger (mJ)	T_P^\dagger, T_O (μ s)	E_O (mJ)	T_X (μ s)	Err (%)	E_X (mJ)	Err (%)
1→2	16.7	-0.56	0.44	5.0	-0.13	14.7	12	0.11	189
1→3	31.2	-1.10	0.53	5.0	-0.57	29.5	5	0.22	138
1→4	48.7	-1.62	0.67	5.0	-0.95	44.2	9	0.32	134
1→5	65.4	-1.83	0.81	5.0	-1.02	58.9	10	0.42	141
1→6	82.6	-1.83	0.91	5.0	-0.92	73.7	11	0.52	156
2→1	17.9	0.62	0.46	6.3	1.08	14.7	18	0.11	90
2→3	17.6	-0.54	0.43	5.0	-0.12	14.7	16	0.11	194
2→4	35.0	-1.06	0.52	5.0	-0.54	29.5	16	0.21	139
2→5	56.1	-1.52	0.66	5.0	-0.86	44.2	21	0.31	136
2→6	76.8	-1.63	0.79	5.0	-0.85	58.9	23	0.41	148
3→1	29.0	1.20	0.55	9.6	1.75	29.5	-2	0.22	87
3→2	16.0	0.59	0.44	6.5	1.03	14.7	8	0.11	89
3→4	19.5	-0.52	0.42	5.0	-0.10	14.7	24	0.10	201
3→5	41.0	-1.02	0.52	5.0	-0.50	29.5	28	0.20	141
3→6	69.2	-1.40	0.67	5.0	-0.73	44.2	36	0.30	141
4→1	39.2	1.76	0.67	15.1	2.43	44.2	-13	0.32	87
4→2	25.7	1.15	0.52	10.1	1.67	29.5	-15	0.21	87
4→3	15.8	0.56	0.42	6.9	0.99	14.7	7	0.10	89
4→5	22.7	-0.50	0.41	5.0	-0.09	14.7	35	0.10	215
4→6	51.7	-0.97	0.53	5.0	-0.45	29.5	43	0.19	144
5→1	45.1	2.28	0.78	21.7	3.06	58.9	-31	0.42	86
5→2	34.6	1.68	0.62	16.2	2.29	44.2	-28	0.31	86
5→3	22.9	1.09	0.49	10.8	1.58	29.5	-29	0.20	87
5→4	15.4	0.53	0.41	7.4	0.94	14.7	4	0.10	89
5→6	28.5	-0.47	0.41	5.0	-0.07	14.7	48	0.10	239
6→1	50.7	2.77	0.87	30.3	3.64	73.7	-45	0.52	86
6→2	41.1	2.17	0.69	23.9	2.86	58.9	-43	0.41	86
6→3	30.5	1.59	0.56	17.3	2.15	44.2	-45	0.30	86
6→4	21.2	1.04	0.46	11.9	1.50	29.5	-39	0.19	87
6→5	15.2	0.50	0.40	8.1	0.90	14.7	3	0.10	89

Power consumption during the PLL lock time: The microprocessor operation is halted during the PLL lock time as explained in Section 2. 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 during the PLL lock time, which is given by

$$E_{PLL} = \int_0^{T_{lock}^\dagger} (\alpha_1 V_{cpu} + \alpha_2) dt. \quad (18)$$

Total DVS transition energy penalty: The total DVS transition energy penalty is given by

$$E_P^\dagger = E_{ud} + E_{PLL}. \quad (19)$$

The total energy overhead of a DVS transition is the sum of the voltage transition energy cost in (13) and energy penalty in (19) as follows:

$$E_O = E_C^\dagger + E_P^\dagger. \quad (20)$$

5. EXPERIMENTAL RESULTS

We perform LTSPICE simulation to obtain $I_L(t)$, $I_{cpu}(t)$, and $V_{cpu}(t)$ during a DVS transition. We calculate the actual overhead values of a DVS transition by using the proposed model for an Intel Core2 Duo E6850 processor and an LTC3733 DC-DC converter based on the simulation results. Table 6 summarizes detailed transition time and energy overheads for all the possible DVS transition cases, and compares the values with the model in [5]. We compare T_X and E_X in [5] with T_X^\dagger and E_O , respectively. Note that the energy overhead model in [5] is mostly based on the charge transfer to and from the bulk capacitor. In practice, the actual E_{cap} does not dominate the transition overhead energy: Furthermore there is energy gain during voltage

downscaling due to the discontinuous mode. As shown in Table 6, the model in [5] has relative error from 86% to 238%. We also perform the same experiment for continuous-mode DC-DC conversion and the range of the relative errors of the model in [5] is -689% to 86%, but do not include them due to the page limitation. Relative errors of the continuous mode is bigger than that of the discontinuous mode because the magnitudes of the overhead values are small.

6. CONCLUSIONS

This paper introduces a correct DVS transition overhead model that concerns the structure and the operation of a DC-DC converter, and makes up for the weak points in existing overhead models. A correct DVS overhead model is crucial to determine a proper break-even time to maximize the energy gain. Our model accommodates modern high-performance DVS systems such as discontinuous-mode DC-DC conversion and a PLL clock generator. We introduce performance underdrive loss, inductor IR loss as well as charge transfer to and from the bulk capacitor, and PLL lock time. We provide experimental results to assess the accuracy of our analytical model with LTSPICE simulation. Our model enhances the DVS overhead model accuracy from 86% to 238% for Intel Core2 Duo E6850 and LTC3733.

7. REFERENCES

- [1] Intel Core2 Extreme Processor QX9000 and Intel Core2 Quad Processor Q9000, Q9000S, Q8000 and Q8000S Series Datasheet. 2009.
- [2] T. Ishihara *et al.*, "Voltage scheduling problem for dynamically variable voltage processors," in *Proc. ISLPED*, pp. 197-202, 1998.
- [3] W. Kim *et al.*, "Preemption-aware dynamic voltage scaling in hard real-time systems," in *Proc. ISLPED*, pp. 09-11, 2004.
- [4] Z. Cao *et al.*, "Optimality and improvement of dynamic voltage scaling algorithms for multimedia applications," in *Proc. DAC*, pp. 179-184, 2008.
- [5] T. D. Burd *et al.*, "Design issues for dynamic voltage scaling," in *Proc. ISLPED*, pp. 9-14, 2000.
- [6] S. M. Martin *et al.*, "Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads," in *Proc. ICCAD*, pp. 721-725, 2002.
- [7] T. Sakurai *et al.*, "Alpha-power law mosfet model and its applications to cmos inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, pp. 584-594, 1990.
- [8] S. Lee *et al.*, "Run-time voltage hopping for low-power real-time systems," in *Proc. DAC*, pp. 806-809, 2000.
- [9] A. Bashir *et al.*, "Fast Lock Scheme for Phase-Locked Loops," in *Proc. CICC*, pp. 319-322, 2009.
- [10] J. Pouwelse *et al.*, "Dynamic voltage scaling on a low-power microprocessor," in *Proc. MobiCom*, pp. 251-259, 2001.
- [11] S. Gochman *et al.*, "Introduction to intel core duo processor architecture," in *Intel Technology Journal*, vol. 10, pp. 89-97, 2006.
- [12] C. Lichtenau *et al.*, "PowerTune: advanced frequency and power scaling on 64b PowerPC microprocessor," in *Proc. ISSCC*, pp. 356-357, 2004.
- [13] B. Mochocki *et al.*, "A realistic variable voltage scheduling model for real-time applications," in *Proc. ICCAD*, pp. 726-731, 2002.
- [14] P. Schaumont *et al.*, "Cooperative multithreading on embedded multiprocessor architectures enables energy-scalable design," in *Proc. DAC*, pp. 27-30, 2005.
- [15] D. Shin *et al.*, "Optimizing intratask voltage scheduling using profile and data-flow information," *IEEE TCAD*, vol. 26, pp. 369-385, Feb. 2007.
- [16] P. Pillai *et al.*, "Real-time dynamic voltage scaling for low-power embedded operating systems," in *Proc. SOSP*, pp. 89-102, 2001.
- [17] LTC3733: 3-Phase Buck Controllers for AMD CPUs. 2003.
- [18] LTSPICE. www.linear.com.
- [19] M. Panagiotidis *et al.*, "Development and use of a regenerative braking model for a parallel hybrid electric vehicle," in *SAE International Congress*, vol. 109, pp. 1180-1191, 2000.